8-bit AVR® Microcontroller with Low Power 2.4GHz Transceiver for ZigBee and IEEE 802.15.4

ATmega128RFA1

PRELIMINARY

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 135 Powerful Instructions – Most Single Clock Cycle Execution
  - 32x8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 16 MIPS Throughput at 16 MHz and 1.8V
  - On-Chip 2-cycle Multiplier
- Non-volatile Program and Data Memories
  - 128K Bytes of In-System Self-Programmable Flash
    - Endurance: 2000 Write/Erase Cycles @ 85°C
  - 4K Bytes EEPROM
    - Endurance: 2000 Write/Erase Cycles @ 85°C
  - 16K Bytes Internal SRAM
- JTAG (IEEE std. 1149.1 compliant) Interface
  - Boundary-scan Capabilities According to the JTAG Standard
  - Extensive On-chip Debug Support
  - Programming of Flash EEPROM, Fuses and Lock Bits through the JTAG interface
- Peripheral Features
  - Multiple Timer/Counter & PWM channels
  - Real Time Counter with Separate Oscillator
  - 10-bit, 330 ks/s A/D Converter; Analog Comparator; On-chip Temperature Sensor
  - Master/Slave SPI Serial Interface
  - Two Programmable Serial USART
  - Byte Oriented 2-wire Serial Interface
- Advanced Interrupt Handler
- Watchdog Timer with Separate On-Chip Oscillator
- Power-on Reset and Low Current Brown-Out Detector
- Advanced Power Save Modes
- Fully integrated Low Power Transceiver for 2.4 GHz ISM Band
  - Supported Data Rates: 250 kb/s and 500 kb/s, 1 Mb/s, 2 Mb/s
  - -100 dBm RX Sensitivity; TX Output Power up to 3.5 dBm
  - Hardware Assisted MAC (Auto-Acknowledge, Auto-Retry)
  - 32 Bit IEEE 802.15.4 Symbol Counter
  - Baseband Signal Processing
  - SFR-Detection, Spreading; De-Spreading; Framing ; CRC-16 Computation
  - Antenna Diversity and TX/RX control
  - TX/RX 128 Byte Frame Buffer
- Hardware Security (AES, True Random Generator)
- Integrated Crystal Oscillators (32.768 kHz & 16 MHz)
- I/O and Package
  - 38 Programmable I/O Lines
  - 64-pad QFN (RoHS/Fully Green)
- Temperature Range: -40°C to 85°C Industrial
- Supply voltage range 1.8V to 3.6V with integrated voltage regulators
- Ultra Low Power consumption (1.8 to 3.6V) for Rx/Tx & AVR: <18.6 mA
  - CPU Active Mode (16MHz): 4.1 mA
  - 2.4GHz Transceiver: RX_ON 12.5 mA / TX 14.5 mA (maximum TX output power)
  - Deep Sleep Mode: <250nA @ 25°C
- Speed Grade: 0 – 16 MHz @ 1.8 – 3.6V

Applications

- ZigBee® / IEEE 802.15.4-2006/2003™ – Full And Reduced Function Device (FFD/RFD)
- General Purpose 2.4GHz ISM Band Transceiver with Microcontroller
- RF4CE, SP100, WirelessHART™, ISM Applications and IPv6 / 6LoWPAN

8266A-MCU Wireless-12/09

1 Pin Configurations

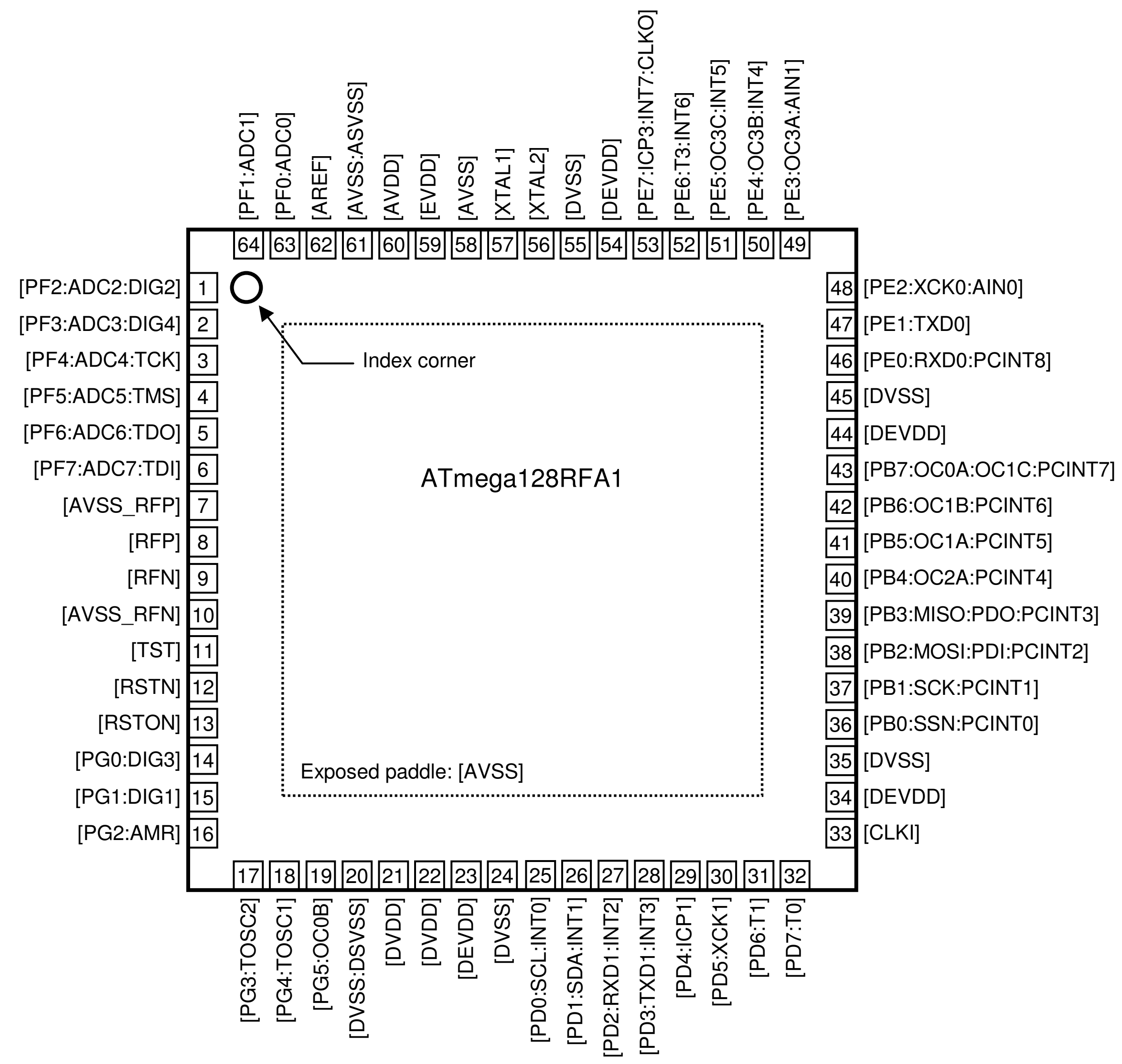


Figure 1-1. Pinout ATmega128RFA1

Note: The large center pad underneath the QFN/MLF package is made of metal and internally connected to AVSS. It should be soldered or glued to the board to ensure good mechanical stability. If the center pad is left unconnected, the package might loosen from the board

2 Disclaimer

Typical values contained in this datasheet are based on simulation and characterization results of other AVR microcontrollers and radio transceivers manufactured in a similar process technology. Minimum and Maximum values will be available after the device is characterized.

3 Overview

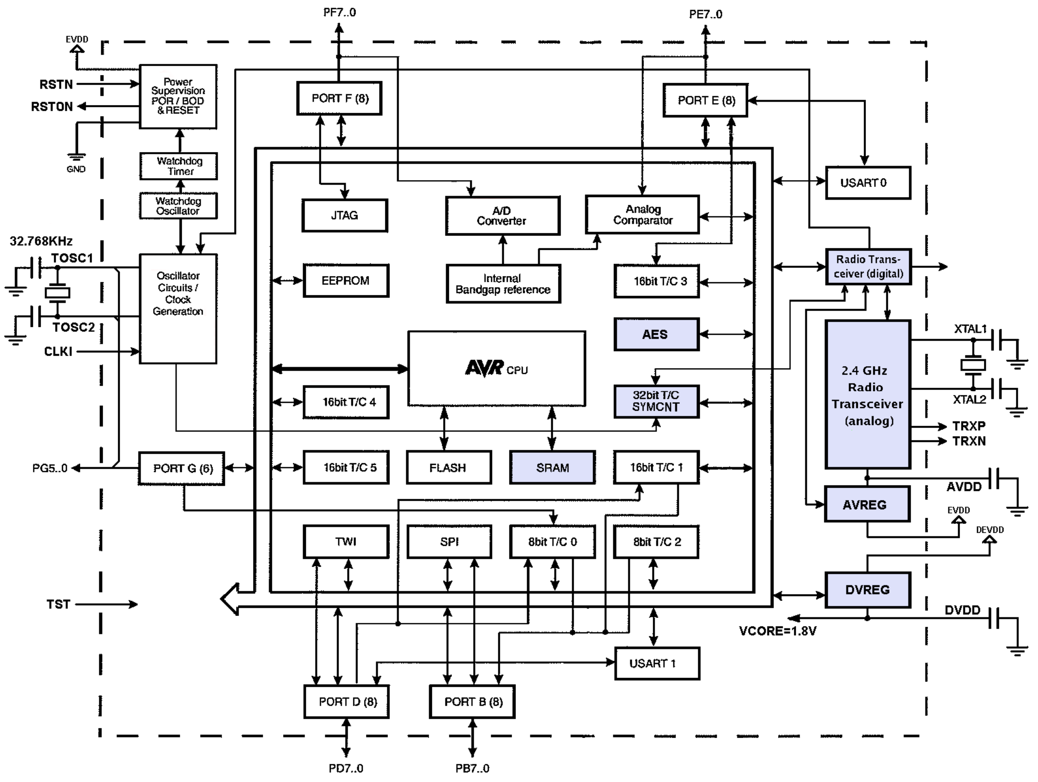
The ATmega128RFA1 is a low-power CMOS 8 bit microcontroller based on the AVR enhanced RISC architecture combined with a high data rate transceiver for the 2.4 GHz ISM band. It is derived from the ATmega1281 microcontroller and the AT86RF231 radio transceiver.

By executing powerful instructions in a single clock cycle, the device achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

The radio transceiver provides high data rates from 250 kb/s up to 2 Mb/s, frame handling, outstanding receiver sensitivity and high transmit output power enabling a very robust wireless communication.

3.1 Block Diagram

Figure 3-1 Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the Arithmetic Logic Unit (ALU). Two independent registers can be accessed with one single instruction executed in one clock cycle. The resulting architecture is very code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers. The system includes internal voltage regulation and an advanced power management. Distinguished by the small leakage current it allows an extended operation time from battery.

The radio transceiver is a fully integrated ZigBee solution using a minimum number of external components. It combines excellent RF performance with low cost, small size and low current consumption. The radio transceiver includes a crystal stabilized fractional-N synthesizer, transmitter and receiver, and full Direct Sequence Spread

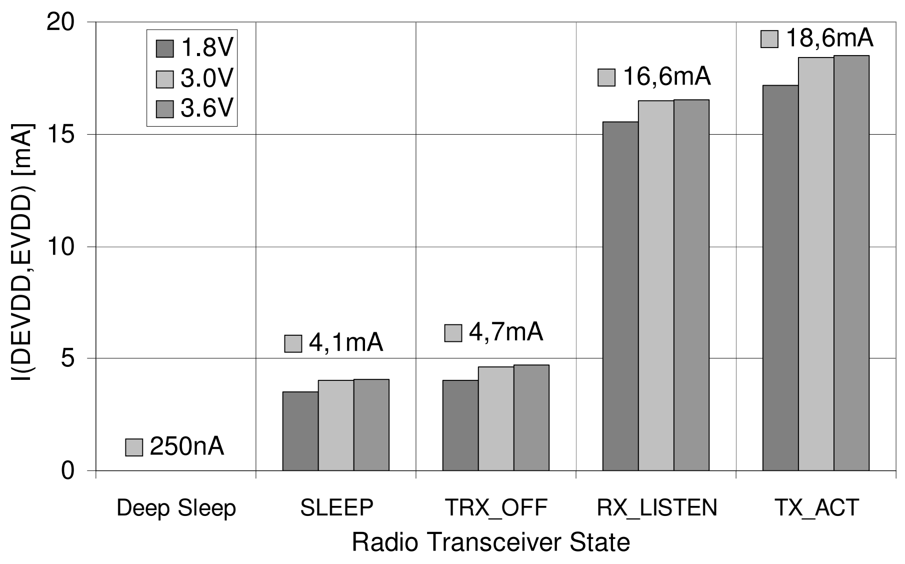
Spectrum Signal (DSSS) processing with spreading and despreading. The device is fully compatible with IEEE802.15.4-2006/2003 and ZigBee standards.

The ATmega128RFA1 provides the following features: 128 kbytes of In-System Programmable (ISP) Flash with read-while-write capabilities, 4 kbytes EEPROM, 16 kbytes SRAM, up to 35 general purpose I/O lines, 32 general purpose working registers, Real Time Counter (RTC), 6 flexible Timer/Counters with compare modes and PWM, USART, a byte oriented 2-wire Serial Interface, a 8 channel, 10 bit analog to digital converter (ADC) with an optional differential input stage with programmable gain, programmable Watchdog Timer with Internal Oscillator, a SPI serial port, IEEE std. 1149.1 compliant JTAG test interface, also used for accessing the On-chip Debug system and programming and 6 software selectable power saving modes.

The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the RC oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption. In Extended Standby mode, both the main RC oscillator and the asynchronous timer continue to run.

Typical supply current of the microcontroller with CPU clock set to 16MHz and the radio transceiver for the most important states is shown in the Figure 3-2 below.

Figure 3-2 Radio transceiver and microcontroller (16MHz) supply current



The transmit output power is set to maximum. If the radio transceiver is in SLEEP mode the current is dissipated by the AVR microcontroller only.

In Deep Sleep mode all major digital blocks with no data retention requirements are disconnected from main supply providing a very small leakage current. Watchdog timer, MAC symbol counter and 32.768kHz oscillator can be configured to continue to run.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed in-system trough an SPI serial interface, by a conventional nonvolatile memory programmer, or by on on-chip boot program running on the AVR core. The boot program can use any interface to download the application program in the application Flash memory. Software in the boot Flash section will continue to run while the application Flash section is updated, providing true Read-While-Write operation. By combining an 8 bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega128RFA1 is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega128RFA1 AVR is supported with a full suite of program and system development tools including: C compiler, macro assemblers, program debugger/simulators, in-circuit emulators, and evaluation kits.

3.2 Pin Descriptions

3.2.1 EVDD

External analog supply voltage;

3.2.2 DEVDD

External digital supply voltage;

3.2.3 AVDD

Regulated analog supply voltage (internally generated);

3.2.4 DVDD

Regulated digital supply voltage (internally generated);

3.2.5 DVSS

Digital ground;

3.2.6 AVSS

Analog ground;

3.2.7 Port B (PB7...PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B also provides functions of various special features of the ATmega128RFA1.

3.2.8 Port D (PD7...PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also provides functions of various special features of the ATmega128RFA1.

3.2.9 Port E (PE7...PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port E also provides functions of various special features of the ATmega128RFA1.

3.2.10 Port F (PF7...PF0)

Port F is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port F also provides functions of various special features of the ATmega128RFA1.

3.2.11 Port G (PG5…PG0)

Port G is a 6-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. However the driver strength of PG3 and PG4 is reduced compared to the other port pins. The output voltage drop (V_{OH}, V_{OL}) is higher while the leakage current is smaller. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port G also provides functions of various special features of the ATmega128RFA1.

3.2.12 AVSS_RFP

AVSS_RFP is a dedicated ground pin for the bi-directional, differential RF I/O port.

3.2.13 AVSS_RFN

AVSS_RFN is a dedicated ground pin for the bi-directional, differential RF I/O port.

3.2.14 RFP

RFP is the positive terminal for the bi-directional, differential RF I/O port.

3.2.15 RFN

RFN is the negative terminal for the bi-directional, differential RF I/O port.

3.2.16 RSTN

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset.

3.2.17 RSTON

Reset output. A low level on this pin indicates a reset initiated by the internal reset sources or the pin RSTN.

3.2.18 XTAL1

Input to the inverting 16MHz crystal oscillator amplifier. In general a crystal between XTAL1 and XTAL2 provides the 16MHz reference clock of the radio transceiver.

3.2.19 XTAL2

Output of the inverting 16MHz crystal oscillator amplifier;

3.2.20 AREF

Reference voltage output of the A/D Converter. In general this pin is left open.

3.2.21 TST

Programming and test mode enable pin;

3.2.22 CLKI

Input to the clock system. If selected, it provides the operating clock of the microcontroller.

3.3 Compatibility to ATmega1281/2561

The basic AVR feature set of the ATmega128RFA1 is derived from the ATmega1281/2561. Address locations and names of the implemented modules and registers are unchanged as long as it fits the target application of a very small and power efficient radio system. In addition, several new features were added.

Backward compatibility of the ATmega128RFA1 to the ATmega1281/2561 is provided in most cases. However some incompatibilities between the microcontrollers exist.

3.3.1 Port A and Port C

Port A and Port C are not implemented. The associated registers are available but will not provide any port control. Remaining ports are kept at their original address location to not require changes of existing software packages.

3.3.2 External Memory Interface

The alternate pin function “External Memory interface” using Port A and Port C is not implemented due to the missing ports.

The large internal data memory (SRAM) does not require an external memory and the associated parallel interface. It keeps the system radiation (EMC) at a very small level to provide very high sensitivity at the antenna input.

3.3.3 High Voltage Programming Mode

Alternate pin function BS2 (high voltage programming) of pin PA0 is mapped to a different pin. Entering the parallel programming mode is controlled by the TST pin.

3.3.4 AVR Oscillators and External Clock

The AVR microcontroller can utilize the high performance crystal oscillator of the 2.4GHz transceiver connected to the pins XTAL1 and XTAL2. An external clock can be applied to the microcontroller using the clock input CLKI.

3.3.5 Analog Frontend

The ATmega128RFA1 has a new A/D converter. Software compatibility is basically assured. Nevertheless to benefit from the higher conversion speeds and the better performance some changes are required.

4 Resources

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

5 About Code Examples

This documentation contains simple code examples that briefly show how to use various parts of the device. 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.

These code examples assume that the part specific header file is included before compilation. For I/O registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

6 Data Retention

Reliability Qualification results show that the projected data retention failure rate for the given ambient temperature is less than TBD PPM

- over 10 years at 85°C
- TBD years at 25°C.

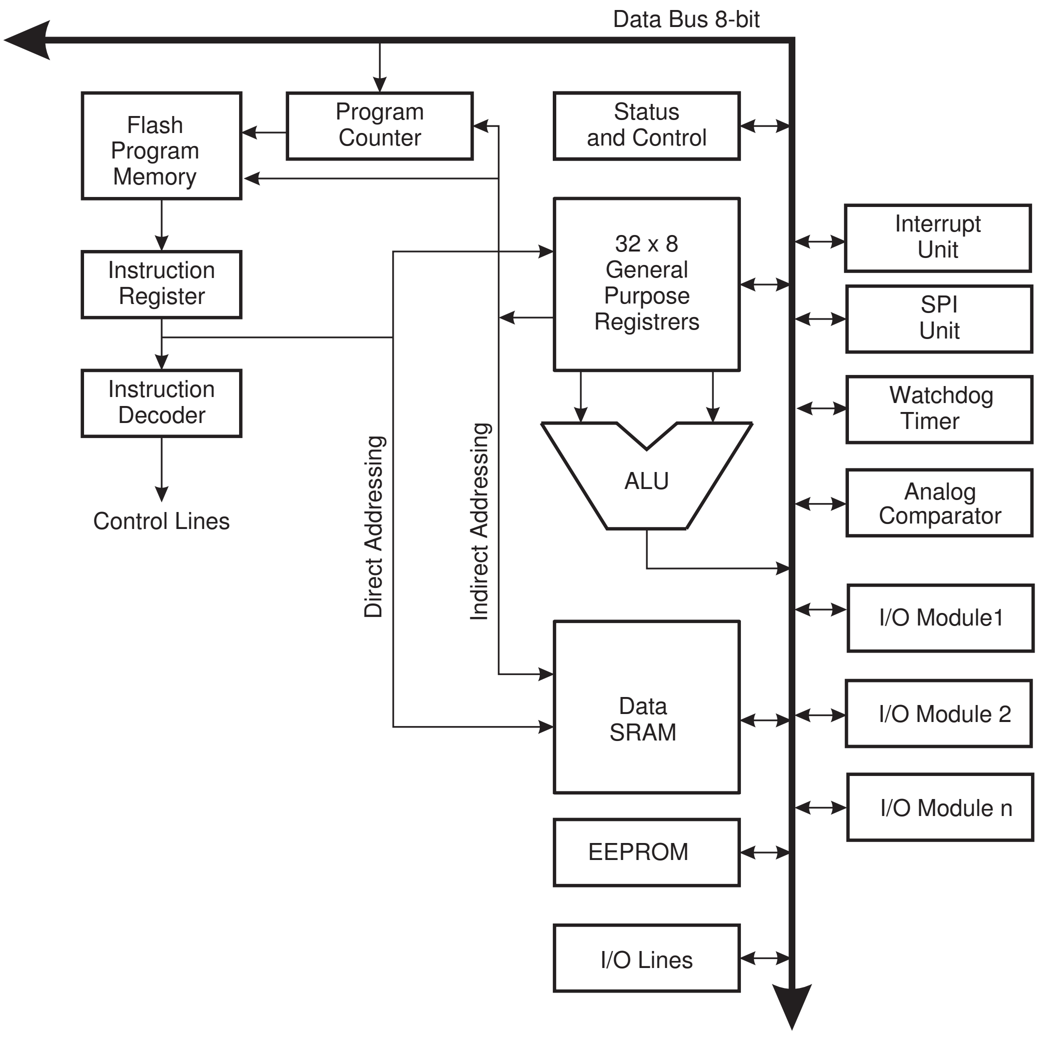
7 AVR CPU Core

7.1 Introduction

This section discusses the AVR core architecture in general. The main function of the CPU core is to ensure correct program execution. The CPU must therefore be able to access memories, perform calculation, control peripherals, and handle interrupts.

7.2 Architectural Overview

Figure 7-1. Block Diagram of the AVR Architecture



In order to maximize performance and parallelism, the AVR uses a Harvard architecture – with separate memories and buses for program and data. Instructions in the program memory are executed with a single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This concept enables instructions to be executed in every clock cycle. The program memory is In-System Reprogrammable Flash memory.

The fast-access Register File contains 32 x 8-bit general purpose working registers with a single clock cycle access time. This allows single-cycle Arithmetic Logic Unit (ALU) operation. In a typical ALU operation, two operands are output from the Register File, the operation is executed, and the result is stored back in the Register File – in one clock cycle.

Six of the 32 registers can be used as three 16-bit indirect address register pointers for Data Space addressing – enabling efficient address calculations. One of these address pointers can also be used as an address pointer for look up tables in Flash program memory. These added function registers are the 16-bit X-, Y-, and Z-register, described later in this section.

The ALU supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed in the ALU. After an arithmetic operation, the Status Register is updated to reflect information about the result of the operation.

Program flow is provided by conditional and unconditional jump and call instructions, able to directly address the whole address space. Most AVR instructions have a single 16-bit word format. Every program memory address contains a 16- or 32-bit instruction.

Program Flash memory space is divided in two sections, the Boot Program section and the Application Program section. Both sections have dedicated Lock bits for write and read/write protection. The SPM instruction that writes into the Application Flash memory section must reside in the Boot Program section.

During interrupts and subroutine calls, the return address Program Counter (PC) is stored on the Stack. The Stack is effectively 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. All user programs must initialize the SP in the Reset routine (before subroutines or interrupts are executed). The Stack Pointer (SP) is read/write accessible in the I/O space. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR architecture.

The memory spaces in the AVR architecture are all linear and regular memory maps.

A flexible interrupt module has its control registers in the I/O space with an additional Global Interrupt Enable bit in the Status Register. All interrupts have a separate Interrupt Vector in the Interrupt Vector table. The interrupts have priority in accordance with their Interrupt Vector position. The lower the Interrupt Vector address, the higher the priority.

The I/O memory space contains 64 addresses for CPU peripheral functions as Control Registers, SPI, and other I/O functions. The I/O Memory can be accessed directly, or as the Data Space locations following those of the Register File, 0x20 - 0x5F. In addition, the ATmega128RFA1 has Extended I/O space from 0x60 - 0x1FF in SRAM where only the ST/STS/STD and LD/LDS/LDD instructions can be used.

7.3 ALU – Arithmetic Logic Unit

The high-performance AVR ALU operates in direct connection with all the 32 general purpose working registers. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. The ALU operations are divided into three main categories – arithmetic, logical, and bit functions. Some implementations of the architecture also provide a powerful multiplier supporting both signed/unsigned multiplication and fractional format. See the "Instruction Set" section for a detailed description.

7.4 Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code. The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

7.4.1 SREG – Status Register

Bit	7	6	5	4	3	2	1	0	
$3F ($5F)	**I**	**T**	**H**	**S**	**V**	**N**	**Z**	**C**	**SREG**
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – I - Global Interrupt Enable**

The global interrupt enable bit must be set (one) for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the global interrupt enable bit is cleared (zero), none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts.

- **Bit 6 – T - Bit Copy Storage**

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

- **Bit 5 – H - Half Carry Flag**

The half carry flag H indicates a half carry in some arithmetic operations. See the Instruction Set Description for detailed information.

- **Bit 4 – S - Sign Bit**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the Instruction Set Description for detailed information.

- **Bit 3 – V - Two's Complement Overflow Flag**

The two's complement overflow flag V supports two's complement arithmetics. See the Instruction Set Description for detailed information.

- **Bit 2 – N - Negative Flag**

The negative flag N indicates a negative result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

- **Bit 1 – Z - Zero Flag**

The zero flag Z indicates a zero result after the different arithmetic and logic operations. See the Instruction Set Description for detailed information.

- **Bit 0 – C - Carry Flag**

The carry flag C indicates a carry in an arithmetic or logic operation. See the Instruction Set Description for detailed information. Note that the status register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt routine. This must be handled by software.

7.5 General Purpose Register File

The Register File is optimized for the AVR Enhanced RISC instruction set. In order to achieve the required performance and flexibility, the following input/output schemes are supported by the Register File:

- One 8-bit output operand and one 8-bit result input
- Two 8-bit output operands and one 8-bit result input
- Two 8-bit output operands and one 16-bit result input
- One 16-bit output operand and one 16-bit result input

Figure 7-1 below shows the structure of the 32 general purpose working registers in the CPU.

Figure 7-1. AVR CPU General Purpose Working Registers

	7 0	Addr.	
	R0	0x00	
	R1	0x01	
	R2	0x02	
	…		
	R13	0x0D	
General	R14	0x0E	
Purpose	R15	0x0F	
Working	R16	0x10	
Registers	R17	0x11	
	…		
	R26	0x1A	X-register Low Byte
	R27	0x1B	X-register High Byte
	R28	0x1C	Y-register Low Byte
	R29	0x1D	Y-register High Byte
	R30	0x1E	Z-register Low Byte
	R31	0x1F	Z-register High Byte

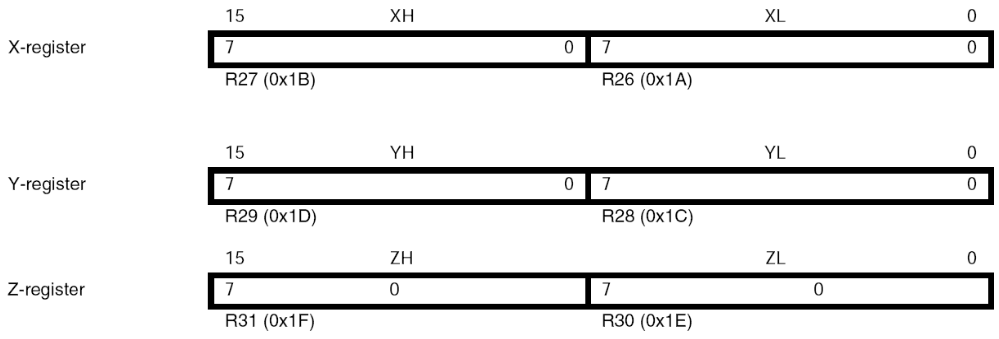
Most of the instructions operating on the Register File have direct access to all registers, and most of them are single cycle instructions.

As shown in Figure 7-1 above on page 12, each register is also assigned a data memory address, mapping them directly into the first 32 locations of the user Data Space. Although not being physically implemented as SRAM locations, this memory organization provides great flexibility in access of the registers, as the X-, Y- and Z-pointer registers can be set to index any register in the file.

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

The registers R26...R31 have some added functions to their general purpose usage. These registers are 16-bit address pointers for indirect addressing of the data space. The three indirect address registers X, Y, and Z are defined as described in Figure 7-2 on page 13.

Figure 7-2. The X-, Y-, Z-registers



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

7.6 Stack Pointer

The Stack is mainly used for storing temporary data, for storing local variables and for storing return addresses after interrupts and subroutine calls. The Stack Pointer Register always points to the top of the Stack. Note that the Stack is implemented as growing from higher memory locations to lower memory locations. This implies that a Stack PUSH command decreases the Stack Pointer.

The Stack Pointer points to the data SRAM Stack area where the Subroutine and Interrupt Stacks are located. This Stack space in the data SRAM must be defined by the program before any subroutine calls are executed or interrupts are enabled. The Stack Pointer must be set to point above 0x0200. The initial value of the stack pointer is the last address of the internal SRAM.

The Stack Pointer is decremented by one when data is pushed onto the Stack with the PUSH instruction, and it is decremented by two when the return address is pushed onto the Stack with subroutine call or interrupt. The Stack Pointer is incremented by one when data is popped from the Stack with the POP instruction, and it is incremented by two when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

When the FLASH memory exceeds 128Kbyte one additional cycle is required. In this case the Stack Pointer is decremented by three when the return address is pushed onto the Stack with subroutine call or interrupt and is incremented by three when data is popped from the Stack with return from subroutine RET or return from interrupt RETI.

7.6.1 SPH – Stack Pointer High

Bit	7	6	5	4	3	2	1	0	
$3E ($5E)	**SP15**	**SP14**	**SP13**	**SP12**	**SP11**	**SP10**	**SP9**	**SP8**	**SPH**
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	1	0	0	0	0	1	

The AVR Stack Pointer is implemented as two 8-bit registers SPL and SPH in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

- **Bit 7:0 – SP15:8 - Stack Pointer High Byte**

7.6.2 SPL – Stack Pointer Low

Bit	7	6	5	4	3	2	1	0	
$3D ($5D)	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0	SPL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	1	1	1	1	1	1	1	1	

The AVR Stack Pointer is implemented as two 8-bit registers SPL and SPH in the I/O space. The number of bits actually used is implementation dependent. Note that the data space in some implementations of the AVR architecture is so small that only SPL is needed. In this case, the SPH Register will not be present.

- **Bit 7:0 – SP7:0 - Stack Pointer Low Byte**

7.6.3 RAMPZ – Extended Z-pointer Register for ELPM/SPM

Bit	7	6	5	4	3	2	1	0	
$3B ($5B)	Res5	Res4	Res3	Res2	Res1	Res0	RAMPZ1	RAMPZ0	RAMPZ
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

For ELPM/SPM instructions, the Z-pointer is a concatenation of RAMPZ, ZH, and ZL. Note that LPM is not affected by the RAMPZ setting.

- **Bit 7:2 – Res5:0 - Reserved**

For compatibility with future devices, be sure to write these bits to zero.

- **Bit 1:0 – RAMPZ1:0 - Extended Z-Pointer Value**

These two bits represent the MSB's of the Z-Pointer.

Table 7-2 RAMPZ Register Bits

Register Bits	Value	Description
RAMPZ1:0	0	Default value of Z-pointer MSB's.

For ELPM/SPM instructions, the Z-pointer is a concatenation of RAMPZ, ZH, and ZL, as shown in Figure 7-3 below. Note that LPM is not affected by the RAMPZ setting.

Figure 7-3. The Z-pointer used by ELPM and SPM

Bit (Individually)	7	0	7	0	7	0
	RAMPZ		ZH		ZL	
Bit (Z-pointer)	23	16	15	8	7	0

The actual number of bits is implementation dependent. Unused bits in an implementation will always read as zero. For compatibility with future devices, be sure to write these bits to zero.

7.7 Instruction Execution Timing

Figure 7-4. The Parallel Instruction Fetches and Instruction Executions

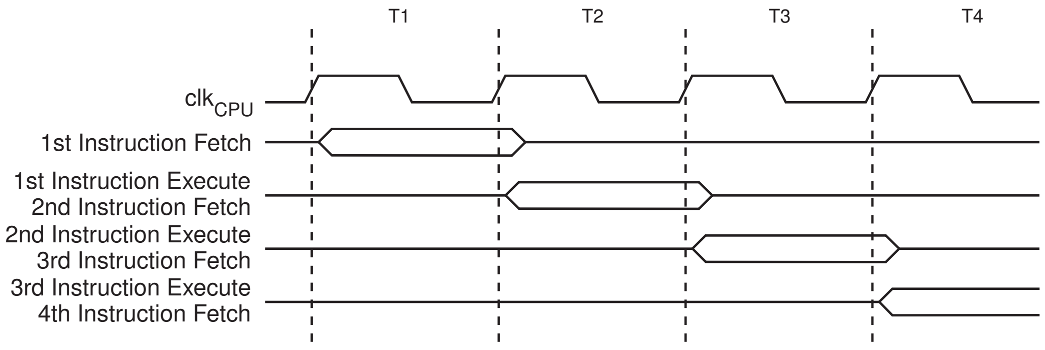
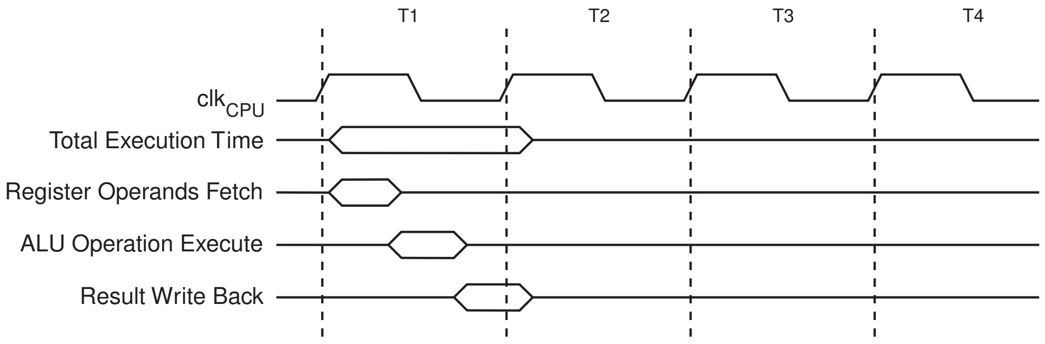


Figure 7-5 below shows the internal timing concept for the Register File. In a single clock cycle an ALU operation using two register operands is executed, and the result is stored back to the destination register.

Figure 7-5. Single Cycle ALU operation



7.8 Reset and Interrupt Handling

The AVR provides several different interrupt sources. These interrupts and the separate Reset Vector each have a separate program vector in the program memory space. All interrupts are assigned individual enable bits which must be written logic one together with the Global Interrupt Enable bit in the Status Register in order to enable the interrupt. Depending on the Program Counter value, interrupts may be automatically disabled when Boot Lock bits BLB02 or BLB12 are programmed. This feature improves software security. See the section "Memory Programming" on page 464 for details.

The lowest addresses in the program memory space are by default defined as the Reset and Interrupt Vectors. The complete list of vectors is shown in "Interrupts" on page 211. The list also determines the priority levels of the different interrupts. The lower the address the higher is the priority level. RESET has the highest priority, and next is INT0 – the External Interrupt Request 0. The Interrupt Vectors can be moved to the start of the Boot Flash section by setting the IVSEL bit in the MCU Control Register (MCUCR). Refer to "Interrupts" on page 211 for more information. The Reset Vector can also be moved to the start of the Boot Flash section by programming the BOOTRST Fuse, see "Memory Programming" on page 464.

When an interrupt occurs, the Global Interrupt Enable I-bit is cleared and all interrupts are disabled. The user software can write logic one to the I-bit to enable nested interrupts. All enabled interrupts can then interrupt the current interrupt routine. The I-bit is automatically set when a Return from Interrupt instruction – RETI – is executed.

There are basically two types of interrupts. The first type is triggered by an event that sets the Interrupt Flag. For these interrupts, the Program Counter is vectored to the actual Interrupt Vector in order to execute the interrupt handling routine, and hardware clears the corresponding Interrupt Flag. Interrupt Flags can also be cleared by writing a logic one to the flag bit position(s) to be cleared. If an interrupt condition occurs while the corresponding interrupt enable bit is cleared, the Interrupt Flag will be set and remembered until the interrupt is enabled, or the flag is cleared by software. Similarly, if one or more interrupt conditions occur while the Global Interrupt Enable bit is cleared, the corresponding Interrupt Flag(s) will be set and remembered until the Global Interrupt Enable bit is set, and will then be executed by order of priority.

The second type of interrupts will trigger as long as the interrupt condition is present. These interrupts do not necessarily have Interrupt Flags. If the interrupt condition disappears before the interrupt is enabled, the interrupt will not be triggered.

When the AVR exits from an interrupt, it will always return to the main program and execute one more instruction before any pending interrupt is served.

Note that the Status Register is not automatically stored when entering an interrupt routine, nor restored when returning from an interrupt routine. This must be handled by software.

When using the CLI instruction to disable interrupts, the interrupts will be immediately disabled. No interrupt will be executed after the CLI instruction, even if it occurs simultaneously with the CLI instruction. The following example shows how this can be used to avoid interrupts during the timed EEPROM write sequence.

Assembly Code Example

```
in r16, SREG ; store SREG value
cli ; disable interrupts during timed sequence
sbi EECR, EEMPE ; start EEPROM write
sbi EECR, EEPE
out SREG, r16 ; restore SREG value (I-bit)
```

C Code Example

```
char cSREG;
cSREG = SREG; /* store SREG value */
/* disable interrupts during timed sequence */
__disable_interrupt();
EECR |= (1<<EEMPE); /* start EEPROM write */
EECR |= (1<<EEPE);
SREG = cSREG; /* restore SREG value (I-bit) */
```

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
```

Assembly Code Example

```
; note: will enter sleep before any pending
; interrupt(s)
```

C Code Example

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

7.8.1 Interrupt Response Time

The interrupt execution response for all the enabled AVR interrupts is five clock cycles minimum. After five clock cycles the program vector address for the actual interrupt handling routine is executed. During these five 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 five 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 five clock cycles. During these five clock cycles, the Program Counter (three bytes) is popped back from the Stack, the Stack Pointer is incremented by three, and the I-bit in SREG is set.

8 AVR Memories

This section describes the different memories in the ATmega128RFA1. The AVR architecture has two main memory spaces, the Data Memory and the Program Memory space. In addition, the ATmega128RFA1 features an EEPROM Memory for data storage. All three memory spaces are linear and regular.

8.1 In-System Reprogrammable Flash Program Memory

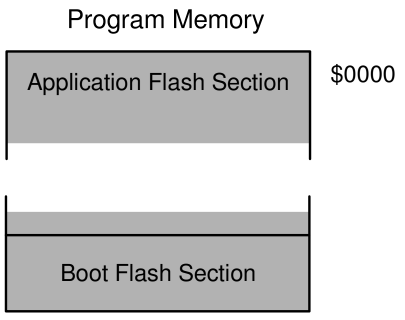
The ATmega128RFA1 contains 128K bytes On-chip In-System Reprogrammable Flash memory for program storage, see Figure 8-6 below. Since all AVR instructions are 16 or 32 bits wide, the Flash is 16 bit wide. For software security, the Flash Program memory space is divided into two sections, Boot Program section and Application Program section.

The Flash memory has an endurance of at least 2000 write/erase cycles. The ATmega128RFA1 Program Counter (PC) is 16 bits wide, thus addressing the required program memory locations. The operation of Boot Program section and associated Boot Lock bits for software protection are described in detail in "Boot Loader Support – Read-While-Write Self-Programming" on page 450. "Memory Programming" on page 464 contains a detailed description on Flash data serial downloading using the SPI pins or the JTAG interface.

Constant tables can be allocated within the entire program memory address space (see the LPM – Load Program Memory instruction description and ELPM – Extended Load Program Memory instruction description).

Timing diagrams for instruction fetch and execution are presented in "Instruction Execution Timing" on page 15.

Figure 8-6. Program Flash Memory Map



8.2 SRAM Data Memory

Figure 8-7 on page 19 shows how the ATmega128RFA1 SRAM Memory is organized. The ATmega128RFA1 is a complex microcontroller with more peripheral units than can be supported within the 64 location reserved in the Opcode for the IN and OUT instructions. For the Extended I/O space from $060 – $1FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

The first 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 416 locations of Extended I/O memory and the following 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.

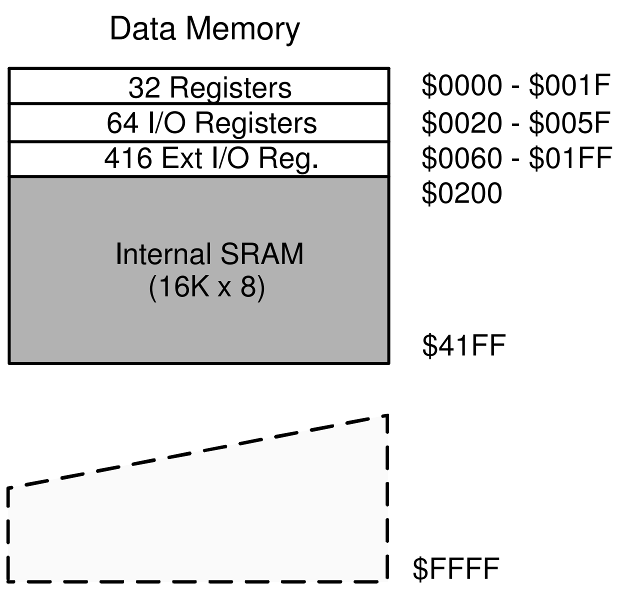
The direct addressing reaches the entire data space.

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, and the internal data SRAM in the ATmega128RFA1 are all accessible through all these addressing modes. The Register File is described in "General Purpose Register File" on page 12.

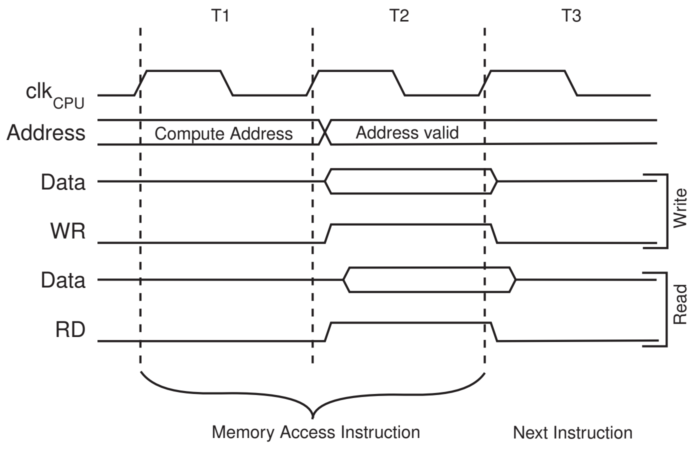
Figure 8-7. Data Memory Map



8.2.1 Data Memory Access Times

This section describes the general access timing concepts for internal memory access. Access to the internal data SRAM is performed in two clk$_{CPU}$ cycles as described in Figure 8-8 on page 20.

Figure 8-8. On-Chip Data SRAM Access Cycles



8.3 EEPROM Data Memory

The ATmega128RFA1 contains 4Kbyte 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 2000 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 SPI, JTAG and Parallel data downloading to the EEPROM, see "Serial Downloading" on page 477, "Programming via the JTAG Interface" on page 481, and "Programming the EEPROM" on page 491 respectively.

8.3.1 EEPROM Read Write Access

The EEPROM Access Registers are accessible in the I/O space, see "EEPROM Register Description" on page 23.

The write access time for the EEPROM is given in Table 8-3 below. 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 heavily filtered power supplies, DVDD is likely to rise or fall slowly on power-up/down. This causes the device for some period of time to run at a voltage lower than specified as minimum for the clock frequency used. See "Preventing EEPROM Corruption" on page 22 for details on how to avoid problems in these situations.

In order to prevent unintentional EEPROM writes, a specific write procedure must be followed. See the description of the EEPROM Control Register for details on this, "EEPROM Register Description" on page 23.

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.

The calibrated Oscillator is used to time the EEPROM accesses. The following table lists the typical programming time for EEPROM access from the CPU.

Table 8-3. EEPROM Programming Time

Symbol	Typical Programming time
EEPROM write (from CPU)	**4ms**
EEPROM erase (from CPU)	**8ms**

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.

Assembly Code Example

```
EEPROM_write:
  ; Wait for completion of previous write
  sbic EECR,EEPE
  rjmp EEPROM_write
  ; Set up address (r18:r17) in address register
  out EEARH, r18
  out EEARL, r17
  ; Write data (r16) to Data Register
  out EEDR,r16
  ; Write logical one to EEMPE
  sbi EECR,EEMPE
  ; Start eeprom write by setting EEPE
  sbi EECR,EEPE
  ret
```

C Code Example

```
void EEPROM_write(unsigned int uiAddress, unsigned char ucData)
{
  /* Wait for completion of previous write */
  while(EECR & (1<<EEPE))
    ;
  /* Set up address and Data Registers */
  EEAR = uiAddress;
  EEDR = ucData;
  /* Write logical one to EEMPE */
  EECR |= (1<<EEMPE);
  /* Start eeprom write by setting EEPE */
  EECR |= (1<<EEPE);
}
```

The next code examples show assembly and C functions for reading the EEPROM. The examples assume that interrupts are controlled so that no interrupts will occur during execution of these functions.

Assembly Code Example

```
EEPROM_read:
  ; Wait for completion of previous write
```

Assembly Code Example

```
    sbic EECR,EEPE
    rjcmp EEPROM_read
    ; Set up address (r18:r17) in address register
    out EEARH, r18
    out EEARL, r17
    ; Start eeprom read by writing EERE
    sbi EECR,EERE
    ; Read data from Data Register
    in r16,EEDR
    ret
```

C Code Example

```
  unsigned char EEPROM_read(unsigned int uiAddress)
  {
    /* Wait for completion of previous write */
    while(EECR & (1<<EEPE))
      ;
    /* Set up address register */
    EEAR = uiAddress;
    /* Start eeprom read by writing EERE */
    EECR |= (1<<EERE);
    /* Return data from Data Register */
    return EEDR;
  }
```

8.3.2 Preventing EEPROM Corruption

During periods of low DEVDD, the EEPROM data can be corrupted because the supply voltage is too low for the CPU and the EEPROM to operate properly. These issues are the same as for board level systems using EEPROM, and the same design solutions should be applied.

An EEPROM data corruption can be caused by two situations when the voltage is too low. First, a regular write sequence to the EEPROM requires a minimum voltage to operate correctly. Secondly, the CPU itself can execute instructions incorrectly, if the supply voltage is too low.

EEPROM data corruption can easily be avoided by following this design recommendation:

Keep the AVR RESET active (low) during periods of insufficient power supply voltage. This can be done by enabling the internal Brown-out Detector (BOD). If the detection level of the internal BOD does not match the needed detection level, an external low VCC reset Protection circuit can be used. If a reset occurs while a write operation is in progress, the write operation will be completed provided that the power supply voltage is sufficient.

8.4 EEPROM Register Description

8.4.1 EEARH – EEPROM Address Register High Byte

Bit	7	6	5	4	3	2	1	0	
$22 ($42)	Res3	Res2	Res1	Res0	EEAR11	EEAR10	EEAR9	EEAR8	EEARH
Read/Write	R	R	R	R	RW	RW	RW	RW	
Initial Value	0	0	0	0	X	X	X	X	

The EEPROM Address Registers EEARH and EEARL specify the EEPROM address in the 4K bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 4096. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

- **Bit 7:4 – Res3:0 - Reserved**
- **Bit 3:0 – EEAR11:8 - EEPROM Address**

8.4.2 EEARL – EEPROM Address Register Low Byte

Bit	7	6	5	4	3	2	1	0	
$21 ($41)	EEAR7	EEAR6	EEAR5	EEAR4	EEAR3	EEAR2	EEAR1	EEAR0	EEARL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	X	X	X	X	X	X	X	X	

The EEPROM Address Registers EEARH and EEARL specify the EEPROM address in the 4K bytes EEPROM space. The EEPROM data bytes are addressed linearly between 0 and 4096. The initial value of EEAR is undefined. A proper value must be written before the EEPROM may be accessed.

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

8.4.3 EEDR – EEPROM Data Register

Bit	7	6	5	4	3	2	1	0	
$20 ($40)	EEDR7:0								EEDR
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

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

- **Bit 7:0 – EEDR7:0 - EEPROM Data**

8.4.4 EECR – EEPROM Control Register

Bit	7	6	5	4	3	2	1	0	
$1F ($3F)	Res1	Res0	EEPM1	EEPM0	EERIE	EEMPE	EEPE	EERE	EECR
Read/Write	R	R	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	X	X	0	0	X	0	

- **Bit 7:6 – Res1:0 - Reserved**
- **Bit 5:4 – EEPM1:0 - EEPROM Programming Mode**

The EEPROM Programming mode bit setting defines which programming action will be triggered when writing EEPE. It is possible to program data in one atomic operation (erase the old value and program the new value) or to split the Erase and Write operations in two different operations. The Programming times for the different modes are shown in the following table. While EEPE is set, any write to EEPM1:0 will be ignored. During reset, the EEPM1:0 bits will be reset to 0 unless the EEPROM is busy programming.

Table 8-4 EEPM Register Bits

Register Bits	Value	Description
EEPM1:0	0x00	Erase and Write in one operation (Atomic Operation)
	0x01	Erase only
	0x02	Write only
	0x03	Reserved for future use

- **Bit 3 – EERIE - EEPROM Ready Interrupt Enable**

Writing EERIE to one enables the EEPROM Ready Interrupt if the I bit in SREG is set. Writing EERIE to zero disables the interrupt. The EEPROM Ready interrupt generates a constant interrupt when EEPE is cleared.

- **Bit 2 – EEMPE - EEPROM Master Write Enable**

The EEMPE bit determines whether setting EEPE to one causes the EEPROM to be written. When EEMPE is set, setting EEPE within four clock cycles will write data to the EEPROM at the selected address If EEMPE is zero, setting EEPE will have no effect. When EEMPE has been written to one by software, hardware clears the bit to zero after four clock cycles. See the description of the EEPE bit for an EEPROM write procedure.

- **Bit 1 – EEPE - EEPROM Programming Enable**

The EEPROM Write Enable Signal EEPE is the write strobe to the EEPROM. When address and data are correctly set up, the EEPE bit must be written to one to write the value into the EEPROM. The EEMPE bit must be written to one before a logical one is written to EEPE, otherwise no EEPROM write takes place. The following procedure should be adopted when writing the EEPROM (the order of steps 3 and 4 is not essential):

1. Wait until EEPE becomes zero.
2. Wait until SPMEN in SPMCSR becomes zero.
3. Write new EEPROM address to EEAR (optional).
4. Write new EEPROM data to EEDR (optional).
5. Write a logical one to the EEMPE bit while writing a zero to EEPE in EECR.
6. Within four clock cycles after setting EEMPE, write a logical one to EEPE.

The EEPROM can not be programmed during a CPU write to the Flash memory. The software must check that the Flash programming is completed before initiating a new EEPROM write. Step 2 is only relevant if the software contains a Boot Loader allowing the CPU to program the Flash. If the Flash is never being updated by the CPU, step 2 can be omitted.

Caution: an interrupt between step 5 and step 6 will make the write cycle fail, since the EEPROM Master Write Enable will time-out. If an interrupt routine accessing the EEPROM is interrupting another EEPROM access, the EEAR 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 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.

- **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 EEAR 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 EEAR Register.

8.5 I/O Memory

The Input/Output (I/O) space definition of the ATmega128RFA1 is shown in "Register Summary" on page 496.

All ATmega128RFA1 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 instructions. Refer to the AVR instruction set 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 ATmega128RFA1 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 – 0x1FF in SRAM, only the ST/STS/STD and LD/LDS/LDD instructions can be used.

For compatibility with future devices, reserved bits may not be modified. Reserved registers and 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 control registers of I/O and peripherals are explained in later sections.

8.6 General Purpose I/O Registers

The ATmega128RFA1 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.

8.6.1 GPIOR0 – General Purpose IO Register 0

Bit	7	6	5	4	3	2	1	0	
$1E ($3E)	GPIOR07:00								GPIOR0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The three General Purpose I/O Registers can be used for storing any information.

- **Bit 7:0 – GPIOR07:00 - General Purpose I/O Register 0 Value**

8.6.2 GPIOR1 – General Purpose IO Register 1

Bit	7	6	5	4	3	2	1	0	
$2A ($4A)	GPIOR17:10								GPIOR1
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The three General Purpose I/O Registers can be used for storing any information.

- **Bit 7:0 – GPIOR17:10 - General Purpose I/O Register 1 Value**

8.6.3 GPIOR2 – General Purpose I/O Register 2

Bit	7	6	5	4	3	2	1	0	
$2B ($4B)	GPIOR27:20								GPIOR2
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The three General Purpose I/O Registers can be used for storing any information.

- **Bit 7:0 – GPIOR27:20 - General Purpose I/O Register 2 Value**

8.7 Other Port Registers

The inherited control registers of missing ports located in the I/O space are kept in the ATmega128RFA1. They can be used as general purpose I/O registers for storing any information. Registers placed in the address range 0x00 – 0x1F are directly bit-accessible using the SBI, CBI, SBIS and SBIC instructions.

8.7.1 PORTA – Port A Data Register

Bit	7	6	5	4	3	2	1	0	
$02 ($22)	PORTA7:0								**PORTA**
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The PORTA register can be used as a General Purpose I/O Register for storing any information.

- **Bit 7:0 – PORTA7:0 - Port A Data Register Value**

8.7.2 DDRA – Port A Data Direction Register

Bit	7	6	5	4	3	2	1	0	
$01 ($21)	DDA7	DDA6	DDA5	DDA4	DDA3	DDA2	DDA1	DDA0	**DDRA**
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DDRA register can be used as a General Purpose I/O Register for storing any information.

- **Bit 7:0 – DDA7:0 - Port A Data Direction Register Value**

8.7.3 PINA – Port A Input Pins Address

Bit	7	6	5	4	3	2	1	0	
$00 ($20)	PINA7:0								**PINA**
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The PINA register is reserved for interal use and cannot be used as a General Purpose I/O Register.

- **Bit 7:0 – PINA7:0 - Port A Input Pins**

8.7.4 PORTC – Port C Data Register

Bit	7	6	5	4	3	2	1	0	
$08 ($28)	PORTC7:0								**PORTC**
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The PORTC register can be used as a General Purpose I/O Register for storing any information.

- **Bit 7:0 – PORTC7:0 - Port C Data Register Value**

8.7.5 DDRC – Port C Data Direction Register

Bit	7	6	5	4	3	2	1	0	
$07 ($27)	DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0	DDRC
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DDRC register can be used as a General Purpose I/O Register for storing any information.

- **Bit 7:0 – DDC7:0 - Port C Data Direction Register Value**

8.7.6 PINC – Port C Input Pins Address

Bit	7	6	5	4	3	2	1	0	
$06 ($26)	PINC7:0								PINC
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

The PINC register is reserved for interal use and cannot be used as a General Purpose I/O Register.

- **Bit 7:0 – PINC7:0 - Port C Input Pins**

9 Low-Power 2.4 GHz Transceiver

9.1 Features

- High performance RF-CMOS 2.4 GHz radio transceiver targeted for IEEE 802.15.4™, ZigBee™, IPv6 / 6LoWPAN, RF4CE, SP100, WirelessHART™ and ISM applications
- Outstanding link budget (103.5 dB):
  - Receiver sensitivity -100 dBm
  - Programmable output power from -17 dBm up to +3.5 dBm
- Ultra-low current consumption:
  - TRX_OFF = 0.4 mA
  - RX_ON = 12.5 mA
  - BUSY_TX = 14.5 mA (at max. transmit power of +3.5 dBm)
- Optimized for low BoM cost and ease of production:
  - Few external components necessary (crystal, capacitors and antenna)
  - Excellent ESD robustness
- Easy to use interface:
  - Registers and frame buffer access from software
  - Dedicated radio transceiver interrupts
- Radio transceiver features:
  - 128 byte FIFO (SRAM) for data buffering
  - Integrated RX/TX switch
  - Fully integrated, fast settling PLL to support frequency hopping
  - Battery monitor
  - Fast wake-up time < 0.25 ms
- Special IEEE 802.15.4 2006 hardware support:
  - FCS computation and clear channel assessment (CCA)
  - RSSI measurement, energy detection and link quality indication
- MAC hardware accelerator:
  - Automated acknowledgement, CSMA-CA and frame retransmission
  - Automatic address filtering
  - Automated FCS check
- Extended Feature Set Hardware Support:
  - AES 128 bit hardware accelerator
  - RX/TX indication (external RF front-end control)
  - RX antenna diversity
  - Supported PSDU data rates: 250 kb/s, 500 kb/s, 1 Mb/s and 2 Mb/s
  - True random number generation for security applications
- Compliant to IEEE 802.15.4-2006, IEEE 802.15.4-2003 and RF4CE
- Compliant to EN 300 328/440, FCC-CFR-47 Part 15, ARIB STD-66, RSS-210

The ATmega128RFA1 features a low-power 2.4 GHz radio transceiver designed for industrial and consumer ZigBee/IEEE 802.15.4, 6LoWPAN, RF4CE and high data rate 2.4 GHz ISM band applications. The radio transceiver is a true peripheral block of the AVR microcontroller. All RF-critical components except the antenna, crystal and de-coupling capacitors are integrated on-chip. Therefore, the ATmega128RFA1 is particularly suitable for applications like:

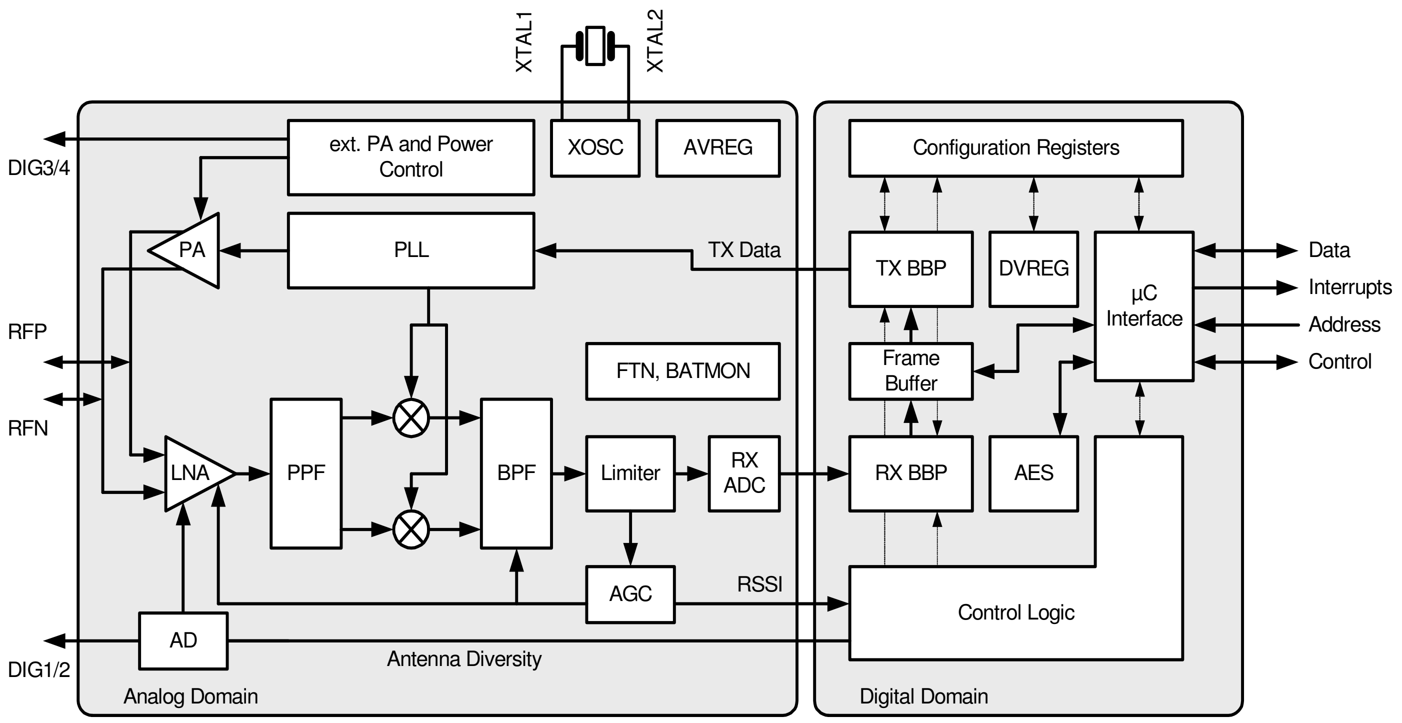
- 2.4 GHz IEEE 802.15.4 and ZigBee systems
- 6LoWPAN and RF4CE systems
- Wireless sensor networks
- Industrial control, sensing and automation (SP100, WirelessHART)
- Residential and commercial automation
- Health care
- Consumer electronics
- PC peripherals

9.2 General Circuit Description

This radio transceiver is part of a system-on-chip solution with an AVR® microcontroller. It comprises a complex peripheral component containing the analog radio, digital modulation and demodulation including time and frequency synchronization and data buffering. The number of external components for the transceiver operation is minimized such that only the antenna, the crystal and decoupling capacitors are required. The bidirectional differential antenna pins (RFP, RFN) are used for transmission and reception, thus no external antenna switch is needed.

The ATmega128RFA1 block diagram is shown in Figure 9-9 below.

Figure 9-9. Transceiver Block Diagram



The received RF signal at pins RFN and RFP is differentially fed through the low-noise amplifier (LNA) to the RF filter (PPF) to generate a complex signal, driving the integrated channel filter (BPF). The limiting amplifier provides sufficient gain to drive the succeeding analog-to-digital converter (RX ADC) and generates a digital RSSI signal. The RX ADC output signal is sampled by the digital base band receiver (RX BBP).

The transmit modulation scheme is offset-QPSK (O-QPSK) with half-sine pulse shaping and 32-length block coding (spreading) according to [1] on page 100 and [2] on page 100. The modulation signal is generated in the digital transmitter (TX BBP) and applied to the fractional-N frequency synthesis (PLL), to ensure the coherent phase modulation required for demodulation of O-QPSK signals. The frequency-modulated signal is fed to the power amplifier (PA).

A differential pin pair DIG3/DIG4 can be enabled to control an external RF front-end.

The two on-chip low-dropout voltage regulators (A|DVREG) provide the analog and digital 1.8V supply.

An internal 128-byte RAM for RX and TX (Frame Buffer) buffers the data to be transmitted or received.

The configuration of the reading and writing of the Frame Buffer is controlled via the microcontroller interface.

The transceiver further contains comprehensive hardware-MAC support (Extended Operating Mode) and a security engine (AES) to improve the overall system power efficiency and timing. The 128-bit AES engine can be accessed in parallel to all PHY operational transactions and states using the microcontroller interface, except during transceiver power down state.

For applications not necessarily targeting IEEE 802.15.4 compliant networks, the radio transceiver also supports alternative data rates up to 2 Mb/s.

For long-range applications or to improve the reliability of an RF connection the RF performance can further be improved by using an external RF front-end or Antenna Diversity. Both operation modes are supported by the radio transceiver with dedicated control pins without the interaction of the microcontroller.

Additional features of the Extended Feature Set, see section "Radio Transceiver Extended Feature Set" on page 85, are provided to simplify the interaction between radio transceiver and microcontroller.

9.3 Transceiver to Microcontroller Interface

This section describes the internal Interface between the transceiver module and the microcontroller. Unlike all other AVR I/O modules, the transceiver module can operate asynchronously to the controller. The transceiver requires an accurate 16MHz crystal clock for operation, but the controller can run at any frequency within its operating limits.

9.3.1 Transceiver Configuration and Data Access

9.3.1.1 Register Access

All transceiver registers are mapped into I/O space of the controller. Due to the asynchronous interface a register access can take up to three transceiver clock cycles. Depending on the controller clock speed, program execution wait cycles are generated. That means if the controller runs with about 16MHz or faster, at least three wait cycles are generated, but if the controller runs with about 4MHz, no wait cycles are inserted. A register access is only possible, if the transceiver clock is available. Therefore the transceiver must be enabled (PRR1 Register) and not in SLEEP state.

9.3.1.2 Frame Buffer Access

The 128-byte Frame Buffer can hold the PHY service data unit (PSDU) data of one IEEE 802.15.4 compliant RX or one TX frame of maximum length at a time. A detailed description of the Frame Buffer can be found in section "Frame Buffer" on page 77. An introduction to the IEEE 802.15.4 frame format can be found in section "Introduction – IEEE 802.15.4-2006 Frame Format" on page 61.

The Frame Buffer is located within the controller I/O address space above of the transceiver register set. The first byte of the Frame Buffer can be accessed with the symbolical address TRXFBST and the last byte can be accessed with the symbolical address TRXFBEND. Random access to single frame bytes is possible with "TRXFBST + byte index" or "TRXFBEND – byte index". In contrast to the transceiver register access, the Frame Buffer allows single cycle read/write operations for all controller clock speeds.

The content of the Frame Buffer is only overwritten by a new received frame or a Frame Buffer write access.

The Frame Buffer usage is different between received and transmitted frames. Therefore it is not possible to retransmit a received frame without modifying the frame buffer.

On received frames, the frame length byte is not stored in the Frame Buffer, but can be accessed over the TST_FRAME_LENGTH register. During frame receive, the Link Quality Indication (LQI) value (refer to "Link Quality Indication (LQI)" on page 72) is appended to the frame data in the Frame Buffer.

For frame transmission, the first byte of the Frame Buffer must contain the frame length information followed by the frame data. The TST_FRAME_LENGTH register does not need to be written in this case.

A detailed description of the Frame Buffer usage for receive and transmit frames can be found in Figure 9-31 on page 78.

Notes:

1. The Frame Buffer is shared between RX and TX; therefore, the frame data are overwritten by new incoming frames. If the TX frame data are to be retransmitted, it must be ensured that no frame was received in the meanwhile.
2. To avoid overwriting during receive, Dynamic Frame Buffer Protection can be enabled. For details about this feature refer to section "Dynamic Frame Buffer Protection" on page 91.
3. It is not possible to retransmit received frames without inserting the frame length information at the beginning of the Frame Buffer. That requires a complete read out of the received frame and rewriting the modified frame to the Frame Buffer.
4. For exceptions, e.g. receiving acknowledgement frames in Extended Operating Mode (TX_ARET) refer to section "TX_ARET_ON – Transmit with Automatic Retry and CSMA-CA Retry" on page 57.

9.3.1.3 Transceiver Pin Register TRXPR

The Transceiver Pin Register TRXPR is located in the Controller clock domain and is accessible even if the transceiver is in sleep state. This register provides access to the pin functionality, known from the RF231 devices (two chip solution).

The register (TRXRST) can be used to reset the transceiver without resetting the controller. After the reset bit was set, it is cleared immediately.

A second configuration bit (SLPTR) is used to control frame transmission or sleep and wakeup of the transceiver. This bit is not cleared automatically.

The function of the SLPTR bit relates to the current state of the transceiver module and is summarized in Table 9-1 below. The radio transceiver states are explained in detail in section "Operating Modes" on page 35.

Table 9-1. SLPTR Multi-functional Configuration bit

Transceiver Status	Function	SLPTR Bit	Description
PLL_ON	TX start	"0" ⇨ "1"	Starts frame transmission
TX_ARET_ON	TX start	"0" ⇨ "1"	Starts TX_ARET transaction
TRX_OFF	Sleep	"0" ⇨ "1"	Takes the radio transceiver into SLEEP state
SLEEP	Wakeup	"1" ⇨ "0"	Takes the radio transceiver back into TRX_OFF state;

In states PLL_ON and TX_ARET_ON, bit SLPTR is used to initiate a TX transaction. Here bit SLPTR is sensitive on the transition from "0" to "1" only. The bit should be cleared before the frame transmission is finished.

After initiating a state change by a "0" to "1" transition at bit SLPTR in radio transceiver states TRX_OFF, RX_ON or RX_AACK_ON, the radio transceiver remains in the new state as long as the bit is logical "1" and returns to the preceding state if the bit is set to "0".

SLEEP state

The SLEEP state is used when radio transceiver functionality is not required, and thus the receiver module can be powered down to reduce the overall power consumption.

When the radio transceiver is in TRX_OFF state the microcontroller forces the transceiver to SLEEP by setting SLPTR = "1". The transceiver awakes when the microcontroller releases bit SLPTR.

9.3.2 Interrupt Logic

9.3.2.1 Overview

The transceiver module differentiates between eight interrupt events. Internally all pending interrupt are stored in a separate bit of the interrupt status register (IRQ_STATUS). Each interrupt is enabled by setting the corresponding bit in the interrupt mask register (IRQ_MASK). If an IRQ is enabled an interrupt service routine must be defined to handle the IRQ. A pending IRQ is cleared automatically if an Interrupt service routine is called. It is also possible to handle IRQs manually by polling the IRQ_STATUS register. If an IRQ occurred, the appropriate IRQ_STATUS register bit is set. The IRQ can be cleared by writing '1' to the register bit. It is recommended to clear the corresponding status bit before enabling an interrupt.

Interrupts are not cleared automatically when the event that caused them vanishes. More information about interrupt handling by the controller can be found in section "Interrupts" on page 211.

The supported interrupts for the Basic Operating Mode are summarized in Table 9-2 on **page 34**.

Table 9-2. Interrupt Description in Basic Operating Mode

IRQ Vector Number/ Priority [(1)]	IRQ Name	Description	Section
64	TRX24_AWAKE	Indicates radio transceiver reached TRX_OFF state RESET, or SLEEP states	"TRX_OFF – Clock State" on page 36
63	TRX24_TX_END	Indicates the completion of a frame transmission	"Frame Transmit Procedure" on page 84
62	TRX24_XAH_AMI	Indicates address matching	"Frame Filtering" on page 54
61	TRX24_CCA_ED_DONE	Indicates the end of a CCA or ED measurement	"Energy Detection (ED)" on page 68
60	TRX24_RX_END	Indicates the completion of a frame reception	"Frame Transmit Procedure" on page 84
59	TRX24_RX_START	Indicates the start of a PSDU reception. The TRX_STATE changes to BUSY_RX, the PHR is ready to be read from Frame Buffer	"Frame Receive Procedure" on page 84
58	TRX24_PLL_UNLOCK	Indicates PLL unlock. If the radio transceiver is in BUSY_TX / BUSY_TX_ARET state, the PA is turned off immediately	"Interrupt Handling" on page 83
57	TRX24_PLL_LOCK	Indicates PLL lock	"Interrupt Handling" on page 83

Note: 1. The lowest IRQ Number has the highest priority.

During startup from SLEEP or RESET, the radio transceiver issues an TRX24_AWAKE interrupt when it enters state TRX_OFF.

If the microcontroller initiates an energy-detect (ED) or clear-channel-assessment (CCA) measurement, the completion of the measurement is indicated by interrupt TRX24_CCA_ED_DONE, refer to sections "Energy Detection (ED)" on page 68 and "Clear Channel Assessment (CCA)" on page 70 for details.

After RESET all interrupts are disabled. During radio transceiver initialization it is recommended to enable AWAKE to be notified once the TRX_OFF state is entered. Note that the TRX24_AWAKE interrupt can usually not be seen when the transceiver enters TRX_OFF state after RESET, because register IRQ_MASK is reset to mask all interrupts. In this case, state TRX_OFF is normally entered before the microcontroller could modify the register.

The interrupt handling in Extended Operating Mode is described in section "Interrupt Handling" on page 59.

9.3.3 Radio Transceiver Identification

The ATmega128RFA1 Transceiver module can be identified by four registers (PART_NUM, VERSION_NUM, MAN_ID_0, MAN_ID_1). One register contains a unique part number and one register the corresponding version number. Two additional registers contain the JTAG manufacture ID. The transceiver identification registers are provided for compatibility to the transceiver only device.

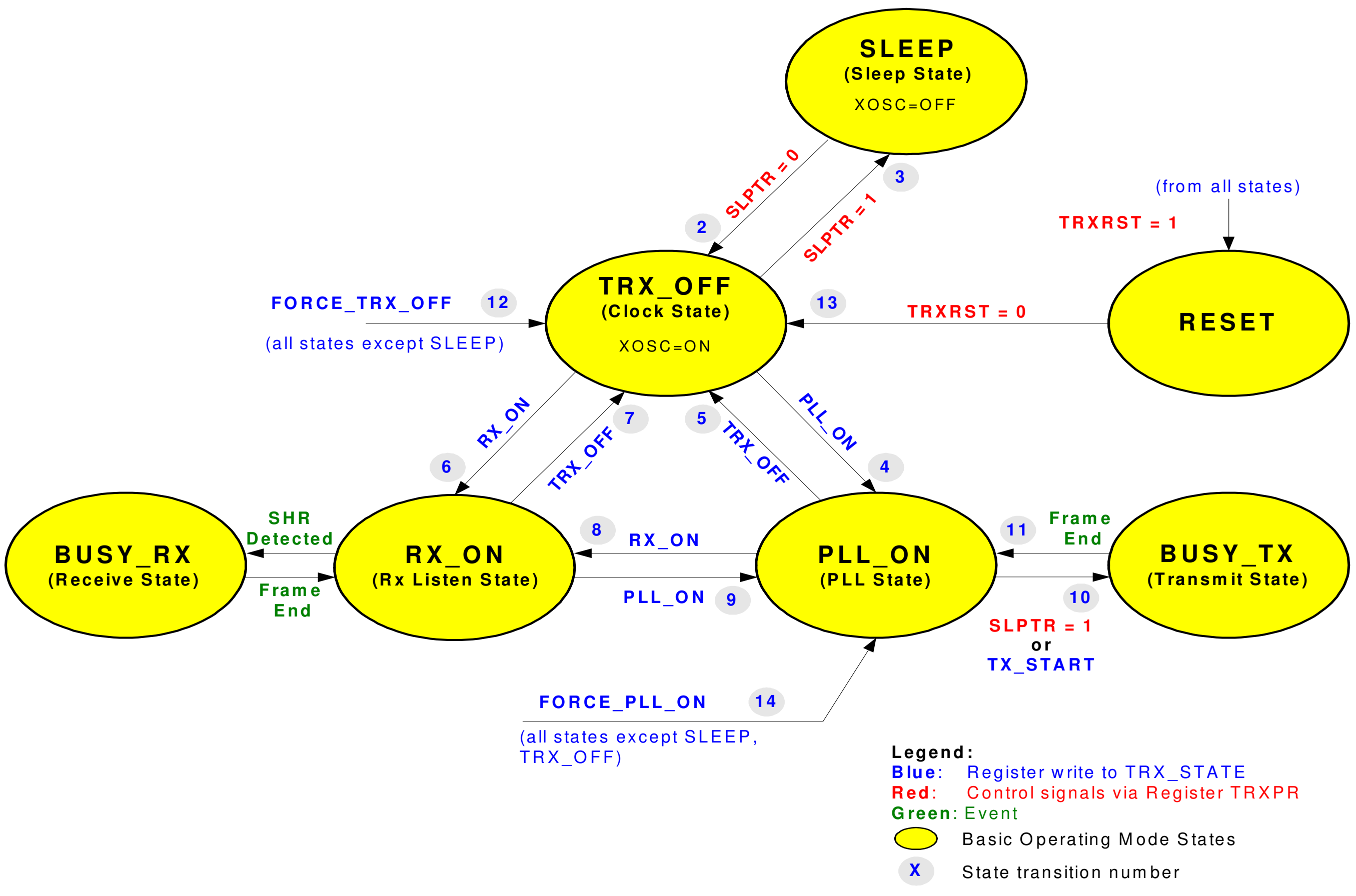
A unique device identification is also possible with the three AVR signature bytes. For details about accessing this information refer to "Signature Bytes" on page 466.

9.4 Operating Modes

9.4.1 Basic Operating Mode

This section summarizes all states to provide the basic functionality of the 2.4GHz radio transceiver, such as receiving and transmitting frames, the power up sequence and radio transceiver sleep. The Basic Operating Mode is designed for IEEE 802.15.4 and ISM applications; the corresponding radio transceiver states are shown in Figure 9-12 below.

Figure 9-12. Basic Operating Mode State Diagram (for timing refer to **Table 9-3 on page 42**)



Note: 1. State transition numbers correspond to Table 9-3 on page 42.

9.4.1.1 State Control

The radio transceiver states are controlled either by writing commands to bits TRX_CMD of register TRX_STATE, or directly by the two control bits SLPTR and TRXRST of the TRXPR register. A successful state change can be verified by reading the radio transceiver status from register TRX_STATUS.

If TRX_STATUS = 0x1F (STATE_TRANSITION_IN_PROGRESS) the radio transceiver is on a state transition. Do not try to initiate a further state change while the radio transceiver is in STATE_TRANSITION_IN_PROGRESS.

Bit SLPTR is a multifunctional bit (refer to section "Transceiver Pin Register TRXPR" on page 32 for more details). Dependent on the radio transceiver state, a “0” to “1” transition on SLPTR causes the following state transitions:

- TRX_OFF ➔ SLEEP
- PLL_ON ➔ BUSY_TX

Whereas resetting bit SLPTR to “0” causes the following state transitions:

- SLEEP ➔ TRX_OFF

Bit TRXRST causes a reset of all radio transceiver registers and forces the radio transceiver into TRX_OFF state.

For all states except SLEEP, the state change commands FORCE_TRX_OFF or TRX_OFF lead to a transition into TRX_OFF state. If the radio transceiver is in active receive or transmit states (BUSY_*), the command FORCE_TRX_OFF interrupts these active processes, and forces an immediate transition to TRX_OFF. In contrast a TRX_OFF command is stored until an active state (receiving or transmitting) has been finished. After that the transition to TRX_OFF is performed.

For a fast transition from receive or active transmit states to PLL_ON state the command FORCE_PLL_ON is provided. In contrast to FORCE_TRX_OFF this command does not disable the PLL and the analog voltage regulator AVREG. It is not available in states SLEEP, and RESET.

The completion of each requested state-change shall always be confirmed by reading the bits TRX_STATUS of register TRX_STATUS.

9.4.1.2 Basic Operating Mode Description

9.4.1.2.1 SLEEP – Sleep State

In radio transceiver SLEEP state, the entire radio transceiver is disabled. No circuitry is operating. The radio transceiver’s current consumption is reduced to leakage current only. This state can only be entered from state TRX_OFF, by setting the bit SLPTR = “1”.

Setting SLPTR = “0” returns the radio transceiver to the TRX_OFF state. During radio transceiver SLEEP the register contents remains valid while the content of the Frame Buffer and the security engine (AES) are cleared.

TRXRST = “1” in SLEEP state returns the radio transceiver to TRX_OFF state and thereby sets all registers to their reset values.

9.4.1.2.2 TRX_OFF – Clock State

This state is reached immediately after Power On or Reset. In TRX_OFF the crystal oscillator is running. The digital voltage regulator is enabled, thus the radio transceiver registers, the Frame Buffer and security engine (AES) are accessible (see section "Frame Buffer" on page 77 and "Security Module (AES)" on page 92).

SLPTR and TRXRST in register TRXPR can be used for state control (see "State Control" on page 35 for details). The analog front-end is disabled during TRX_OFF.

Entering the TRX_OFF state from radio transceiver SLEEP, or RESET state is indicated by the TRX24_AWAKE interrupt.

9.4.1.2.3 PLL_ON – PLL State

Entering the PLL_ON state from TRX_OFF state first enables the analog voltage regulator (AVREG). After the voltage regulator has been settled the PLL frequency synthesizer is enabled. When the PLL has been settled at the receive frequency to a channel defined by bits CHANNEL of register PHY_CC_CCA a successful PLL lock is indicated by issuing a TRX24_PLL_LOCK interrupt.

If an RX_ON command is issued in PLL_ON state, the receiver is immediately enabled. If the PLL has not been settled before the state change nevertheless takes place. Even if the register bits TRX_STATUS of register TRX_STATUS indicates RX_ON, actual frame reception can only start once the PLL has locked.

The PLL_ON state corresponds to the TX_ON state in IEEE 802.15.4.

9.4.1.2.4 RX_ON and BUSY_RX – RX Listen and Receive State

In RX_ON state the receiver blocks and the PLL frequency synthesizer are enabled.

The receive mode is internally separated into the RX_ON and BUSY_RX states. There is no difference between these states with respect to the analog radio transceiver circuitry, which are always turned on. In both states the receiver and the PLL frequency synthesizer are enabled.

During RX_ON state the receiver listens for incoming frames. After detecting a valid synchronization header (SHR), the receiver automatically enters the BUSY_RX state. The reception of a valid PHY header (PHR) generates an TRX24_RX_START interrupt and receives and demodulates the PSDU data.

During PSDU reception the frame data are stored continuously in the Frame Buffer until the last byte was received. The completion of the frame reception is indicated by an TRX24_RX_END interrupt and the radio transceiver reenters the state RX_ON. At the same time the bits RX_CRC_VALID of register PHY_RSSI are updated with the result of the FCS check (see "Frame Check Sequence (FCS)" on page 66).

Received frames are passed to the frame filtering unit, refer to section "Frame Filtering" on page 54. If the content of the MAC addressing fields of a frame (refer to IEEE 802.15.4 section 7.2.1) matches to the expected addresses, which is further dependent on the addressing mode, an address match interrupt (TRX24_XAH_AMI) is issued, refer to "Interrupt Logic" on page 33. The expected address values are to be stored in the registers Short-Address, PAN-ID and IEEE-address. Frame filtering is available in Basic and Extended Operating Mode, refer to section "Frame Filtering" on page 54.

Leaving state RX_ON is only possible by writing a state change command to bits TRX_CMD of register TRX_STATE.

9.4.1.2.5 BUSY_TX – Transmit State

A transmission can only be initiated in state PLL_ON. There are two ways to start a transmission:

- Setting Bit SLPTR of register TRXPR to ‘1’. The bit should be cleared before the frame has been transmitted. This mode is for legacy operation and should be replaced by the TX_START command below.
- TX_START command to bits TRX_CMD of register TRX_STATE.

Either of these causes the radio transceiver into the BUSY_TX state.

During the transition to BUSY_TX state, the PLL frequency shifts to the transmit frequency. The actual transmission of the first data chip of the SHR starts after 16 µs to allow PLL settling and PA ramp-up, see Figure 9-16 on page 40. After transmission of the SHR, the Frame Buffer content is transmitted. In case the PHR indicates a frame length of zero, the transmission is aborted.

After the frame transmission has completed, the radio transceiver automatically turns off the power amplifier, generates a TRX24_TX_END interrupt and returns into PLL_ON state.

9.4.1.2.6 RESET State

The RESET state is used to set back the state machine and to reset all registers of the radio transceiver to their default values.

A reset forces the radio transceiver into the TRX_OFF state.

A reset is initiated by a ATmega128RFA1 main reset (see "Resetting the AVR" on page 176) or a radio transceiver reset (see "Transceiver Pin Register TRXPR" on page 32).

During radio transceiver reset the TRXPR register is not cleared and therefore the application software has to set the SLPTR bit to “0”.

9.4.1.3 Interrupt Handling

All interrupts provided by the radio transceiver are supported in Basic Operating Mode (see Table 9-2 on page 34).

Required interrupts must be enabled by writing to register IRQ_MASK and the global interrupt enable flag must be set. For a general explanation of the interrupt handling refer to "Reset and Interrupt Handling" on page 15 and "Interrupt Logic" on page 33.

For example, interrupts are provided to observe the status of the RX and TX operations.

On receive the TRX24_RX_START interrupt indicates the detection of a valid PHR, the TRX24_XAH_AMI interrupt an address match and the TRX24_RX_END interrupt the completion of the frame reception.

On transmit the TRX24_TX_END interrupt indicates the completion of the frame transmission.

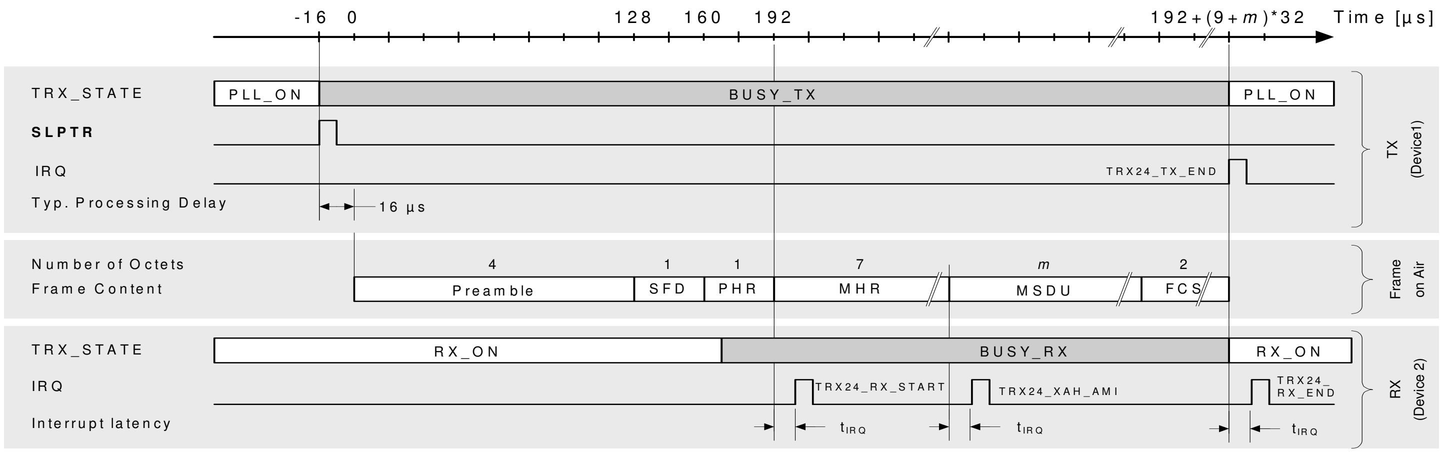
Figure 9-13 on page 39 shows an example for a transmit/receive transaction between two devices and the related interrupt events in Basic Operating Mode. Device 1 transmits a frame containing a MAC header (in this example of length 7), payload and valid FCS. The frame is received by Device 2 which generates the interrupts during the processing of the incoming frame. The received frame is stored in the Frame Buffer.

If the received frame passes the address filter (refer to section "Frame Filtering" on page 54) an address match TRX24_XAH_AMI interrupt is issued after the reception of the MAC header (MHR).

In Basic Operating Mode the TRX24_RX_END interrupt is issued at the end of the received frame. In Extended Operating Mode (refer to "Extended Operating Mode" on page 43) the interrupt is only issued if the received frame passes the address filter and the FCS is valid. Further exceptions are explained in "Extended Operating Mode" on page 43.

Processing delay t_{IRQ} is a typical value (see chapter "Digital Interface Timing Characteristics" on page 507).

Figure 9-13. Timing of TRX24_RX_START, TRX24_XAH_AMI, TRX24_TX_END and TRX24_RX_END Interrupts in Basic Operating Mode



9.4.1.4 Basic Operating Mode Timing

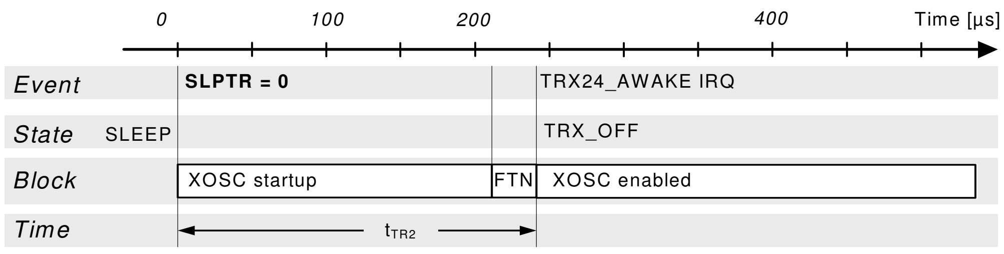
The following paragraphs depict state transitions and their timing properties. Timing figures are explained in Table 9-3 on page 42 and section "Digital Interface Timing Characteristics" on page 507.

9.4.1.4.1 Wake-up Procedure

The wake-up procedure from radio transceiver SLEEP state is shown in Figure 9-14 below. This figure implies, that the microcontroller is already running and hence, the digital voltage regulator is enabled. If the microcontroller clock source is set to Transceiver Clock, the crystal oscillator is also running, which reduces the radio transceiver wake-up time further. For information about the wake-up timing of the microcontroller, depending on the different clock source options, refer to "System Clock and Clock Options" on page 147.

In order to calculate the total wake-up delay from microcontroller sleep mode (see "Power Management and Sleep Modes" on page 156), the microcontroller wake-up time, including the voltage regulator ramp-up and the radio transceiver wake-up time has to be added.

Figure 9-14. Wake-up Procedure from Transceiver SLEEP State



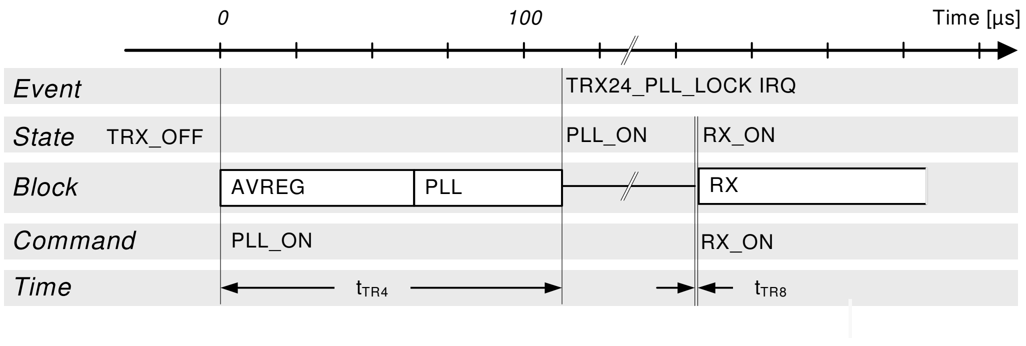
The radio transceiver SLEEP state is left by releasing bit SLPTR to “0”. This restarts the XOSC if it is not already running. After t_{TR2} = 215 µs + 25 µs = 240 µs (see Table 9-3 on page 42) the radio transceiver enters TRX_OFF state. If the XOSC is already running, the radio transceiver enters TRX_OFF state after 25 µs.

During this wake-up procedure the calibration of the filter-tuning network (FTN) is performed. Entering TRX_OFF state is signaled by the TRX24_AWAKE interrupt, if enabled.

9.4.1.4.2 PLL_ON and RX_ON States

The transition from TRX_OFF to PLL_ON and RX_ON mode is shown in Figure 9-15 below.

Figure 9-15. Transition from TRX_OFF to PLL_ON and RX_ON State



Note: 1. If TRX_CMD = RX_ON in TRX_OFF state RX_ON state is entered immediately, even if the PLL has not settled.

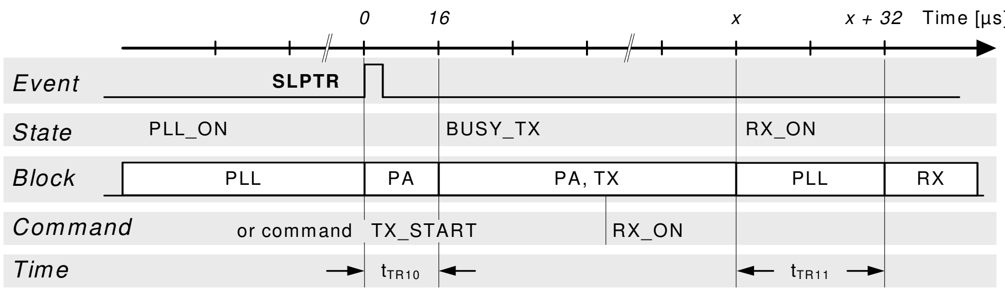
2. If the AVR ADC module is enabled, the AVREG is already started and thus the state transition time t_{TR4} is reduced.

Entering the commands PLL_ON or RX_ON in TRX_OFF state initiates a ramp-up sequence of the internal 1.8V voltage regulator for the analog domain (AVREG), if AVREG is not already enabled by the AVR ADC module. RX_ON state can be entered any time from PLL_ON state regardless whether the PLL has already locked as indicated by the TRX24_PLL_LOCK interrupt.

9.4.1.4.3 BUSY_TX and RX_ON States

The transition from PLL_ON to BUSY_TX state and subsequent to RX_ON state is shown in Figure 9-16 below.

Figure 9-16. PLL_ON to BUSY_TX to RX_ON Timing



Starting from PLL_ON state it is assumed that the PLL is already locked. A transmission is initiated either by writing “1” to bit SLPTR or by command TX_START. The PLL settles to the transmit frequency and the PA is enabled.

t_{TR10} = 16 µs after initiating the transmission, the radio transceiver changes into BUSY_TX state and the internally generated SHR is transmitted. After that the PSDU data are transmitted from the Frame Buffer.

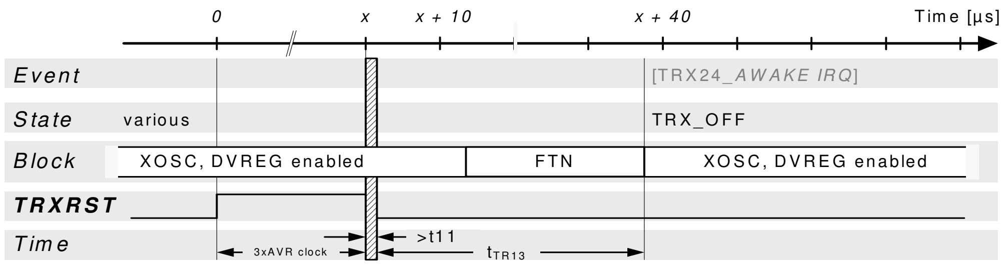
After completing the frame transmission, indicated by the TRX24_TX_END interrupt, the PLL settles back to the receive frequency within t_{TR11} = 32 µs in state PLL_ON.

If during TX_BUSY the radio transmitter is programmed to change to a receive state it automatically proceeds the state change to RX_ON state after finishing the transmission.

9.4.1.4.4 Reset Procedure

The radio transceiver reset procedure is shown in Figure 9-17 below.

Figure 9-17. Reset Procedure



Note: 1. Timing parameter t_{TR13} = 37 µs refers to Table 9-3 on page 42; t_{11} refers to "Digital Interface Timing Characteristics" on page 507.

2. If TRXRST is set during radio transceiver SLEEP state, the XOSC startup delay is extended by the XOSC startup time.

TRXRST = “1” resets all radio transceiver registers to their default values.

The radio transceiver reset is released automatically after 3 AVR clock cycles and the wake-up sequence without restarting XOSC and DVREG, nevertheless an FTN calibration cycle is performed, refer to "Automatic Filter Tuning (FTN)" on page 83. After that the TRX_OFF state is entered.

Figure 9-17 above illustrates the radio transceiver reset procedure if the radio transceiver is in any state but not in SLEEP state.

If the radio transceiver was in SLEEP state, the SLPTR bit in the TRXPR register must be cleared prior to clearing the TRXRST bit in order to enter the TRX_OFF state. Otherwise the radio transceiver enters the SLEEP state immediately.

If the radio transceiver was in SLEEP state and the Transceiver Clock is not selected as the microcontroller clock source, the XOSC is enabled before entering TRX_OFF state.

If register TRX_STATUS indicates STATE_TRANSITION_IN_PROGRESS during system initialization until the radio transceiver reaches TRX_OFF, do not try to initiate a further state change while the radio transceiver is in this state.

Note that before accessing the radio transceiver module the TRX24_AWAKE event should be checked.

9.4.1.4.5 State Transition Timing Summary

The transition numbers correspond to Table 9-3 below. See measurement setup in "Basic Application Schematic" on page 493.

Table 9-3. Radio Transceiver State Transition Timing

No	Symbol	Transition			Time [µs], (typ)	Comments
1	t_{TR2}	SLEEP	⇨	TRX_OFF	240	Depends on crystal oscillator setup (CL = 10 pf) TRX_OFF state indicated by TRX24_AWAKE interrupt
2	t_{TR3}	TRX_OFF	⇨	SLEEP	$35 \cdot 1 / f_{CLKM}$	For f_{CLKM} > 250 kHz
3	t_{TR4}	TRX_OFF	⇨	PLL_ON	110	Depends on external capacitor at AVDD (1 µF nom)
4	t_{TR5}	PLL_ON	⇨	TRX_OFF	1	
5	t_{TR6}	TRX_OFF	⇨	RX_ON	110	Depends on external capacitor at AVDD (1 µF nom)
6	t_{TR7}	RX_ON	⇨	TRX_OFF	1	
7	t_{TR8}	PLL_ON	⇨	RX_ON	1	
8	t_{TR9}	RX_ON	⇨	PLL_ON	1	Transition time is also valid for TX_ARET_ON, RX_AACK_ON
9	t_{TR10}	PLL_ON	⇨	BUSY_TX	16	When setting bit SLPTR or TRX_CMD = TX_START, the first symbol transmission is delayed by 16 µs (PLL settling and PA ramp up).
10	t_{TR11}	BUSY_TX	⇨	PLL_ON	32	PLL settling time from TX_BUSY to PLL_ON state
11	t_{TR12}	All modes	⇨	TRX_OFF	1	Using TRX_CMD = FORCE_TRX_OFF (see register TRX_STATE), Not valid for SLEEP state
12	t_{TR13}	RESET	⇨	TRX_OFF	37	Not valid for SLEEP state
13	t_{TR14}	Various states	⇨	PLL_ON	1	Using TRX_CMD = FORCE_PLL_ON (see register TRX_STATE), Not valid for SLEEP, RESET and TRX_OFF

The state transition timing is calculated based on the timing of the individual blocks shown in Table 9-8 on page 51. The worst case values include maximum operating temperature, minimum supply voltage, and device parameter variations.

Table 9-8. Analog Block Initialization and Settling Time

No	Symbol	Block	Time [µs], (typ)	Time [µs], (max)	Comments
15	t_{TR15}	XOSC	215	1000	Leaving SLEEP state, depends on crystal Q factor and load capacitor
16	t_{TR16}	FTN		25	FTN tuning time, fixed
17	t_{TR17}	DVREG	60	1000	Depends on external bypass capacitor at DVDD (CB3 = 1 µF nom., 10 µF worst case), depends on V_{DEVDD}
18	t_{TR18}	AVREG	60	1000	Depends on external bypass capacitor at AVDD (CB1 = 1 µF nom., 10 µF worst case) , depends on V_{EVDD}
19	t_{TR19}	PLL, initial	110	155	PLL settling time TRX_OFF ⇨ PLL_ON, including 60 µs AVREG settling time
20	t_{TR20}	PLL, settling	11	24	Settling time between channel switch
21	t_{TR21}	PLL, CF cal	35		PLL center frequency calibration, refer to "Calibration Loops" on page 83
22	t_{TR22}	PLL, DCU cal	6		PLL DCU calibration, refer to "Calibration Loops" on page 83
23	t_{TR23}	PLL, RX ⇨ TX	16		Maximum PLL settling time RX ⇨ TX

No	Symbol	Block	Time [µs], (typ)	Time [µs], (max)	Comments
24	t_{TR24}	PLL, TX ⇨ RX	32		Maximum PLL settling time TX ⇨ RX
25	t_{TR25}	RSSI, update	2		RSSI update period in receive states, refer to "Reading RSSI" on page 68
26	t_{TR26}	ED	140		ED measurement period, refer to "Measurement Description" on page 69
27	t_{TR27}	SHR, sync	96		Typical SHR synchronization period, refer to "Measurement Description" on page 69
28	t_{TR28}	CCA	140		CCA measurement period, refer to "Configuration and CCA Request" on page 71
29	t_{TR29}	Random value	1		Random value update period, refer to "Random Number Generator" on page 85

9.4.2 Extended Operating Mode

The Extended Operating Mode is a hardware MAC accelerator and goes beyond the basic radio transceiver functionality provided by the Basic Operating Mode. It handles time critical MAC tasks requested by the IEEE 802.15.4 standard or by hardware such as automatic acknowledgement, automatic CSMA-CA and retransmission. This results in a more efficient IEEE 802.15.4 software MAC implementation including reduced code size and may allow operating at lower microcontroller clock rates.

The Extended Operating Mode is designed to support IEEE 802.15.4-2006 compliant frames; the mode is backward compatible to IEEE 802.15.4-2003 and supports non IEEE 802.15.4 compliant frames. This mode comprises the following procedures:

Automatic acknowledgement (RX_AACK) divides into the tasks:

- Frame reception and automatic FCS check;
- Configurable addressing fields check;
- Interrupt indicating address match;
- Interrupt indicating frame reception, if it passes address filtering and FCS check;
- Automatic ACK frame transmission (if the received frame passed the address filter and FCS check and if an ACK is required by the frame type and ACK request);
- Support of slotted acknowledgment using SLPTR bit for frame start.

Automatic CSMA-CA and Retransmission (TX_ARET) divides into the tasks:

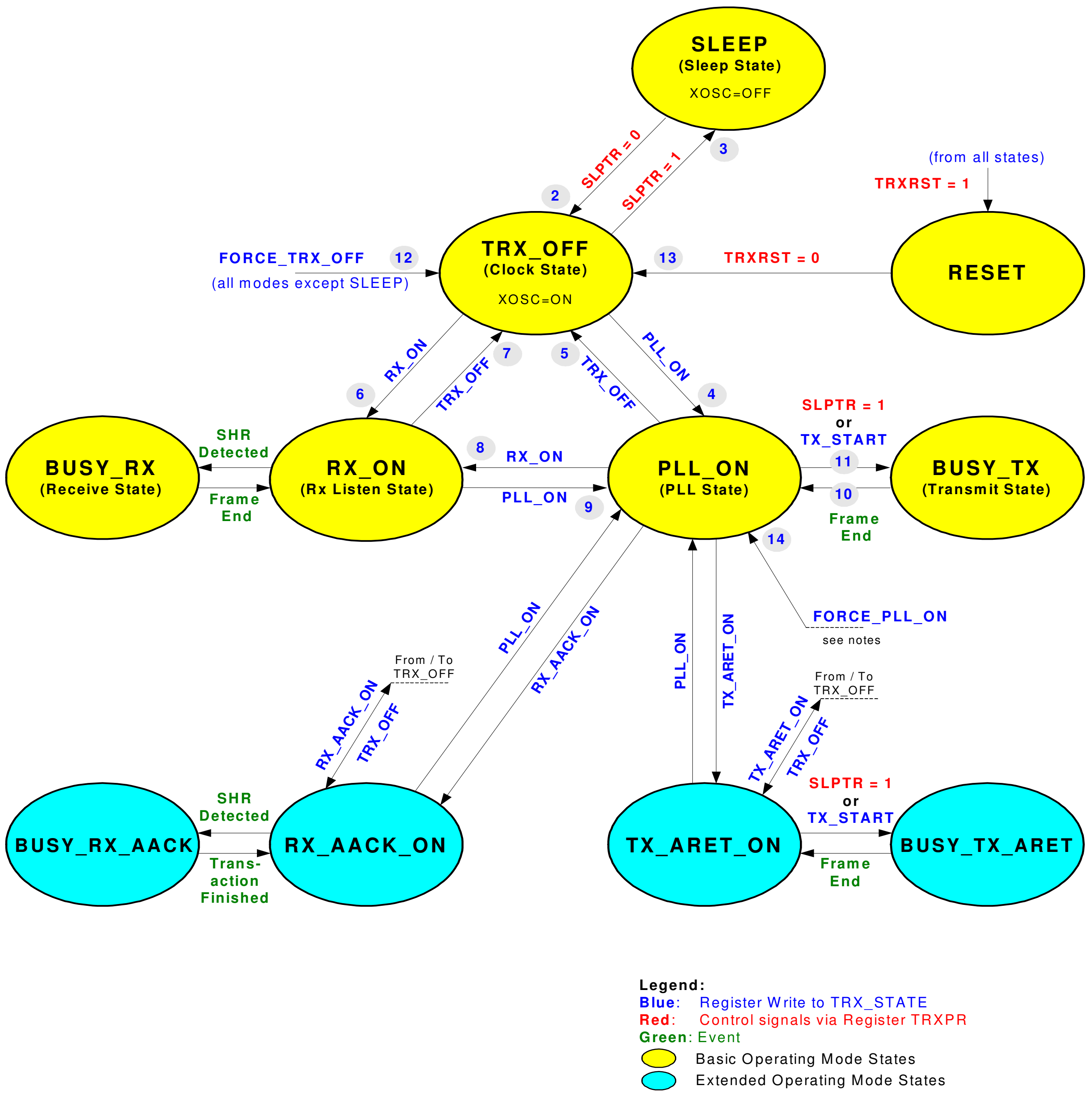
- CSMA-CA including automatic CCA retry and random back-off;
- Frame transmission and automatic FCS field generation;
- Reception of ACK frame (if an ACK was requested);
- Automatic frame retry if ACK was expected but not received;
- Interrupt signaling with transaction status.

Automatic FCS check and generation (refer to "Frame Check Sequence (FCS)" on page 66) is used by the RX_AACK and TX_ARET modes. In RX_AACK mode an automatic FCS check is always performed for incoming frames.

An ACK received in TX_ARET mode within the time required by IEEE 802.15.4 is accepted if the FCS is valid and if the sequence number of the ACK matches the sequence number of the previously transmitted frame. Dependent on the value of the frame pending subfield in the received acknowledgement frame the transaction status is set according to Table 9-16 on page 58.

The state diagram including the Extended Operating Mode states is shown in Figure 9-18 below. Yellow marked states represent the Basic Operating Mode; blue marked states represent the Extended Operating Mode.

Figure 9-18. Extended Operating Mode State Diagram



Note: 1. State transition numbers correspond to Table 9-3 on page 42.

9.4.2.1 State Control

The Extended Operating Mode states RX_AACK and TX_ARET are controlled via the bits TRX_CMD of register TRX_STATE, which receives the state transition commands. The states are entered from TRX_OFF or PLL_ON state as illustrated in Figure 9-18 on page 44. The completion of each state change command shall always be confirmed by reading the TRX_STATUS register.

RX_AACK - Receive with Automatic ACK

A state transition to RX_AACK_ON from PLL_ON or TRX_OFF is initiated by writing the command RX_AACK_ON to the register bits TRX_CMD. The state change can be confirmed by reading register TRX_STATUS, those changes to RX_AACK_ON or BUSY_RX_AACK on success. BUSY_RX_AACK is returned if a frame is currently being received.

The RX_AACK state is left by writing command TRX_OFF or PLL_ON to the register bits TRX_CMD. If the radio transceiver is within a frame receive or acknowledgment procedure (BUSY_RX_AACK) the state change is executed after finish. Alternatively, the commands FORCE_TRX_OFF or FORCE_PLL_ON can be used to cancel the RX_AACK transaction and change into radio transceiver state TRX_OFF or PLL_ON respectively.

TX_ARET - Transmit with Automatic Retry and CSMA-CA Retry

Similarly, a state transition to TX_ARET_ON from PLL_ON or TRX_OFF is initiated by writing command TX_ARET_ON to register bits TRX_CMD. The radio transceiver is in the TX_ARET_ON state after TRX_STATUS register changes to TX_ARET_ON. The TX_ARET transaction is started with writing ‘1’ to the SLPTR bit of the TRXPR register or writing the command TX_START to register bits TRX_CMD.

TX_ARET state is left by writing the command TRX_OFF or PLL_ON to the register bits TRX_CMD. If the radio transceiver is within a CSMA-CA, a frame-transmit or an acknowledgment procedure (BUSY_TX_ARET) the state change is executed after finish. Alternatively, the command FORCE_TRX_OFF or FORCE_PLL_ON can be used to instantly terminate the TX_ARET transaction and change into radio transceiver states TRX_OFF or PLL_ON, respectively.

Note that a state change request from TRX_OFF to RX_AACK_ON or TX_ARET_ON internally passes the state PLL_ON to initiate the radio transceiver. Thus the readiness to receive or transmit data is delayed accordingly. It is recommended to use interrupt TRX24_PLL_LOCK as an indicator.

9.4.2.2 Configuration

The use of the Extended Operating Mode is based on Basic Operating Mode functionality. Only features beyond the basic radio transceiver functionality are described in the following sections. For details on the Basic Operating Mode refer to section "Basic Operating Mode" on page 35.

When using the RX_AACK or TX_ARET modes, the following registers needs to be configured.

RX_AACK configuration steps:

- Short address, PAN-ID and IEEE address (register SHORT_AADR_0, SHORT_ADDR_1, PAN_ID_0, PAN_ID_1, IEEE_ADDR_0 … IEEE_ADDR_7)
- Configure RX_AACK properties (register XAH_CTRL_0, CSMA_SEED_1)
    - Handling of Frame Version Subfield

- Handling of Pending Data Indicator
    - Characterize as PAN coordinator
    - Handling of Slotted Acknowledgement
- Additional Frame Filtering Properties (register XAH_CTRL_1, CSMA_SEED_1)
    - Promiscuous Mode
    - Enable or disable automatic ACK generation
    - Handling of reserved frame types

The addresses for the address match algorithm are to be stored in the appropriate address registers. Additional control of the RX_AACK mode is done with registers XAH_CTRL_1 and CSMA_SEED_1.

As long as a short address has not been set, only broadcast frames and frames matching the IEEE address can be received.

Configuration examples for different device operating modes and handling of various frame types can be found in section "Description of RX_AACK Configuration Bits" on page 49.

TX_ARET configuration steps:

- Leave register bit TX_AUTO_CRC_ON = 1   register TRX_CTRL_1
- Configure CSMA-CA

○ MAX_FRAME_RETRIES	register XAH_CTRL_0
○ MAX_CSMA_RETRIES	register XAH_CTRL_0
○ CSMA_SEED	registersCSMA_SEED_0, CSMA_SEED_1
○ MAX_BE, MIN_BE	register CSMA_BE

- Configure CCA (see section "Configuration and CCA Request" on page 71)

MAX_FRAME_RETRIES (register XAH_CTRL_0) defines the maximum number of frame retransmissions.

The register bits MAX_CSMA_RETRIES (register XAH_CTRL_0) configure the number of CSMA-CA retries after a busy channel is detected.

The CSMA_SEED_0 and CSMA_SEED_1 registers define a random seed for the back-off-time random-number generator of the radio transceiver.

The MAX_BE and MIN_BE register bits (register CSMA_BE) set the maximum and minimum CSMA back-off exponent (according to [1] on page 100).

9.4.2.3 RX_AACK_ON – Receive with Automatic ACK

The general functionality of the RX_AACK procedure is shown in Figure 9-19 on page 48.

The gray shaded area is the standard flow of a RX_AACK transaction for IEEE 802.15.4 compliant frames (refer to section "Configuration of IEEE Scenarios" on page 50). All other procedures are exceptions for specific operating modes or frame formats (refer to section "Configuration of non IEEE 802.15.4 Compliant Scenarios" on page 52).

The frame filtering operation is described in detail in section "Frame Filtering" on page 54.

In RX_AACK_ON state, the radio transceiver listens for incoming frames. After detecting SHR and a valid PHR, the radio transceiver parses the frame content of the MAC header (MHR) as described in section "PHY Header (PHR)" on page 61.

Generally, at nodes, configured as a normal device or PAN coordinator, a frame is not indicated if the frame filter does not match and the FCS is invalid. Otherwise, the TRX_24_RX_END interrupt is issued after the completion of the frame reception. The microcontroller can then read the frame. An exception applies if promiscuous mode is enabled (see section "Configuration of IEEE Scenarios" on page 50). In that case a TRX_24_RX_END interrupt is issued even if the FCS fails.

If the content of the MAC addressing fields of the received frame (refer to IEEE 802.15.4 section 7.2.1) matches one of the configured addresses, dependent on the addressing mode, an address match interrupt (TRX24_XAH_AMI) is issued (refer to section "Frame Filtering" on page 54). The expected address values are to be stored in registers Short-address, PAN-ID and IEEE-address. Frame filtering as described in section "Frame Filtering" on page 54 is also valid for Basic Operating Mode.

During reception the radio transceiver parses bit[5] (ACK Request) of the frame control field of the received data or the MAC command frame to check if an ACK reply is expected. In that case and if the frame passes the third level of filtering (see IEEE 802.15.4-2006, section 7.5.6.2), the radio transceiver automatically generates and transmits an ACK frame. After the ACK transmission is finished, a TRX24_TX_END interrupt is generated.

The content of the frame pending subfield of the ACK response is set by bit AACK_SET_PD of register CSMA_SEED_1 when the ACK frame is sent in response to a data request MAC command frame, otherwise this subfield is set to “0”. The sequence number is copied from the received frame.

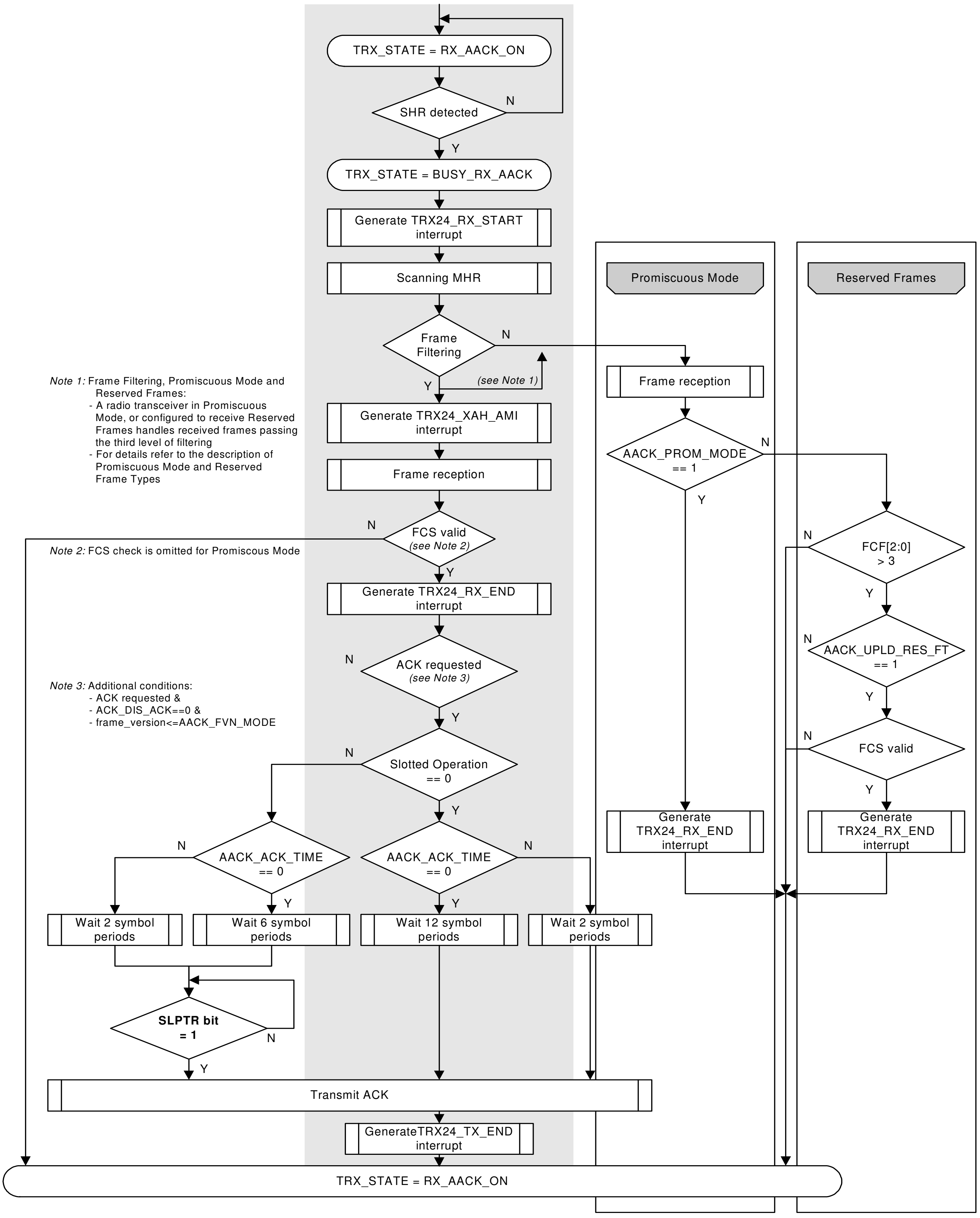
Optionally, the start of the transmission of the acknowledgement frame can be influenced by register bit AACK_ACK_TIME. Default value (according to standard IEEE 802.15.4, page 54) is 12 symbol times after the reception of the last symbol of a data or MAC command frame.

If the bit AACK_DIS_ACK of register CSMA_SEED_1 is set, no acknowledgement frame is sent even if an acknowledgment frame was requested. This is useful for operating the MAC hardware accelerator in promiscuous mode (see section "Configuration of non IEEE 802.15.4 Compliant Scenarios" on page 52).

The status of the RX_AACK operation is indicated by the bits TRAC_STATUS of register TRAC_STATUS.

During the operations described above the radio transceiver remains in BUSY_RX_AACK state.

Figure 9-19. Flow Diagram of RX_AACK



Note 1: Frame Filtering, Promiscuous Mode and Reserved Frames:
- A radio transceiver in Promiscuous Mode, or configured to receive Reserved Frames handles received frames passing the third level of filtering
- For details refer to the description of Promiscuous Mode and Reserved Frame Types

Note 2: FCS check is omitted for Promiscuous Mode

Note 3: Additional conditions:
- ACK requested &
- ACK_DIS_ACK==0 &
- frame_version<=AACK_FVN_MODE

9.4.2.3.1 Description of RX_AACK Configuration Bits

Overview

The following table summarizes all register bits which affect the behavior of a RX_AACK transaction. For address filtering it is further required to setup address registers to match to the expected address.

Configuration and address bits are to be set in TRX_OFF or PLL_ON state prior to switching to RX_AACK mode.

A graphical representation of various operating modes is illustrated in Figure 9-19 on page 48.

Table 9-5. Overview of RX_AACK Configuration Bits

Register Name	Register Bits	Description
SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7		Set node addresses
RX_SAFE_MODE	7	Protect buffer after frame receive
AACK_PROM_MODE	1	Support promiscuous mode
AACK_ACK_TIME	2	Change auto acknowledge start time
AACK_UPLD_RES_FT	4	Enable reserved frame type reception, needed to receive non-standard compliant frames
AACK_FLTR_RES_FT	5	Filter reserved frame types like data frame type, needed for filtering of non-standard compliant frames
SLOTTED_OPERATION	0	If set, acknowledgment transmission has to be triggered by register bit SLPTR
AACK_I_AM_COORD	3	If set, the device is a PAN coordinator
AACK_DIS_ACK	4	Disable generation of acknowledgment
AACK_SET_PD	5	Set frame pending subfield in Frame Control Field (FCF), refer to section "Overview" on page 66
AACK_FVN_MODE	7:6	Controls the ACK behavior, depending on FCF frame version number

The usage of the RX_AACK configuration bits for various operating modes of a node is explained in the following sections. Configuration bits not mentioned in the following two sections should be set to their reset values.

All registers mentioned in Table 9-5 above are described in section "Register Summary" on page 60.

Note, that the general behavior of the Extended Feature Set settings:

• OQPSK_DATA_RATE	(PSDU data rate)
• SFD_VALUE	(alternative SFD value)
• ANT_DIV	(Antenna Diversity)
• RX_PDT_LEVEL	(blocking frame reception of lower power signals)

are completely independent from RX_AACK mode (see "Radio Transceiver Extended Feature Set" on page 85). Each of these operating modes can be combined with the RX_AACK mode.

9.4.2.3.2 Configuration of IEEE Scenarios

Normal Device

The Table 9-6 below shows a typical RX_AACK configuration of an IEEE 802.15.4 device operated as a normal device rather than a PAN coordinator or router.

Table 9-6. Configuration of IEEE 802.15.4 Devices

Register Name	Register Bits	Description
SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7		Set node addresses
RX_SAFE_MODE	7	*0*: disable frame protection *1*: enable frame protection
SLOTTED_OPERATION	0	*0*: if transceiver works in unslotted mode *1*: if transceiver works in slotted mode
AACK_FVN_MODE	7:6	Controls the ACK behavior, depending on FCF frame version number *0x00* : acknowledges only frames with version number 0, i.e. according to IEEE 802.15.4-2003 frames *0x01* : acknowledges only frames with version number 0 or 1, i.e. frames according to IEEE 802.15.4-2006 *0x10* : acknowledges only frames with version number 0 or 1 or 2 *0x11* : acknowledges all frames, independent of the FCF frame version number

Notes: 1. If no short address has been configured before the device has been assigned one by the PAN-coordinator, only frames directed to either the broadcast address or the IEEE address are received.

2. In IEEE 802.15.4-2003 standard the frame version subfield did not yet exist but was marked as reserved. According to this standard, reserved fields have to be set to zero. On the other hand, IEEE 802.15.4-2003 standard requires ignoring reserved bits upon reception. Thus, there is a contradiction in the standard which can be interpreted in two ways:

a) If a network should only allow access to nodes which use the IEEE 802.15.4-2003, then AACK_FVN_MODE should be set to 0.

b) If a device should acknowledge all frames independent of its frame version, AACK_FVN_MODE should be set to 3. However, this can result in conflicts with co-existing IEEE 802.15.4-2006 standard compliant networks.

The same holds for PAN coordinators as described below.

PAN-Coordinator

Table 9-7 on page 51 shows the RX_AACK configuration for a PAN coordinator.

Table 9-7. Configuration of a PAN Coordinator

Register Name	Register Bits	Description
SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7		Set node addresses
RX_SAFE_MODE	7	*0*: disable frame protection *1*: enable frame protection
SLOTTED_OPERATION	0	*0*: if transceiver works in unslotted mode *1*: if transceiver works in slotted mode
AACK_I_AM_COORD	3	*1*: device is PAN coordinator
AACK_SET_PD	5	*0*: frame pending subfield is not set in FCF *1*: frame pending subfield is set in FCF
AACK_FVN_MODE	7:6	Controls the ACK behavior, depends on FCF frame version number *0x00* : acknowledges only frames with version number 0, i.e. according to IEEE 802.15.4-2003 frames *0x01* : acknowledges only frames with version number 0 or 1, i.e. frames according to IEEE 802.15.4-2006 *0x10* : acknowledges only frames with version number 0 or 1 or 2 *0x11* : acknowledges all frames, independent of the FCF frame version number

Promiscuous Mode

The promiscuous mode is described in IEEE 802.15.4-2006, section 7.5.6.5. This mode is further illustrated in Radio Transceiver Extended Feature Set on page 85. According to IEEE 802.15.4-2006 when in promiscuous mode, the MAC sub layer shall pass received frames with correct FCS to the next higher layer without further processing. That implies that frames should never be acknowledged.

Only second level filter rules as defined by IEEE 802.15.4-2006, section 7.5.6.2, are applied to the received frame.

Table 9-8 below shows the typical configuration of a device operating in promiscuous mode.

Table 9-8. Configuration of Promiscuous Mode

Register Name	Register Bits	Description
SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7		Each address shall be set: 0x00
AACK_PROM_MODE	1	*1*: Enable promiscuous mode
AACK_DIS_ACK	4	*1*: Disable generation of acknowledgment

Register Name	Register Bits	Description
AACK_FVN_MODE	7:6	Controls the ACK behavior, depends on FCF frame version number *0x00* : acknowledges only frames with version number 0, i.e. according to IEEE 802.15.4-2003 frames *0x01* : acknowledges only frames with version number 0 or 1, i.e. frames according to IEEE 802.15.4-2006 *0x10* : acknowledges only frames with version number 0 or 1 or 2 *0x11* : acknowledges all frames, independent of the FCF frame version number

Second level of filtering according to IEEE 802.15.4-2006, section 7.5.6.2, is applied to a received frame if the radio transceiver is in promiscuous mode. However, a TRX24_RX_END interrupt is issued even if the FCS is invalid. Thus it is necessary to read bit RX_CRC_VALID of register PHY_RSSI after the TRX24_RX_END interrupt in order to verify the reception of a frame with a valid FCS.

If a device, operating in promiscuous mode, receives a frame with a valid FCS which in addition passed the third level filtering according to IEEE 802.15.4-2006, section 7.5.6.2, an acknowledgement frame would be transmitted. According to the definition of the promiscuous mode a received frame shall not be acknowledged even if it is requested. Thus bit AACK_DIS_ACK of register CSMA_SEED_1 has to be set to 1.

In all receive modes a TRX24_AMI interrupt is issued, when the received frame matches the node's address according to the filter rules described in section "Frame Filtering" on page 54.

Alternatively, in RX_ON state of the Basic Operating Mode when a valid PHR is detected a TRX24_RX_START interrupt is generated and the frame is received. The end of the frame reception is signalized with a TRX24_RX_END interrupt. At the same time the bit RX_CRC_VALID of register PHY_RSSI is updated with the result of the FCS check (see "Overview" on page 66). The RX_CRC_VALID bit must be checked in order to dismiss corrupted frames according to the definition of the promiscuous mode.

9.4.2.3.3 Configuration of non IEEE 802.15.4 Compliant Scenarios

Sniffer

Table 9-9 below shows a RX_AACK configuration to setup a sniffer device. Other RX_AACK configuration bits should be set to their reset values (see Table 9-5 on page 49). All frames received are indicated by a TRX24_RX_START and TRX24_RX_END interrupt. Bit RX_CRC_VALID of register PHY_RSSI is updated after frame reception with the result of the FCS check (see "Overview" on page 66). The RX_CRC_VALID bit needs to be checked in order to dismiss corrupted frames.

Table 9-9. Configuration of a Sniffer Device

Register Name	Register Bits	Description
AACK_PROM_MODE	1	*1*: Enable promiscuous mode
AACK_DIS_ACK	4	*1*: Disable generation of acknowledgment

This operating mode is similar to the promiscuous mode.

Reception of Reserved Frames

Frames with reserved frame types (see section Table 9-16 on page 63) can also be handled in RX_AACK mode. This might be required when implementing proprietary, non-standard compliant protocols. It is an extension of the address filtering in RX_AACK mode. Received frames are either handled similar to data frames or may be allowed to completely bypass the address filter.

Table 9-10 below shows the required configuration for a node to receive reserved frames and Figure 9-19 on page 48 shows the corresponding flow chart.

Table 9-10. RX_AACK Configuration to Receive Reserved Frame Types

Register Name	Register Bits	Description
SHORT_ADDR_0/1 PAN_ADDR_0/1 IEEE_ADDR_0 … IEEE_ADDR_7		Set node addresses
RX_SAFE_MODE	7	*0*: disable frame protection *1*: enable frame protection
AACK_UPLD_RES_FT	4	*1* : Enable reserved frame type reception
AACK_FLTR_RES_FT	5	Filter reserved frame types like data frame type, see note below *0* : disable *1* : enable
SLOTTED_OPERATION	0	*0*: if transceiver works in un-slotted mode *1*: if transceiver works in slotted mode
AACK_I_AM_COORD	3	*0*: device is not PAN coordinator *1*: device is PAN coordinator
AACK_DIS_ACK	4	*0*: Enable generation of acknowledgment *1*: Disable generation of acknowledgment
AACK_FVN_MODE	7:6	Controls the ACK behavior, depends on FCF frame version number *0x00* : acknowledges only frames with version number 0, i.e. according to IEEE 802.15.4-2003 frames *0x01* : acknowledges only frames with version number 0 or 1, i.e. frames according to IEEE 802.15.4-2006 *0x10* : acknowledges only frames with version number 0 or 1 or 2 *0x11* : acknowledges all frames, independent of the FCF frame version number

There are two different options for handling reserved frame types.

1. AACK_UPLD_RES_FT = 1, AACK_FLT_RES_FT = 0:

   Any non-corrupted frame with a reserved frame type is indicated by a TRX24_RX_END interrupt. No further address filtering is applied on those frames. A TRX24_AMI interrupt is never generated and the acknowledgment subfield is ignored.

2. AACK_UPLD_RES_FT = 1, AACK_FLT_RES_FT = 1:

If AACK_FLT_RES_FT = 1 any frame with a reserved frame type is filtered by the address filter similar to a data frame as described in the standard. Consequently, a TRX24_AMI interrupt is generated upon address match. A TRX24_RX_END interrupt is only generated if the address matched and the frame was not corrupted. An acknowledgment is only send, when the ACK request subfield was set in the received frame and a TRX24_RX_END interrupt occurred.

Note that It is not allowed to set AACK_FLTR_RES_FT = 1 and have register bit AACK_FLTR_RES_FT set to 0.

Short Acknowledgment Frame (ACK) Start Timing

The bit AACK_ACK_TIME of register XAH_CTRL_1 defines the symbol time between frame reception and transmission of an acknowledgment frame.

Table 9-11. Overview of RX_AACK Configuration Bits

Register Name	Register Bit	Description
AACK_ACK_TIME	2	*0*: Standard compliant acknowledgement timing of 12 symbol periods. In slotted acknowledgement operation mode, the acknowledgment frame transmission can be triggered 6 symbol periods after reception of the frame earliest. *1*: Reduced acknowledgment timing of 2 symbol periods (32 µs).

Note that this feature can be used in all scenarios, independent of other configurations. However, shorter acknowledgment timing is especially useful when using High Data Rate Modes to increase battery lifetime and to improve the overall data throughput; see "High Data Rate Modes" on page 86 for details.

9.4.2.4 Frame Filtering

Frame Filtering is an evaluation whether or not a received frame is dedicated for this node. To accept a received frame and to generate an address match interrupt (TRX24_AMI) a filtering procedure as described in IEEE 802.15.4-2006 chapter 7.5.6.2. (Third level of filtering) is applied to the frame. The radio transceiver's RX_AACK mode accepts only frames that satisfy all of the following requirements (quote from IEEE 802.15.4-2006, 7.5.6.2):

1. The Frame Type subfield shall not contain a reserved frame type.
2. The Frame Version subfield shall not contain a reserved value.
3. If a destination PAN identifier is included in the frame, it shall match macPANId or shall be the broadcast PAN identifier (0xFFFF).
4. If a short destination address is included in the frame, it shall match either macShortAddress or the broadcast address (0xFFFF). Otherwise, if an extended destination address is included in the frame, it shall match aExtendedAddress.
5. If the frame type indicates that the frame is a beacon frame, the source PAN identifier shall match macPANId unless macPANId is equal to 0xFFFF, in which case the beacon frame shall be accepted regardless of the source PAN identifier.
6. If only source addressing fields are included in a data or MAC command frame, the frame shall be accepted only if the device is the PAN coordinator and the source PAN identifier matches macPANId.

The radio transceiver requires two additional rules:

1. The frame type indicates that the frame is not an ACK frame (refer toTable 9-6 on page 50).

2. At least one address field must be configured.

Address match, indicated by the TRX24_AMI interrupt is further controlled by the content of subfields of the frame control field of a received frame according to the following rule:

If (Destination Addressing Mode = 0 OR 1) AND (Source Addressing Mode = 0) no TRX24_AMI interrupt is generated, refer to Figure 9-26 on page 63. This effectively causes all acknowledgement frames not to be announced which otherwise always pass the filter regardless of whether they are intended for this device or not.

For backward compatibility to IEEE 802.15.4-2003 third level filter rule 2 (Frame Version) can be disabled by the bits AACK_FVN_MODE of register CSMA_SEED_1.

Frame filtering is available in Extended and Basic Operating Mode (see section "Basic Operating Mode" on page 35); a frame passing the frame filtering generates an TRX24_AMI interrupt, if enabled.

Note:
1. Filter rule 1 is affected by register bits AACK_FLTR_RES_FT and AACK_UPLD_RES_FT (see register "XAH_CTRL_1 – Transceiver Acknowledgment Frame Control Register 1" on page 119).
2. Filter rule 2 is affected by register bits AACK_FVN_MODE (see register "CSMA_SEED_1 – Transceiver Acknowledgment Frame Control Register 2" on page 129).

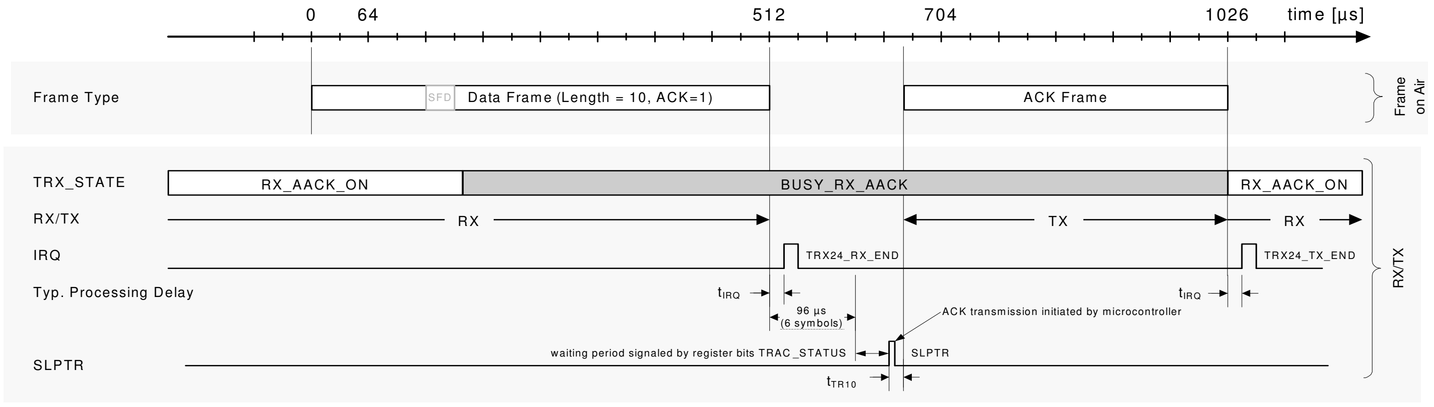
9.4.2.4.1 RX_AACK Slotted Operation – Slotted Acknowledgement

The radio transceiver supports slotted acknowledgement operation according to IEEE 802.15.4-2006, section 5.5.4.1.

In RX_AACK mode with bit SLOTTED_OPERATION of register XAH_CTRL_0 set, the transmission of an acknowledgement frame has to be controlled by the microcontroller. If an ACK frame has to be transmitted the radio transceiver expects writing SLPTR=1 to actually start the transmission. This waiting state is signaled 6 symbol periods after the reception of the last symbol of a data or MAC command frame by bits TRAC_STATUS of register XAH_CTRL_0, which are set to SUCCESS_WAIT_FOR_ACK in that case. In networks using slotted operation the start of the acknowledgment frame and thus the exact timing must be provided by the microcontroller.

A timing example of an RX_AACK transaction with bit SLOTTED_OPERATION of register XAH_CTRL_0 set is shown in the next figure. The acknowledgement frame is ready to transmit 6 symbol times after the reception of the last symbol of a data or MAC command frame. The transmission of the acknowledgement frame is initiated by the microcontroller by writing SLPTR=1 and starts 16µs (t_{TR10}) later. The interrupt latency t_{IRQ} is specified in section "Digital Interface Timing Characteristics" on page 507.

Figure 9-10. Example Timing of an RX_AACK Transaction for Slotted Operation

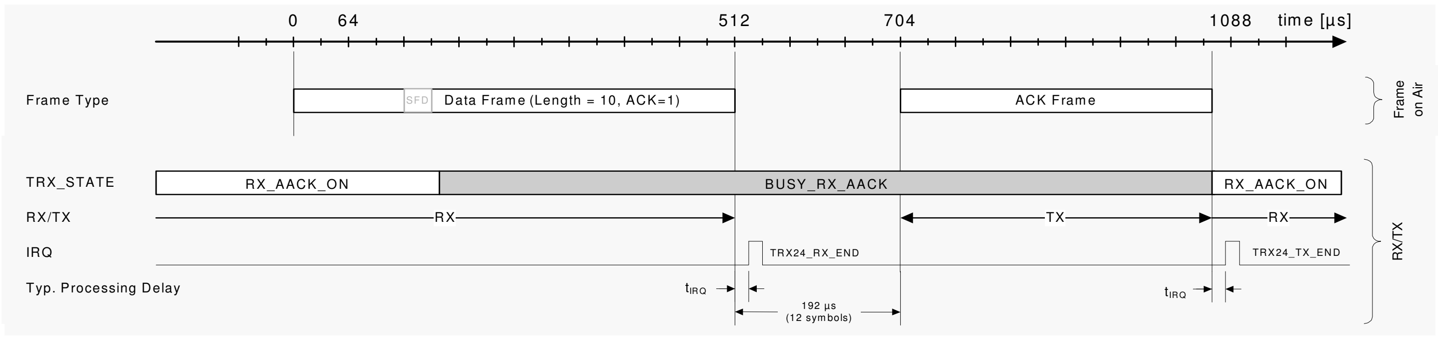


If bit AACK_ACK_TIME of register XAH_CTRL_1 is set, an acknowledgment frame can be sent already 2 symbol times after the reception of the last symbol of a data or MAC command frame.

9.4.2.4.2 RX_AACK Mode Timing

A timing example of an RX_AACK transaction is shown in the next figure. In this example a data frame of length 10 with an ACK request is received. The radio transceiver changes to state BUSY_RX_AACK after SFD detection. The completion of the frame reception is indicated by a TRX24_RX_END interrupt. Interrupts TRX24_RX_START and TRX24_AMI are disabled in this example. The ACK frame is automatically transmitted after a default wait period of 12 symbols (192 µs), bit AACK_ACK_TIME = 0 (reset value). The interrupt latency t_{IRQ} is specified in section "Digital Interface Timing Characteristics" on page 507.

Figure 9-11. Example Timing of an RX_AACK Transaction



If bit AACK_ACK_TIME of register XAH_CTRL_1 is set, an acknowledgment frame is sent already 2 symbol times after the reception of the last symbol of a data or MAC command frame.

9.4.2.5 TX_ARET_ON – Transmit with Automatic Retry and CSMA-CA Retry

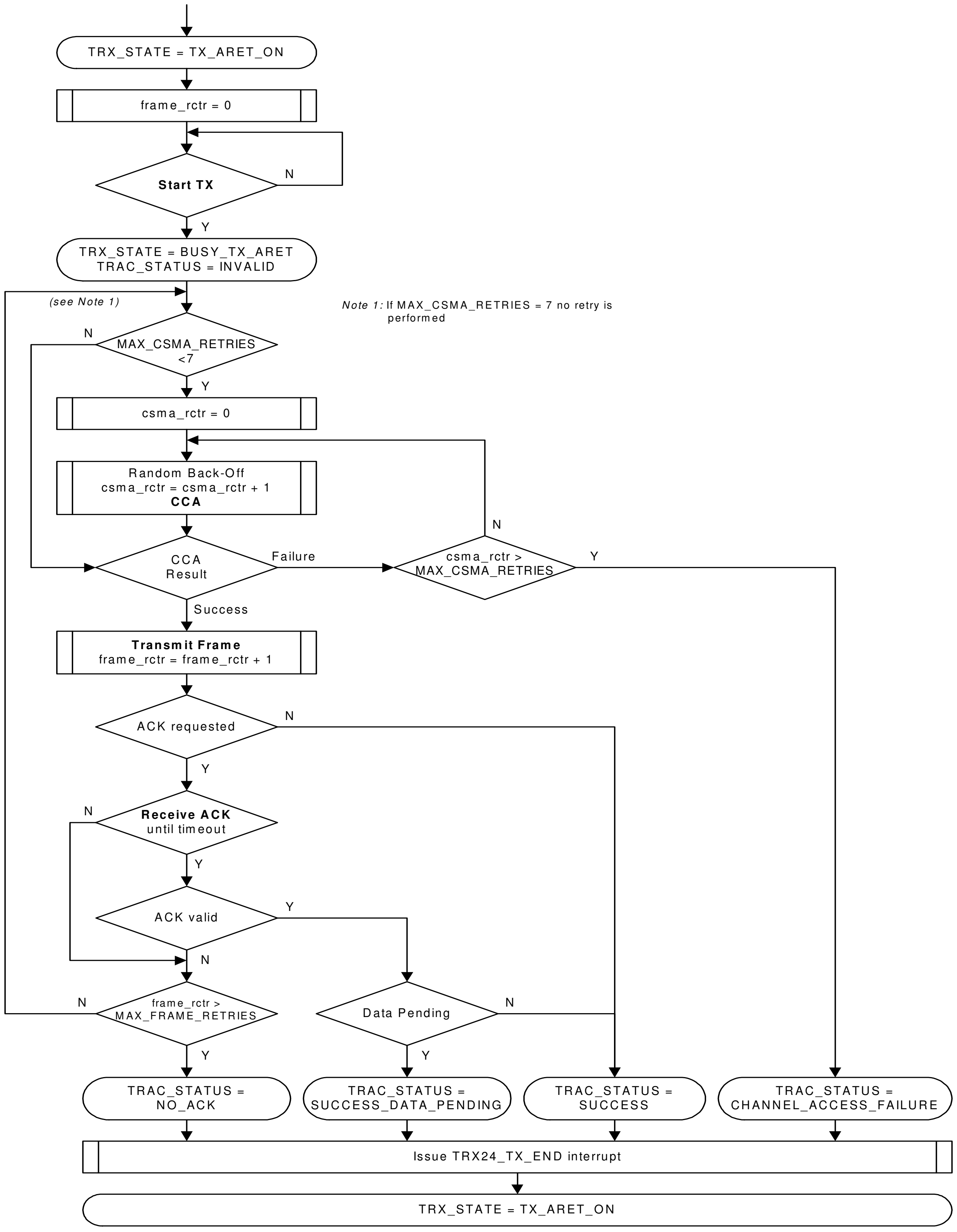


Figure 9-12. Flow Diagram of TX_ARET

Overview

The implemented TX_ARET algorithm is shown in Figure 9-12 on page 57.

In TX_ARET mode, the radio transceiver first executes the CSMA-CA algorithm, as defined by IEEE 802.15.4–2006, section 7.5.1.4, initiated by a transmit start event. If the channel is IDLE a frame is transmitted from the Frame Buffer. If the acknowledgement frame is requested the radio transceiver additionally checks for an ACK reply.

A TRX24_TX_END interrupt indicates the completion of the TX_ARET transmit transaction.

Description

Configuration and address bits are to be set in TRX_OFF or PLL_ON state prior to switching to TX_ARET mode. It is further recommended to transfer the PSDU data to the Frame Buffer in advance. The transaction is started by either writing SLPTR=1 as described in section "Transceiver Pin Register TRXPR" on page 32 or writing a TX_START command to register TRX_STATE.

If the CSMA-CA detects a busy channel, it is retried as specified by bits MAX_CSMA_RETRIES of register XAH_CTRL_0. In case that CSMA-CA does not detect a clear channel after MAX_CSMA_RETRIES it aborts the TX_ARET transaction, issues a TRX24_TX_END interrupt and sets the value of the TRAC_STATUS register bits to CHANNEL_ACCESS_FAILURE.

During transmission of a frame the radio transceiver parses bit 5 (ACK Request) of the MAC header (MHR) frame control field of the PSDU data (PSDU octet #1) to be transmitted to check if an ACK reply is expected.

If an ACK is expected the radio transceiver automatically switches into receive mode to wait for a valid ACK reply. After receiving an ACK frame the Frame Pending subfield of that frame is parsed and the status register bits TRAC_STATUS are updated accordingly (see Table 9-16 below). This receive procedure does not overwrite the Frame Buffer content. Transmit data in the Frame Buffer is not changed during the entire TX_ARET transaction. Received frames other than the expected ACK frame are discarded.

If no valid ACK is received or after timeout of 54 symbol periods (864 µs), the radio transceiver retries the entire transaction (including CSMA-CA) until the maximum number of retransmissions as set by the bits MAX_FRAME_RETRIES in register XAH_CTRL_0 is exceeded.

After that, the microcontroller may read the value of the bits TRAC_STATUS of register TRX_STATE to verify whether the transaction was successful or not. The register bits are set according to the following cases:

Table 9-16. Interpretation of the TRAC_STATUS register bits

Value	Name	Description
0	SUCCESS	The transaction was responded by a valid ACK, or, if no ACK is requested, after a successful frame transmission
1	SUCCESS_DATA_PENDING	Equivalent to SUCCESS; indicates pending frame data according to the MHR frame control field of the received ACK response
3	CHANNEL_ACCESS_FAILURE	Channel is still busy after MAX_CSMA_RETRIES of CSMA-CA

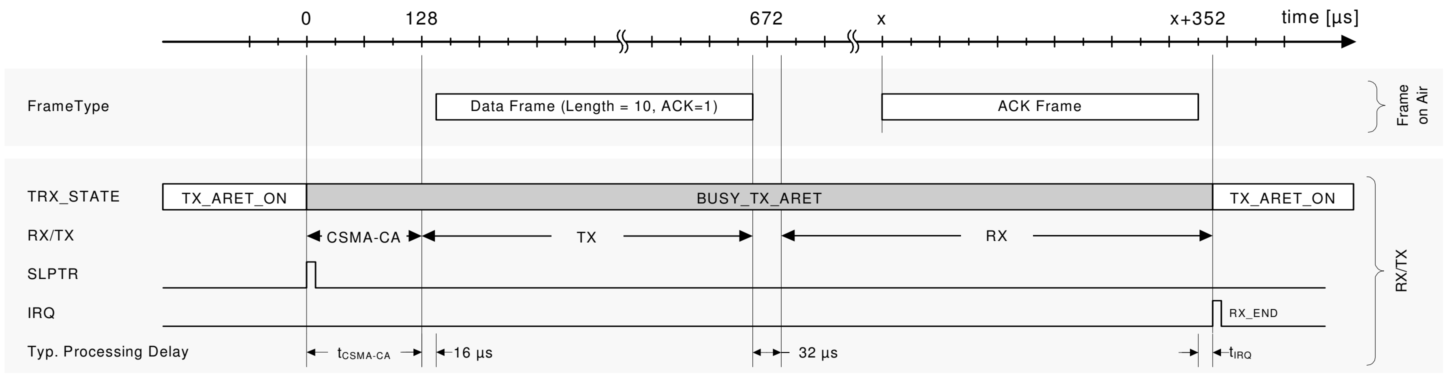
Value	Name	Description
5	NO_ACK	No acknowledgement frames were received during all retry attempts
7	INVALID	Entering TX_ARET mode sets TRAC_STATUS = 7

Note that if no ACK is expected (according to the content of the received frame in the Frame Buffer), the radio transceiver issues a TRX24_TX_END interrupt directly after the frame transmission has been completed. The value of the bits TRAC_STATUS of register TRX_STATE is set to SUCCESS.

A value of MAX_CSMA_RETRIES = 7 initiates an immediate TX_ARET transaction without performing CSMA-CA. This is required to support slotted acknowledgement operation. Further the value MAX_FRAME_RETRIES is ignored and the TX_ARET transaction is performed only once.

A timing example of a TX_ARET transaction is shown in Figure 9-13 below .

Figure 9-13. Example Timing of a TX_ARET Transaction



Note: 1. $t_{CSMA-CA}$ defines the random CSMA-CA processing time.

Here an example data frame of length 10 with an ACK request is transmitted, see Table 9-13 on page 60. After the transmission the radio transceiver switches to receive mode and expects an acknowledgement response. During the whole transaction including frame transmit, wait for ACK and ACK receive the radio transceiver status register TRX_STATUS signals BUSY_TX_ARET.

A successful reception of the acknowledgment frame is indicated by the TRX24_TX_END interrupt. The status register TRX_STATUS changes back to TX_ARET_ON. The TX_ARET status register TRAC_STATUS changes as well to TRAC_STATUS = SUCCESS or TRAC_STATUS = SUCCESS_DATA_PENDING if the frame pending subfield of the received ACK frame was set to 1.

9.4.2.6 Interrupt Handling

The interrupt handling in the Extended Operating Mode is similar to the Basic Operating Mode (see section "Interrupt Handling" on page 38). The microcontroller enables interrupts by setting the appropriate bit in register IRQ_MASK.

For RX_AACK and TX_ARET the following interrupts (Table 9-13 on page 60) inform about the status of a frame reception and transmission:

Table 9-13. Interrupt Handling in Extended Operating Mode

Mode	Interrupt	Description
RX_AACK	TRX24_RX_START	Indicates a PHR reception
	TRX24_AMI	Issued at address match
	TRX24_RX_END	Signals completion of RX_AACK transaction if successful – A received frame must pass the address filter; – The FCS is valid
TX_ARET	TRX24_TX_END	Signals completion of TX_ARET transaction
Both	TRX24_PLL_LOCK	Entering RX_AACK_ON or TX_ARET_ON state from TRX_OFF state, the TRX24_PLL_LOCK interrupt signals that the transaction can be started

RX_AACK

For RX_AACK it is recommended to enable the TRX24_RX_END interrupt. This interrupt is issued only if a frame passes the frame filtering (see section "Frame Filtering" on page 54) and has a valid FCS. This is different to Basic Operating Mode (see section "Basic Operating Mode" on page 35). The use of the other interrupts is optional.

On reception of a valid PHR a TRX24_RX_START interrupt is issued. The TRX24_AMI interrupt indicates an address match (see filter rules in section "Frame Filtering" on page 54). The completion of a frame reception with a valid FCS is indicated by the TRX24_RX_END interrupt.

Thus it can happen that a TRX24_RX_START and/or a TRX24_AMI interrupt are issued, but no TRX24_RX_END interrupt.

The end of an acknowledgment transmission is confirmed by a TRX24_TX_END interrupt.

TX_ARET

In TX_ARET interrupt TRX24_TX_END is only issued after completing the entire TX_ARET transaction.

Acknowledgement frames do not issue a TRX24_RX_START, TRX24_AMI or a TRX24_RX_END interrupt.

All other interrupts as described in section Table 9-2 on page 34 are also available in Extended Operating Mode.

9.4.2.7 Register Summary

The following registers (Table 9-14 below) are to be configured to control the Extended Operating Mode:

Table 9-14. Register Summary

Register Name	Description
TRX_STATUS	Radio transceiver status, CCA result
TRX_STATE	Radio transceiver state control, TX_ARET status
TRX_CTRL_1	TX_AUTO_CRC_ON
PHY_CC_CCA	CCA mode control, Table 9-21 on page 70
CCA_THRES	CCA threshold settings, see "Overview" on page 70
XAH_CTRL_1	RX_AACK control

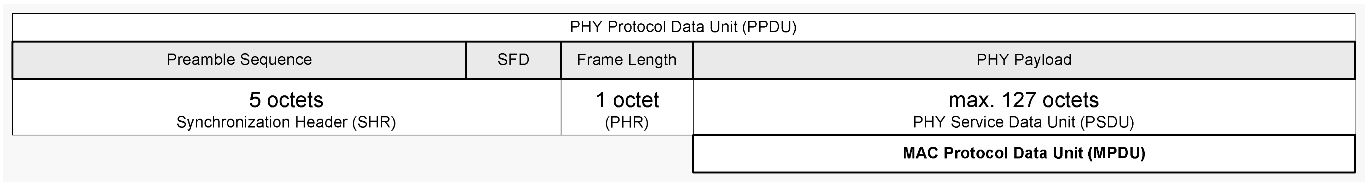
Register Name	Description
IEEE_ADDR7 .... IEEE_ADDR0 PAN_ID1 PAN_ID0 SHORT_ADDR1 SHORT_ADDR0	Address filter configuration Short address, PAN-ID and IEEE address
XAH_CTRL_0	TX_ARET control, retries value control
CSMA_SEED_0	CSMA-CA seed value
CSMA_SEED_1	CSMA-CA seed value, RX_AACK control
CSMA_BE	CSMA-CA back-off exponent control

9.5 Functional Description

9.5.1 Introduction – IEEE 802.15.4-2006 Frame Format

Figure 9-14 below provides an overview of the physical layer (PHY) frame structure as defined by IEEE 802.15.4. Figure 9-15 on page 62 shows the frame structure of the medium access control (MAC) layer.

Figure 9-14. IEEE 802.15.4 Frame Format - PHY-Layer Frame Structure (PPDU)


PHY Protocol Data Unit (PPDU)
Preamble Sequence
SFD
Frame Length
PHY Payload
5 octets
Synchronization Header (SHR)
1 octet
(PHR)
max. 127 octets
PHY Service Data Unit (PSDU)
MAC Protocol Data Unit (MPDU)

9.5.1.1 PHY Protocol Layer Data Unit (PPDU)

9.5.1.1.1 Synchronization Header (SHR)

The SHR consists of a four-octet preamble field (all zero), followed by a single byte start-of-frame delimiter (SFD) which has the predefined value 0xA7. During transmit, the SHR is automatically generated by the radio transceiver, thus the Frame Buffer shall contain PHR and PSDU only.

The transmission of the SHR requires 160 µs (10 symbols). As the frame buffer access is normally faster than the over-air data rate, this allows the application software to initiate a transmission without having transferred the full frame data already. Instead it is possible to subsequently write the frame content.

During frame reception, the SHR is used for synchronization purposes. The matching SFD determines the beginning of the PHR and the following PSDU payload data.

9.5.1.1.2 PHY Header (PHR)

The PHY header is a single octet following the SHR. The least significant 7 bits denote the frame length of the following PSDU, while the most significant bit of that octet is reserved, and shall be set to 0 for IEEE 802.15.4 compliant frames.

On receive the PHR is returned as the first octet during Frame Buffer read access. Even though the standard only defines frame lengths ≤127 bytes, the radio transceiver is able to transmit and receive frame length values >127. For IEEE 802.15.4 compliant operation bit 8 has to be masked by software. The reception of a valid PHR is signaled by a TRX24_RX_START interrupt.

On transmit the PHR has to be written first to the Frame Buffer.

9.5.1.1.3 PHY Payload (PHY Service Data Unit, PSDU)

The PSDU has a variable length between 0 and *aMaxPHYPacketSize* (127, maximum PSDU size in octets) whereas the last two octets are used for the Frame Check Sequence (FCS). The length of the PSDU is signaled by the frame length field (PHR) as described in Table 9-15 below. The PSDU contains the MAC Protocol Layer Data Unit (MPDU).

Received frames with a frame length field set to 0x00 (invalid PHR) are not by an interrupt.

Table 9-15 below summarizes the type of payload versus the frame length value.

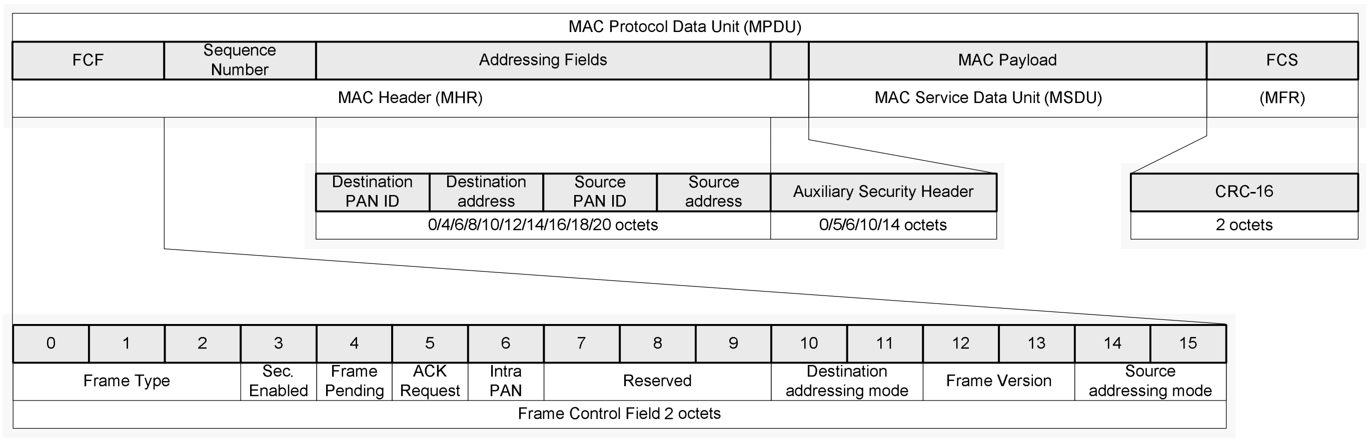
Table 9-15. Frame Length Field - PHR

Frame Length Value	Payload
0 - 4	Reserved
5	MPDU (Acknowledgement)
6 – 8	Reserved
9 - *aMaxPHYPacketSize*	MPDU

9.5.1.2 MAC Protocol Layer Data Unit (MPDU)

Figure 9-15 below shows the frame structure of the MAC layer.

Figure 9-15. IEEE 802.15.4 Frame Format - MAC-Layer Frame Structure (MPDU)



9.5.1.2.1 MAC Header (MHR) Fields

The MAC header consists of the Frame Control Field (FCF), a sequence number, and the addressing fields (which are of variable length and can even be empty in certain situations).

9.5.1.2.2 Frame Control Field (FCF)

The FCF consists of 16 bits, and occupies the first two octets of either the MPDU or the PSDU, respectively.

Figure 9-26. IEEE 802.15.4-2006 Frame Control Field (FCF)

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Frame Type			Sec. Enabled	Frame Pending	ACK Request	Intra PAN	Reserved			Destination addressing mode		Frame Version		Source addressing mode	
Frame Control Field 2 octets															

Bit [2:0]: describe the frame type. Table 9-16 below summarizes frame types defined by IEEE 802.15.4, section 7.2.1.1.1.

Table 9-16. Frame Control Field – Frame Type Subfield

Frame Control Field Bit Assignments		**Description**
Frame Type Value **b_2 b_1 b_0**	**Value**	
000	0	Beacon
001	1	Data
010	2	Acknowledge
011	3	MAC command
100 – 111	4 – 7	Reserved

This subfield is used for address filtering by the third level filter rules. Only frame types 0 – 3 pass the third level filter rules (refer to section "Frame Filtering" on page 54). Automatic address filtering of the radio transceiver is enabled when using the RX_AACK mode (refer to "RX_AACK_ON – Receive with Automatic ACK" on page 46).

A reserved frame (frame type value > 3) can be received if bit AACK_UPLD_RES_FT of register XAH_CTRL_1 is set. For details refer to chapter "Configuration of non IEEE 802.15.4 Compliant Scenarios" on page 52. Address filtering is also provided in Basic Operating Mode as explained in "Basic Operating Mode" on page 35.

Bit 3: indicates whether security processing applies to this frame.

Bit 4: is the “Frame Pending” subfield. This field can be set in an acknowledgment frame (ACK) in response to a data request MAC command frame. This bit indicates that the node, which transmitted the ACK, has more data to send to the node receiving the ACK.

For acknowledgment frames automatically generated by the radio transceiver, this bit is set according to the content of bit AACK_SET_PD of register CSMA_SEED_1 if the received frame was a data request MAC command frame.

Bit 5: forms the “Acknowledgment Request” subfield. If this bit is set within a data or MAC command frame that is not broadcast, the recipient shall acknowledge the reception of the frame within the time specified by IEEE 802.15.4 (i.e. within 192 µs for non beacon-enabled networks).

The radio transceiver parses this bit during RX_AACK mode and transmits an acknowledgment frame if necessary.

In TX_ARET mode this bit indicates if an acknowledgement frame is expected after transmitting a frame. If this is the case, the receiver waits for the acknowledgment frame, otherwise the TX_ARET transaction is finished.

Bit 6: the “Intra-PAN” subfield indicates that in a frame, where both, the destination and source addresses are present, the PAN-ID of the source address filed is omitted. In RX_AACK mode this bit is evaluated by the address filter logic of the radio transceiver.

Bit [11:10]: the “Destination Addressing Mode” subfield describes the format of the destination address of the frame. The values of the address modes are summarized in Table 9-17 below according to IEEE 802.15.4:

Table 9-17. Frame Control Field – Destination and Source Addressing Mode

Frame Control Field Bit Assignments		Description
Addressing Mode **b_{11} b_{10}** **b_{15} b_{14}**	**Value**	
00	0	PAN identifier and address fields are not present
01	1	Reserved
10	2	Address field contains a 16-bit short address
11	3	Address field contains a 64-bit extended address

If the destination address mode is either 2 or 3 (i.e. if the destination address is present), it always consists of a 16-bit PAN-ID first followed by either the 16-bit or 64-bit address as defined by the mode.

Bit [13:12]: the “Frame Version” subfield specifies the version number corresponding to the frame. These register bits are reserved in IEEE-802.15.4-2003.

This subfield shall be set to 0 to indicate a frame compatible with IEEE 802.15.4-2003 and 1 to indicate an IEEE 802.15.4-2006 frame. All other subfield values shall be reserved for future use.

The bit AACK_FVN_MODE of register CSMA_SEED_1 controls the RX_AACK behavior of frame acknowledgements. This register determines if, depending on the Frame Version Number, a frame is acknowledged or not. This is necessary for backward compatibility to IEEE 802.15.4-2003 and for future use. Even if frame version numbers 2 and 3 are reserved, it can be handled by the radio transceiver. For details refer to "CSMA_SEED_1 – Transceiver Acknowledgment Frame Control Register 2" on page 129.

See IEEE 802.15.4-2006, section 7.2.3 for details on frame compatibility.

Table 9-18. Frame Control Field – Frame Version Subfield

Frame Control Field Bit Assignments		Description
Frame Version **b_{13} b_{12}**	**Value**	
00	0	Frames are compatible with IEEE 802.15.4-2003
01	1	Frames are compatible with IEEE 802.15.4-2006
10	2	Reserved
11	3	Reserved

Bit [15:14]: the “Source Addressing Mode” subfield, with similar meaning as “Destination Addressing Mode” (refer to Table 9-17 above).

The subfields of the FCF (Bits 0–2, 3, 6, 10–15) affect the address filter logic of the radio transceiver while executing a RX_AACK operation (see "RX_AACK_ON – Receive with Automatic ACK" on page 46).

9.5.1.2.3 Frame Compatibility between IEEE 802.15.4-2003 and IEEE 802.15.4-2006

All unsecured frames according to IEEE 802.15.4-2006 are compatible with unsecured frames compliant with IEEE 802.15.4-2003 with two exceptions: a coordinator realignment command frame with the “Channel Page” field present (see IEEE 802.15.4-2006 7.3.8) and any frame with a MAC Payload field larger than *aMaxMACSafePayloadSize* octets.

Compatibility for secured frames is shown in the following table, which identifies the security operating modes for IEEE 802.15.4-2006.

Table 9-19. Frame Control Field – Security and Frame Version

Frame Control Field Bit Assignments		**Description**
Security Enabled b_3	**Frame Version** b_{13} b_{12}	
0	00	No security. Frames are compatible between IEEE 802.15.4-2003 and IEEE 802.15.4-2006.
0	01	No security. Frames are not compatible between IEEE 802.15.4-2003 and IEEE 802.15.4-2006.
1	00	Secured frame formatted according to IEEE 802.15.4-2003. This frame type is not supported in IEEE 802.15.4-2006.
1	01	Secured frame formatted according to IEEE 802.15.4-2006

9.5.1.2.4 Sequence Number

The one-octet sequence number following the FCF identifies a particular frame, so that duplicated frame transmissions can be detected. While operating in RX_AACK mode, the content of this field is copied from the frame to be acknowledged into the acknowledgment frame.

9.5.1.2.5 Addressing Fields

The addressing fields of the MPDU are used by the radio transceiver for address matching indication. The destination address (if present) is always first, followed by the source address (if present). Each address field consists of the Intra PAN-ID and a device address. If both addresses are present and the “Intra PAN-ID compression” subfield in the FCF is set to one, the source Intra PAN-ID is omitted.

Note that in addition to these general rules IEEE 802.15.4 further restricts the valid address combinations for the individual possible MAC frame types. For example the situation where both addresses are omitted (source addressing mode = 0 and destination addressing mode = 0) is only allowed for acknowledgment frames. The address filter in the radio transceiver has been designed to apply to IEEE 802.15.4 compliant frames. It can be configured to handle other frame formats and exceptions.

9.5.1.2.6 Auxiliary Security Header Field

The Auxiliary Security Header specifies information required for security processing and has a variable length. This field determines how the frame is actually protected (security level) and which keying material from the MAC security PIB is used (see IEEE 802.15.4-2006, 7.6.1). This field shall be present only if the Security Enabled subfield b3 is set to one (see section "Frame Compatibility between IEEE 802.15.4-2003 and IEEE 802.15.4-2006" on page 65). For details of its structure see IEEE 802.15.4-2006, 7.6.2 Auxiliary security header.

9.5.1.2.7 MAC Service Data Unit (MSDU)

This is the actual MAC payload. It is usually structured according to the individual frame type. A description can be found in IEEE 802.15.4-2006, chapter 5.5.3.2.

9.5.1.2.8 MAC Footer (MFR) Fields

The MAC footer consists of a two-octet Frame Checksum (FCS). For details refer to the following section "Frame Check Sequence (FCS)" below.

9.5.2 Frame Check Sequence (FCS)

The Frame Check Sequence (FCS) is characterized by:

- Indicate bit errors based on a cyclic redundancy check (CRC) of 16 bit length;
- Uses International Telecommunication Union (ITU) CRC polynomial;
- Automatically evaluated during reception;
- Can be automatically generated during transmission.

9.5.2.1 Overview

The FCS is intended for use at the MAC layer to detect corrupted frames at a first level of filtering. It is computed by applying an ITU CRC polynomial to all transferred bytes following the length field (MHR and MSDU fields). The frame check sequence has a length of 16 bit and is located in the last two bytes of a frame (MAC footer, see Figure 9-15 on page 62).

The radio transceiver applies an FCS check on each received frame. The result of the FCS check is stored in bit RX_CRC_VALID of register PHY_RSSI.

On transmit the radio transceiver generates and appends the FCS bytes during the frame transmission. This behavior can be disabled by setting the bit TX_AUTO_CRC_ON = 0 in register TRX_CTRL_1.

9.5.2.2 CRC calculation

The CRC polynomial used in IEEE 802.15.4 networks is defined by

$$G_{16}(x) = x^{16} + x^{12} + x^{5} + 1$$

The FCS shall be calculated for transmission using the following algorithm:

Let

$$M(x) = b_0 x^{k-1} + b_1 x^{k-2} + \ldots + b_{k-2} x + b_{k-1}$$

be the polynomial representing the sequence of bits for which the checksum is to be computed. Multiply M(x) by x^{16} giving the polynomial

$$N(x) = M(x) \cdot x^{16}$$

Divide $N(x)$ modulo 2 by the generator polynomial $G_{16}(x)$ to obtain the remainder polynomial

$$R(x) = r_0 x^{15} + r_1 x^{14} + \ldots + r_{14} x + r_{15}$$

The FCS field is given by the coefficients of the remainder polynomial, *R(x)*.

Example:

Consider a 5 octet ACK frame. The MHR field consists of

0100 0000 0000 0000 0101 0110.

The leftmost bit (b_0) is transmitted first in time. The FCS is in this case

0010 0111 1001 1110.

The leftmost bit (r_0) is transmitted first in time.

9.5.2.3 Automatic FCS generation

The automatic FCS generation is performed with register bit TX_AUTO_CRC_ON = 1 (reset value). This allows the radio transceiver to autonomously compute the FCS. For a frame with a frame length specified as N ($3 \leq N \leq 127$), the FCS is calculated on the first N-2 octets in the Frame Buffer and the resulting FCS field is transmitted in place of the last two octets from the Frame Buffer.

If the automatic FCS generation of the radio transceivers is enabled, the Frame Buffer write access can be stopped right after MAC payload. There is no need to write FCS dummy bytes.

In RX_AACK mode, when a received frame needs to be acknowledged, the FCS of the ACK frame is always automatically generated by the radio transceiver, independent of the TX_AUTO_CRC_ON setting.

Example:

A frame transmission of length five with TX_AUTO_CRC_ON set, is started with a Frame Buffer write access of five bytes (the last two bytes can be omitted). The first three bytes are used for FCS generation; the last two bytes are replaced by the internally calculated FCS.

9.5.2.4 Automatic FCS check

An automatic FCS check is applied on each received frame with a frame length $N \geq 2$. The bit RX_CRC_VALID of register PHY_RSSI is set if the FCS of a received frame is valid. The register bit is updated when issuing a TRX24_RX_END interrupt and remains valid until a new frame reception causes the next TRX24_RX_END interrupt.

In RX_AACK mode, the radio transceiver rejects the frame and the TRX24_RX_END interrupt is not issued if the FCS of the received frame is not valid.

In TX_ARET mode, the FCS and the sequence number of an ACK are automatically checked. The ACK is not accepted if one of those is not correct.

9.5.3 Received Signal Strength Indicator (RSSI)

The Received Signal Strength Indicator is characterized by:

- Minimum RSSI level is -90 dBm (RSSI_BASE_VAL);
- Dynamic range is 81 dB;
- Minimum RSSI value is 0;
- Maximum RSSI value is 28.

9.5.3.1 Overview

The RSSI is a 5-bit value indicating the receive power in the selected channel in steps of 3 dB. No attempt is made to distinguish IEEE 802.15.4 signals from others. Only the received signal strength is evaluated. The RSSI provides the basis for an ED measurement. See section "Energy Detection (ED)" below for details.

9.5.3.2 Reading RSSI

In Basic Operating Mode the RSSI value is valid in any receive state, and is updated every t_{TR25} = 2 µs to register PHY_RSSI.

It is not recommended to read the RSSI value when using the Extended Operating Mode. The automatically generated ED value should then be used (see section "Energy Detection (ED)" below).

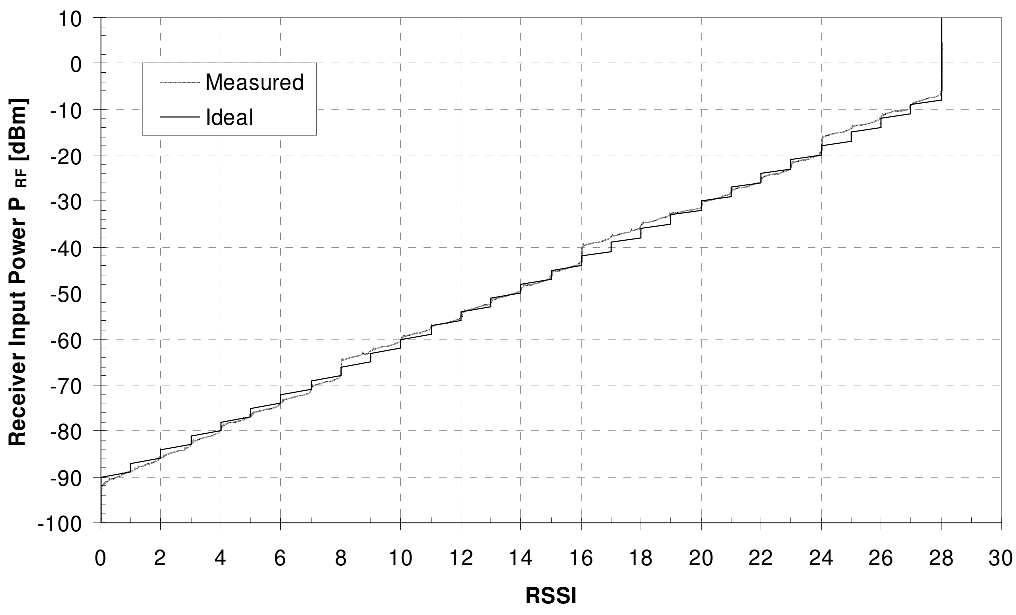
9.5.3.3 Data Interpretation

The RSSI value is a 5-bit value indicating the receive power in steps of 3 dB and with a range of 0- 28.

An RSSI value of 0 indicates a receiver RF input power of $P_{RF} < -90$ dBm. For an RSSI value in the range of 1 to 28, the RF input power can be calculated as follows:

$$P_{RF} = \text{RSSI\_BASE\_VAL} + 3 \cdot (\text{RSSI} - 1)\ [\text{dBm}]$$

Figure 9-17. Mapping between RSSI Value and Received Input Power



9.5.4 Energy Detection (ED)

The Energy Detection (ED) module is characterized by:

- 85 unique energy levels defined;
- 1 dB resolution.

9.5.4.1 Overview

The receiver ED measurement is used by the network layer as part of a channel selection algorithm. It is an estimation of the received signal power within the bandwidth of an IEEE 802.15.4 channel. No attempt is made to identify or decode signals on the channel. The ED value is calculated by averaging RSSI values over eight symbols (128 µs).

For High Data Rate Modes the automated ED measurement duration is reduced to 32 µs as described in "High Data Rate Modes" on page 86. The measurement period in these modes is still 128 µs for manually initiated ED measurements as long as the receiver is in RX_ON state.

9.5.4.2 Measurement Description

There are two ways to initiate an ED measurement:

- Manually, by writing an arbitrary value to register PHY_ED_LEVEL, or
- Automatically, after detection of a valid SHR of an incoming frame.

For manually initiated ED measurements the radio transceiver needs to be in one of the states RX_ON or BUSY_RX. The end of the ED measurement is indicated by a TRX24_CCA_ED_DONE interrupt.

The automatic ED measurement is started if a SHR is detected. The end of the automatic measurement is not signaled by an interrupt.

The measurement result is stored after t_{TR26} = 140 µs (128 µs measurement duration and processing delay) in register PHY_ED_LEVEL.

Thus by using Basic Operating Mode a valid ED value from the currently received frame is accessible 108 µs after the TRX24_RX_START interrupt and remains valid until the next incoming frame generates a new TRX24_RX_START interrupt or until another ED measurement is initiated.

When using the Extended Operating Mode it is recommended to mask the TRX24_RX_START interrupt. Hence the interrupt cannot be used as timing reference. A successful frame reception is signalized by the TRX24_RX_END interrupt. The minimum time span between a TRX24_RX_END interrupt and a following SFD detection is t_{TR27} = 96 µs due to the length of the SHR. The ED value needs to be read within 224 µs including the ED measurement time after the TRX24_RX_END interrupt. Otherwise it could be overwritten by the result of the next measurement cycle. This is important for time critical applications or if the TRX24_RX_START interrupt is not used to indicate the reception of a frame.

The values of the register PHY_ED_LEVEL are:

Table 9-20. Register Bit PHY_ED_LEVEL Interpretation

PHY_ED_LEVEL	Description
0xFF	Reset value
0x00 … 0x54	ED measurement result of the last ED measurement

Note: 1. It is not recommended to manually initiate an ED measurement when using the Extended Operating Mode.

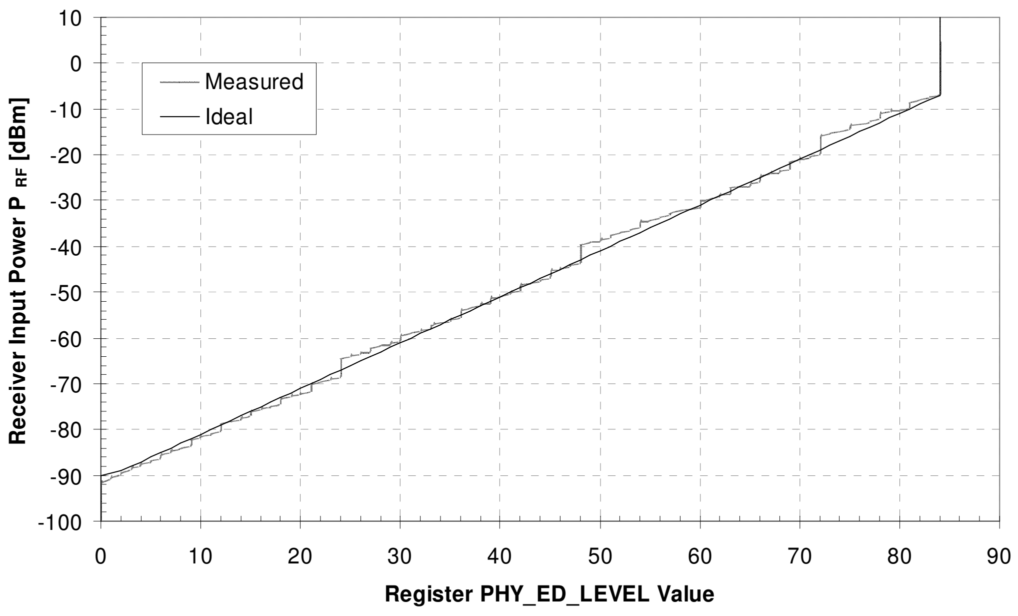
9.5.4.3 Data Interpretation

The PHY_ED_LEVEL is an 8-bit register. The ED value of the radio transceiver has a valid range from 0x00 to 0x54 with a resolution of 1 dB. All other values do not occur. A value of 0xFF indicates the reset value. A value of PHY_ED_LEVEL = 0 indicates that the measured energy is less than -90 dBm (see parameter RSSI_BASE_VAL in section "Receiver Characteristics" on page 508). Due to environmental conditions (temperature, voltage, semiconductor parameters etc.) the calculated ED value has a maximum tolerance of ±5 dB, this is to be considered as constant offset over the measurement range.

An ED value of 0 indicates an RF input power of P_{RF} ≤ -90 dBm. For an ED value in the range of 0 to 84, the RF input power can be calculated as follows:

$P_{RF} = -90 + ED$ [dBm]

Figure 9-18. Mapping between values in PHY_ED_LEVEL and Received Input Power



9.5.4.4 Interrupt Handling

The TRX24_CCA_ED_DONE interrupt is issued at the end of a manually initiated ED measurement.

Note that an ED request should only be initiated in one of the receive states. Otherwise the radio transceiver generates a TRX24_CCA_ED_DONE interrupt but no ED measurement was performed.

9.5.5 Clear Channel Assessment (CCA)

The main features of the Clear Channel Assessment (CCA) module are:

- All 4 modes are available as defined by IEEE 802.15.4-2006 in section 6.9.9;
- Adjustable threshold for energy detection algorithm.

9.5.5.1 Overview

A CCA measurement is used to detect a clear channel. Four modes are specified by IEEE 802.15.4-2006:

Table 9-21. CCA Mode Overview

CCA Mode	Description
1	*Energy above threshold.* CCA shall report a busy medium upon detecting any energy above the ED threshold.
2	*Carrier sense only.* CCA shall report a busy medium only upon the detection of a signal with the modulation and spreading characteristics of an IEEE 802.15.4 compliant signal. The signal strength may be above or below the ED threshold.

CCA Mode	Description
0, 3	*Carrier sense with energy above threshold.* CCA shall report a busy medium using a logical combination of – Detection of a signal with the modulation and spreading characteristics of this standard and – Energy above the ED threshold. Where the logical operator may be configured as either OR (mode 0) or AND (mode 3).

9.5.5.2 Configuration and CCA Request

The CCA modes are configurable via register PHY_CC_CCA.

Usimg the Basic Operating Mode, a CCA request can be initiated manually by setting CCA_REQUEST = 1 of register PHY_CC_CCA, if the radio transceiver is in any RX state. The current channel status (CCA_STATUS) and the CCA completion status (CCA_DONE) are accessible in register TRX_STATUS.

The CCA evaluation is done over eight symbol periods and the result is accessible t_{TR28} = 140 µs (128 µs measurement duration and processing delay) after the request. The end of a manually initiated CCA measurement is indicated by a TRX24_CCA_ED_DONE interrupt.

The sub-register CCA_ED_THRES of register CCA_THRES defines the received power threshold of the “energy above threshold” algorithm. The threshold is calculated by RSSI_BASE_VAL + 2 • CCA_ED_THRES [dBm]. Any received power above this level is interpreted as a busy channel.

Note that it is not recommended to manually initiate a CCA measurement when using the Extended Operating Mode.

9.5.5.3 Data Interpretation

The current channel status (CCA_STATUS) and the CCA completion status (CCA_DONE) are accessible in register TRX_STATUS. Note, register bits CCA_DONE and CCA_STATUS are cleared in response to a CCA_REQUEST.

The completion of a measurement cycle is indicated by CCA_DONE = 1. If the radio transceiver detected no signal (idle channel) during the measurement cycle, the CCA_STATUS bit is set to 1.

When using the “energy above threshold” algorithm, any received power above CCA_ED_THRES level is interpreted as a busy channel. The “carrier sense” algorithm reports a busy channel when detecting an IEEE 802.15.4 signal above the RSSI_BASE_VAL (see parameter RSSI_BASE_VAL in "Transceiver Electrical Characteristics" on page 507). The radio transceiver is also able to detect signals below this value, but the detection probability decreases with the signal power.

9.5.5.4 Interrupt Handling

The TRX24_CCA_ED_DONE interrupt is issued at the end of a manually initiated CCA measurement.

Note: A CCA request should only be initiated in the receive states of Basic Operating Mode. Otherwise the radio transceiver generates a TRX24_CCA_ED_DONE interrupt and sets the register bit CCA_DONE = 1 even if no CCA measurement was performed.

9.5.5.5 Measurement Time

The response time for a manually initiated CCA measurement depends on the receiver state.

In RX_ON state the CCA measurement is done over eight symbol periods and the result is accessible 140 µs after the request (see section "Configuration and CCA Request" on page 71).

In BUSY_RX state the CCA measurement duration depends on the CCA Mode and the CCA request relative to the reception of an SHR. The end of the CCA measurement is indicated by a TRX24_CCA_ED_DONE interrupt. The variation of a CCA measurement period in BUSY_RX state is described in Table 9-22 below.

Table 9-22. CCA Measurement Period and Access in BUSY_RX state

CCA Mode	Request within ED measurement[1]	Request after ED measurement
1	*Energy above threshold.*	
	CCA result is available after finishing automated ED measurement period.	CCA result is immediately available after request.
2	*Carrier sense only.*	
	CCA result is immediately available after request.	
3	*Carrier sense with Energy above threshold (AND).*	
	CCA result is available after finishing automated ED measurement period.	CCA result is immediately available after request.
0	*Carrier sense with Energy above threshold (OR).*	
	CCA result is available after finishing automated ED measurement period.	CCA result is immediately available after request.

Note: 1. After receiving the SHR an automated ED measurement is started with a length of 8 symbol periods (PSDU rate 250 kb/s), refer to section "Energy Detection (ED)" on page 68. This automated ED measurement must be finished to provide a result for the CCA measurement. Only one automated ED measurement per frame is performed.

It is recommended to perform CCA measurements in RX_ON state only. To avoid accidental switching to BUSY_RX state the SHR detection can be disabled by setting bit RX_PDT_DIS of register RX_SYN. Refer to section "Receiver (RX)" on page 74 for details. The receiver remains in RX_ON state to perform a CCA measurement until the register bit RX_PDT_DIS is set back to continue the frame reception. In this case the CCA measurement duration is 8 symbol periods.

9.5.6 Link Quality Indication (LQI)

According to IEEE 802.15.4 the LQI measurement is a characterization of the strength and/or quality of a received packet. The measurement may be implemented using receiver ED, a signal-to-noise ratio estimation or a combination of these methods. The use of the LQI result by the network or application layers is not specified in this standard. LQI values shall be an integer ranging from 0x00 to 0xFF. The minimum and maximum LQI values (0x00 and 0xFF) should be associated with the lowest and highest quality compliant signals, respectively, and LQI values in between should be uniformly distributed between these two limits.

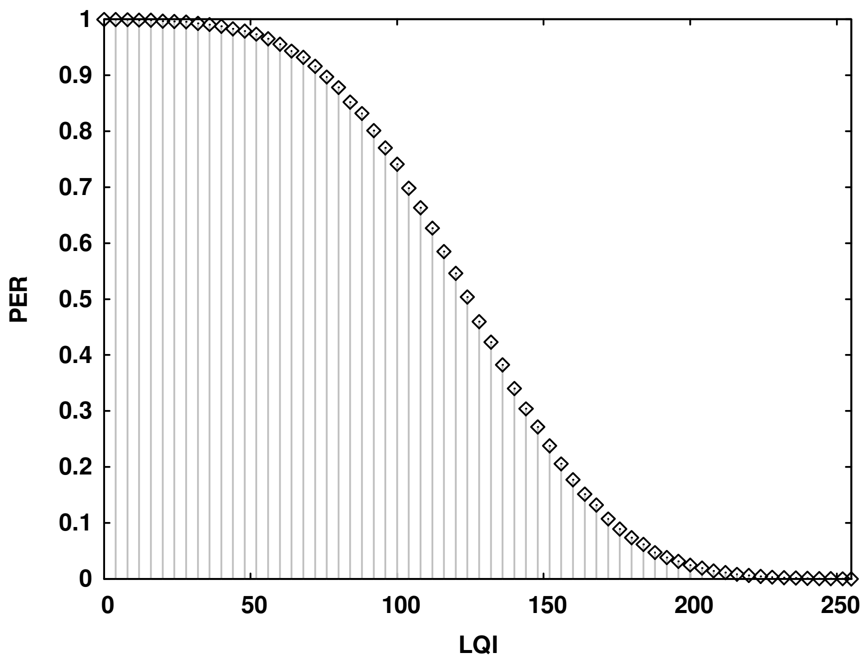
9.5.6.1 Overview

The LQI measurement of the radio transceiver is implemented as a measure of the link quality which can be described with the packet error rate (PER) of this link. A LQI value can be associated with an expected packet error rate. The PER is the ratio of erroneous received frames to the total number of received frames. A PER of zero indicates no frame error whereas at a PER of one no frame was received correctly.

The radio transceiver uses correlation results of multiple symbols within a frame to determine the LQI value. This is done for each received frame. The minimum frame length for a valid LQI value is two octets PSDU. LQI values are integers ranging from 0 to 255.

The following figure shows an example of a conditional packet error rate when receiving a certain LQI value.

Figure 9-19. Conditional Packet Error Rate versus LQI



The values are taken from received frames of PSDU length of 20 octets on transmission channels with reasonable low multipath delay spreads. If the transmission channel characteristic has a higher multipath delay spread than assumed in the example, the PER is slightly higher for a certain LQI value. Since the packet error rate is a statistical value, the PER shown in Figure 9-19 above is based on a huge number of transactions. A reliable estimation of the packet error rate cannot be based on a single or a small number of LQI values.

9.5.6.2 Request a LQI Measurement

The LQI byte can be obtained after a frame has been received by the radio transceiver. One additional byte is automatically attached to the received frame containing the LQI value. This information can also be read via Frame Buffer read access, see "User accessible Frame Content" on page 78. The LQI byte can be read after the TRX24_RX_END interrupt.

9.5.6.3 Data Interpretation

According to IEEE 802.15.4 a low LQI value is associated with low signal strength and/or high signal distortions. Signal distortions are mainly caused by interference signals and/or multipath propagation. High LQI values indicate a sufficient high signal power and low signal distortions.

Note that the received signal power as indicated by the received signal strength indication (RSSI) value or energy detection (ED) value of the radio transceiver do not characterize the signal quality and the ability to decode a signal.

As an example, a received signal with an input power of about 6 dB above the receiver sensitivity likely results in a LQI value close to 255 for radio channels with very low signal distortions. For higher signal power the LQI value becomes independent of the actual signal strength. This is because the packet error rate for these scenarios tends towards zero and further increased signal strength i.e. increasing the transmission power does not decrease the error rate any further. In this case RSSI or ED can be used to evaluate the signal strength and the link margin.

ZigBee networks often require the identification of the “best” routing between two nodes. Both the LQI and the RSSI/ED can be used for this, dependent on the optimization criteria. If a low packet error rate (corresponding to high throughput) is the optimization criteria then the LQI value should be taken into consideration. If a low transmission power or the link margin is the optimization criteria then the RSSI/ED value is also helpful.

Combinations of LQI, RSSI and ED are possible for routing decisions. As a rule of thumb RSSI and ED values are useful to differentiate between links with high LQI values. Transmission links with low LQI values should be discarded for routing decisions even if the RSSI/ED values are high. This is because RSSI and ED do not say anything about the possibility to decode a signal. It is only an information about the received signal strength whereas the source can be an interferer.

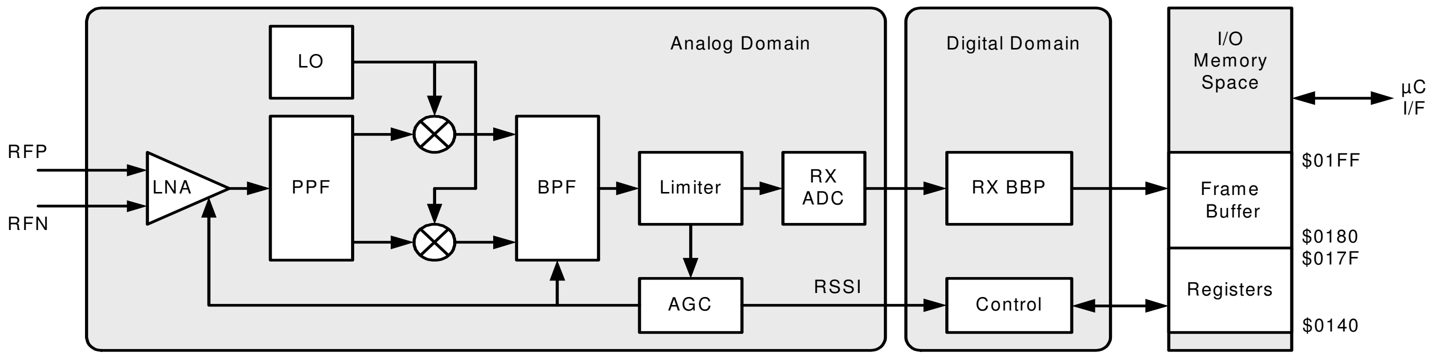
9.6 Module Description

9.6.1 Receiver (RX)

9.6.1.1 Overview

The receiver is split into an analog radio front-end and a digital base band processor (RX BBP) according to the following figure. The digital base band processor and the control engine are connected to the Frame Buffer and control registers which are located in the microcontroller I/O memory space (see "I/O Memory" on page 25 and "Transceiver to Microcontroller Interface" on page 31).

Figure 9-20. Receiver Block Diagram



The differential RF signal is amplified by a low noise amplifier (LNA), filtered (PPF) and down converted to an intermediate frequency by a mixer. Channel selectivity is performed using an integrated band pass filter (BPF). A limiting amplifier (Limiter) provides sufficient gain to overcome the DC offset of the succeeding analog-to-digital converter (RX ADC) and generates a digital RSSI signal. The ADC output signal is sampled and processed further by the digital base band receiver (RX BBP).

The RX BBP performs additional signal filtering and signal synchronization. The frequency offset of each frame is calculated by the synchronization unit and is used during the remaining receive process to correct the offset. The receiver is designed to handle frequency and symbol rate deviations up to ±120 ppm caused by combined receiver and transmitter deviations. For details refer to chapter "General RF Specifications" on page 507. Finally the signal is demodulated and the data are stored in the Frame Buffer.

In Basic Operating Mode (see "Basic Operating Mode" on page 35), the reception of a frame is indicated by a TRX24_RX_START interrupt. Accordingly its end is signalized by a TRX24_RX_END interrupt. Based on the quality of the received signal a link quality indicator (LQI) is calculated and appended to the frame. For details refer to. Additional signal processing is applied to the frame data to provide further status information like ED value (register PHY_ED_LEVEL) and FCS correctness (register PHY_RSSI).

Beyond these features the Extended Operating Mode of the radio transceiver supports address filtering and pending data indication. For details refer to "Extended Operating Mode" on page 43.

9.6.1.2 Frame Receive Procedure

The frame receive procedure including the radio s setup for reception and reading PSDU data from the Frame Buffer is described in "Frame Receive Procedure" on page 84.

9.6.1.3 Configuration

In Basic Operating Mode the receiver is enabled by writing command RX_ON to the TRX_CMD bits of register TRX_STATE in the states TRX_OFF or PLL_ON. Similarly in Extended Operating Mode the receiver is enabled for RX_AACK operation from the states TRX_OFF or PLL_ON by writing the command RX_AACK_ON. There is no additional configuration required to receive IEEE 802.15.4 compliant frames when using the Basic Operating Mode. However, the frame reception in the Extended Operating Mode requires further register configurations. For details refer to "Extended Operating Mode" on page 43.

The receiver has an outstanding sensitivity performance of -100 dBm. At certain environmental conditions or for High Data Rate Modes (see "High Data Rate Modes" on page 86) it may be useful to manually decrease this sensitivity. This is achieved by adjusting the detector threshold of the synchronization header using the RX_PDT_LEVEL bits of register RX_SYN. Received signals with a RSSI value below the threshold do not activate the demodulation process.

Furthermore, it may be useful to protect a received frame against overwriting by subsequent received frames.

A Dynamic Frame Buffer Protection is enabled with register bit RX_SAFE_MODE (TRX_CTRL_2) set (refer to "Dynamic Frame Buffer Protection" on page 91). After a frame has been received, the buffer is protected for new incoming frames and the receiver remains in RX_ON or RX_AACK_ON state until the RX_SAFE_MODE bit is cleared by the controller. The Frame Buffer content is only protected if the FCS is valid.

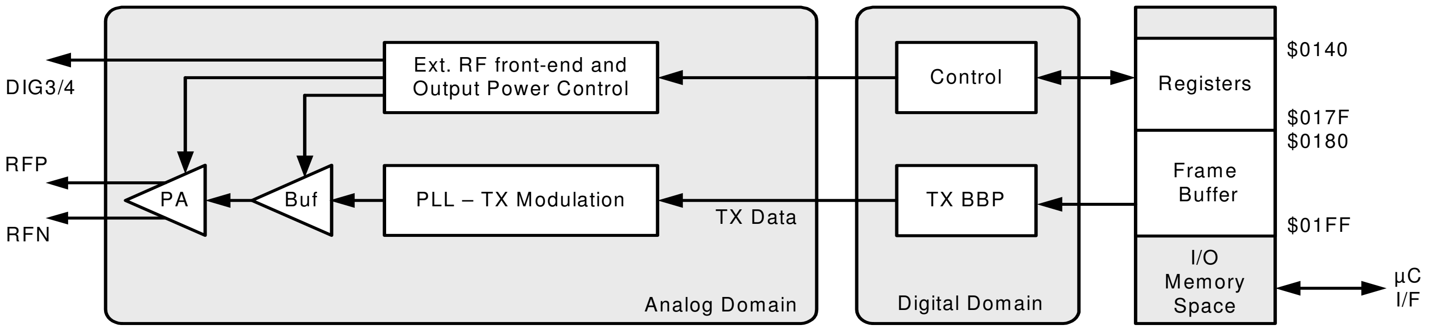
A Static Frame Buffer Protection is enabled with bit RX_PDT_DIS of register RX_SYN set. The receiver remains in RX_ON or RX_AACK_ON state and no further SHR is detected until the register bit RX_PDT_DIS is set back.

9.6.2 Transmitter (TX)

9.6.2.1 Overview

The transmitter consists of a digital base band processor (TX BBP) and an analog front end as shown in the following figure.

Figure 9-21. Transmitter Block Diagram



The TX BBP reads the frame data from the Frame Buffer and performs the bit-to-symbol and symbol-to-chip mapping as specified by IEEE 802.15.4 in section 6.5.2. The O-QPSK modulation signal is generated and fed into the analog radio front end.

The fractional-N frequency synthesizer (PLL) converts the baseband transmit signal to the RF signal which is amplified by the power amplifier (PA). The PA output is internally connected to bidirectional differential antenna pins (RFP, RFN) so that no external antenna switch is needed.

9.6.2.2 Frame Transmit Procedure

The frame transmit procedure including writing PSDU data in the Frame Buffer and initiating a transmission is described in section "Frame Transmit Procedure" on page 84. The controller must ensure to provide valid frame data before starting the frame transmission. For save operation, it is recommended to write the complete frame into the Frame Buffer before starting the frame transmission.

9.6.2.3 Configuration

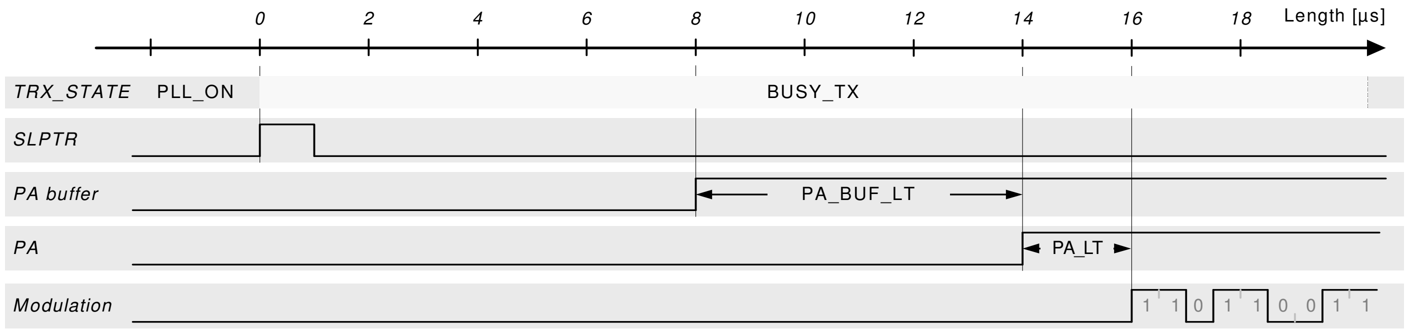
The maximum output power of the transmitter is typically +3.5 dBm. The output power can be configured via the TX_PWR bits of register PHY_TX_PWR. The output power of the transmitter can be controlled over a 20 dB range.

A transmission can be started from PLL_ON or TX_ARET_ON state by writing ‘1’ to bit SLPTR of the TRXPR register or by writing TX_START command to the TRX_CMD bits of register TRX_STATE.

9.6.2.4 TX Power Ramping

The PA buffer and PA are enabled sequentially to optimize the output power spectral density (PSD). A timing example using default settings illustrates the sequence in the next figure. In this example the transmission is initiated with the rising edge of the SLPTR bit. The radio transceiver state changes from PLL_ON to BUSY_TX. The modulation starts 16 µs after SLPTR.

Figure 9-22. TX Power Ramping



When using an external RF front-end (refer to "RX/TX Indicator" on page 90) it may be required to adjust the startup time of the external PA relative to the internal building blocks to optimize the overall PSD. This can be achieved using register bits PA_BUF_LT and PA_LT of register PHY_TX_PWR.

9.6.3 Frame Buffer

The radio transceiver contains a 128 byte dual port SRAM. One port of the frame buffer is directly connected to the controller I/O space. Therefore random access to single frame bytes is possible. The other port connects to the internal transmitter and receiver modules. Both ports are independent and simultaneously accessible for data communication.

The Frame Buffer uses the controller I/O address space 0x180 to 0x1FF for RX and TX operation of the radio transceiver and can keep one IEEE 802.15.4 RX or one TX frame of maximum length at a time.

Frame Buffer access is only possible if the radio transceiver is enabled (PRTRX24 bit in the Power Reduction Register PRR1 is not set) and not in SLEEP state.

9.6.3.1 Data Management

Data in the Frame Buffer (received data or data to be transmitted) remain valid as long as:

- No new frame or other data are written into the buffer;
- No new frame is received (in any BUSY_RX state);
- No state change into radio transceiver SLEEP state is made;
- No radio transceiver RESET (see bit TRXRST in "TRXPR – Transceiver Pin Register" on page 169) or system reset took place;
- Bit PRTRX24 in register "PRR1 – Power Reduction Register 1" on page 168 is not set;

By default there is no protection of the Frame Buffer against overwriting. If a frame is received during a Frame Buffer read access of a previously received frame, the stored data might be overwritten.

Finally the application software should check the transferred frame data integrity by a FCS check.

The state of the radio transceiver should be changed to PLL_ON state after reception to protect the Frame Buffer content against overwriting with new, incoming frames. This can be achieved by writing immediately the command PLL_ON to the TRX_CMD bits of register TRX_STATE after receiving the frame indicated by a TRX24_RX_END interrupt.

Alternatively Dynamic Frame Buffer Protection can be used to protect received frames against overwriting. For details refer to "Dynamic Frame Buffer Protection" on page 91.

Both procedures do not protect the Frame Buffer from overwriting by the application software.

In Extended Operating Mode during TX_ARET operation (see "TX_ARET_ON – Transmit with Automatic Retry and CSMA-CA Retry" on page 57) the radio transceiver switches to receive if an acknowledgement of a previously transmitted frame was requested. During this period received frames are evaluated but not stored in the Frame Buffer. This allows the radio transceiver to wait for an acknowledgement frame and retry the frame transmission without writing the frame data to the Frame Buffer again.

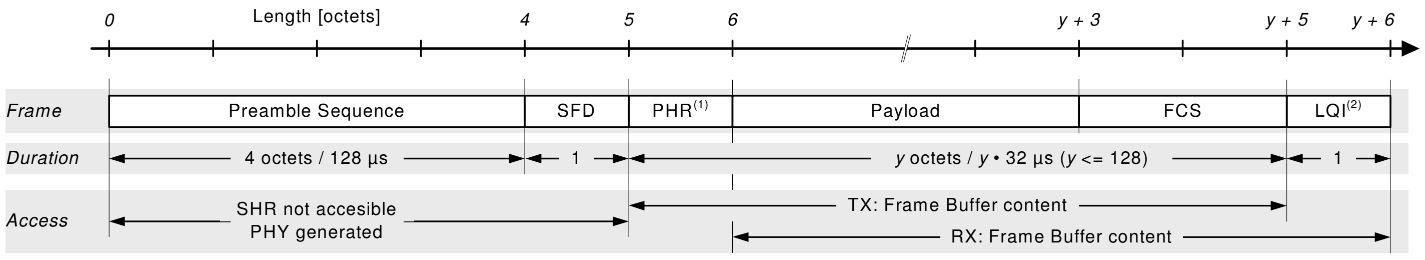
A radio transceiver state change except a transition to radio transceiver SLEEP state or a radio transceiver RESET does not affect the Frame Buffer content. The Frame Buffer is powered off and the stored data gets lost if the radio transceiver is forced into radio transceiver SLEEP state.

Access conflicts may occur when reading and writing data simultaneously at the two independent ports of the Frame Buffer TX/RX BBP and Controller interface.

9.6.3.2 User accessible Frame Content

The radio transceiver supports an IEEE 802.15.4 compliant frame format as shown in the following figure.

Figure 9-31. Transceiver Frame Structure



Notes: 1. Stored into Frame Buffer for TX operation
2. Stored into Frame Buffer during frame reception.

A frame comprises two sections. The radio transceiver internally generated SHR field and the user accessible part are stored in the Frame Buffer. The SHR contains the preamble and the SFD field. The variable frame section contains the PHR and the PSDU including the FCS (see "Overview" on page 66).

The Frame Buffer content differs depending on the direction of the communication (receive or transmit). To access the data follow the procedures described in "Radio Transceiver Usage" on page 83.

During frame reception, the payload and the link quality indicator (LQI) value of a successfully received frame are stored in the Frame Buffer. The radio transceiver appends the LQI value to the frame data after the last received octet. Information of the frame length is not stored in the Frame Buffer. The frame length information is located in register TST_RX_LENGTH.

The SHR (except the SFD used to generate the last octet of the SHR) can generally not be read by the application software.

The PHR and the PSDU need to be stored in the Frame Buffer for frame transmission. The PHR byte is the first byte in the Frame Buffer (address 0x180) and must be calculated based on the PHR and the PSDU. The maximum frame size supported by the radio transceiver is 128 bytes. If the TX_AUTO_CRC_ON bit is set in register PHY_TX_PWR, the FCS field of the PSDU is replaced by the automatically calculated FCS during frame transmission. There is no need to write the FCS field when using the automatic FCS generation.

Manipulating individual bytes of the Frame Buffer is simply possible by accessing the appropriate buffer address.

The minimum frame length supported by the radio transceiver for non IEEE 802.15.4 compliant frames is one byte (Frame Length Field + 1 byte of data).

9.6.4 Battery Monitor (BATMON)

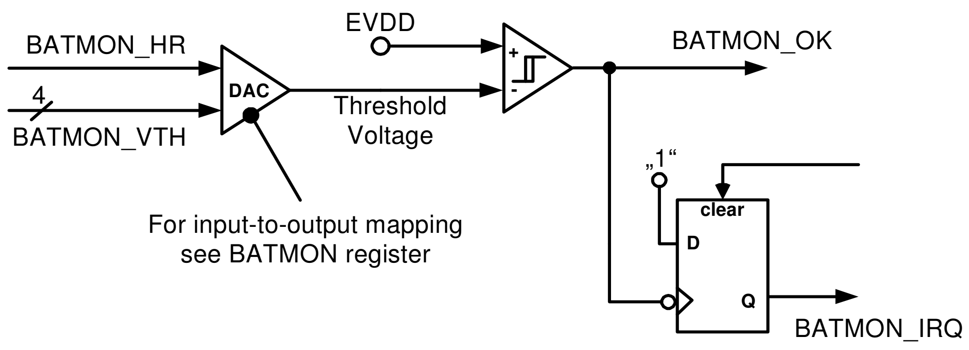
The main features of the battery monitor are:

- Configurable voltage threshold range from 1.7V to 3.675V
- Generates an interrupt when supply voltage drops below the threshold

9.6.4.1 Overview

The battery monitor (BATMON) detects and indicates a low supply voltage of EVDD. This is done by comparing the voltage of EVDD with a configurable, internal threshold voltage. A simplified schematic of the BATMON with the most important input and output signals is shown in the following figure.

Figure 9-24. Simplified Schematic of BATMON



9.6.4.2 Configuration

The Battery Monitor can be configured using the BATMON register. Register subfield BATMON_VTH sets the threshold voltage. It is configurable with a resolution of 75 mV in the upper voltage range (BATMON_HR = 1) and with a resolution of 50 mV in the lower voltage range (BATMON_HR = 0).

9.6.4.3 Data Interpretation

The bit BATMON_OK of register BATMON monitors the current value of the battery voltage:

- If BATMON_OK = 0 then the battery voltage is lower than the threshold voltage;
- If BATMON_OK = 1 then the battery voltage is higher than the threshold voltage;

The value BATMON_OK should be read out to verify the current supply voltage value after setting a new threshold.

Note: The battery monitor is inactive during SLEEP states. Refer to status register TRX_STATUS for details.

9.6.4.4 Interrupt Handling

A supply voltage drop below the configured threshold value is indicated by the BAT_LOW interrupt. The BAT_LOW status bit as well as the BATLOW_EN bit is located in the BATMON register. If BATLOW_EN =0, no IRQ is issued, but the status flag is set if the battery low event occurs.

The interrupt is only issued if BATMON_OK changes from 1 to 0 and the event is stored until the IRQ handler is called or the BAT_LOW IRQ is cleared manually by writing '1' to the BAT_LOW status flag.

No interrupt is generated when:

- The battery voltage is below the default 1.8V threshold at power up (BATMON_OK was never 1) or
- A new threshold is set which is still above the current supply voltage (BATMON_OK remains 0).

Noise or temporary voltage drops may generate unwanted interrupts when the battery voltage is close to the programmed threshold voltage. To avoid this:

- Disable the BAT_LOW interrupt with the BATLOW_EN Bit in the BATMON register and treat the battery as empty or
- Set a lower threshold value.

9.6.5 Crystal Oscillator (XOSC)

The main features of the crystal oscillator are:

- Amplitude controlled 16 MHz generation;
- 215 µs typical settling time after leaving SLEEP state;
- Configurable trimming with a capacitance array;

9.6.5.1 Overview

The crystal oscillator generates the reference frequency for the radio transceiver. All other internally generated frequencies of the radio transceiver are derived from this unique frequency. The overall system performance is therefore critically determined by the accuracy of the crystal reference frequency. The external components of the crystal oscillator should be selected carefully and the related board layout should be done with caution as described in section "Application Circuits" on page 493.

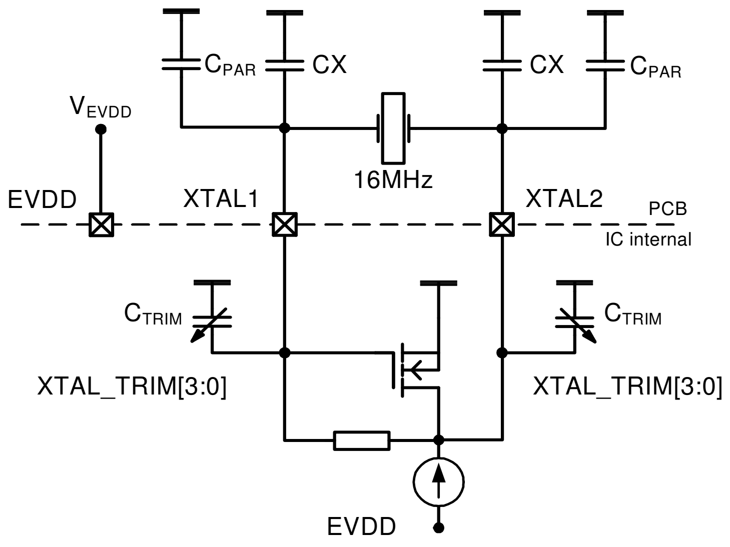
The register XOSC_CTRL provides access to the control signals of the oscillator. Two operating modes are supported. It is recommended to use the integrated oscillator setup as described in Figure 9-25 on page 81. Nevertheless a reference frequency can be fed to the internal circuitry by using an external clock reference as shown in Figure 9-26 on page 82.

9.6.5.2 Integrated Oscillator Setup

The output frequency of the internal oscillator depends on the load capacitance between the crystal pins XTAL1 and XTAL2. The total load capacitance C_L must be equal to the specified load capacitance of the crystal itself. It consists of the external capacitors CX and parasitic capacitances connected to the XTAL nodes.

The following figure shows all parasitic capacitances, such as PCB stray capacitances and the pin input capacitance summarized to C_{PAR}.

Figure 9-25. Simplified XOSC Schematic with External Components



Additional internal trimming capacitors C_{TRIM} are available. Any value in the range from 0 pF to 4.5 pF with a 0.3 pF resolution is selectable using XTAL_TRIM of register XOSC_CTRL. To calculate the total load capacitance, the following formula can be used

$$C_L = 0.5 \cdot (CX + C_{TRIM} + C_{PAR}).$$

The trimming capacitors provide the possibility to reduce frequency deviations caused by variations of the production process or by tolerances of external components. Note that the oscillation frequency can only be reduced by increasing the trimming capacitance. The frequency deviation caused by one step of C_{TRIM} decreases with increasing values of the crystal load capacitor.

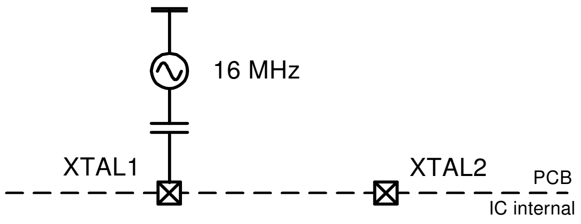
An amplitude control circuit is included to ensure stable operation under different operating conditions and for different crystal types. Enabling the crystal oscillator after leaving SLEEP state causes a slightly higher current during the amplitude build-up phase to guarantee a short start-up time. The current is reduced to the amount necessary for a robust oscillation during stable operation. This also keeps the drive level of the crystal low.

Crystals with a higher load capacitance are generally less sensitive to parasitic pulling effects caused by variations of external components or board and circuit parasitics. On the other hand a larger crystal load capacitance results in a longer start-up time and a higher steady state current consumption.

9.6.5.3 External Reference Frequency Setup

When using an external reference frequency, the signal must be connected to pin XTAL1 as indicated in Figure 9-26 on page 82 and the bits XTAL_MODE of register XOSC_CTRL need to be set to the external oscillator mode. The oscillation peak-to-peak amplitude shall between 100 mV and 500 mV, the optimum range is between 400 mV and 500 mV. Pin XTAL2 should not be wired

Figure 9-26. Setup for Using an External Frequency Reference



9.6.6 Frequency Synthesizer (PLL)

The main features of the phase-locked loop are:

- Generate RX/TX frequencies for all 2.4 GHz channels of IEEE 802.15.4;
- Autonomous calibration loops for stable operation within the operating range;
- Two PLL-interrupts for status indication;
- Fast PLL settling to support frequency hopping;

9.6.6.1 Overview

The PLL generates the RF frequencies for the radio transceiver. During receive operation the frequency synthesizer works as a local oscillator for the receive frequency of the radio transceiver. During transmit operation the voltage-controlled oscillator (VCO) is directly modulated to generate the RF transmit signal. The frequency synthesizer is implemented as a fractional-N PLL.

Two calibration loops ensure correct PLL functionality within the specified operating limits.

9.6.6.2 RF Channel Selection

The PLL is designed to support 16 channels in the 2.4 GHz ISM band with channel spacing of 5 MHz according to IEEE 802.15.4. The center frequency of these channels is defined as follows:

$F_c = 2405 + 5\,(k - 11)$ in [MHz], for $k = 11, 12 \ldots 26$

where *k* is the channel number.

The channel *k* is selected by the CHANNEL bits of register PHY_CC_CA.

9.6.6.3 Frequency Agility

When the PLL is enabled during state transition from TRX_OFF to PLL_ON the settling time is typically t_{TR4} = 110 µs including the settling time of the analog voltage regulator (AVREG) and the PLL self calibration (refer to Table 9-8 on page 42Table 9-8 and ? ? ? ? ? ? ? ? ? ? ?). A lock of the PLL is indicated with a TRX24_PLL_LOCK interrupt.

Switching between 2.4 GHz ISM band channels in PLL_ON or RX_ON states is typically done within t_{TR20} = 11 µs. This makes the radio transceiver highly suitable for frequency hopping applications.

The PLL frequency is changed to the transmit frequency within t_{TR23} = 16 µs after starting the transmit procedure and before starting the transmission. After the transmission the PLL settles back to the receive frequency within t_{TR24} = 32 µs. This frequency step does not generate a TRX24_PLL_LOCK or TRX24_PLL_UNLOCK interrupt within these time spans.

9.6.6.4 Calibration Loops

Due to temperature, supply voltage and part-to-part variations of the radio transceiver the VCO characteristics diverge. Two automated control loops are implemented to ensure a stable operation: center frequency (CF) tuning and delay cell (DCU) calibration. Both calibration loops are initiated automatically when the PLL is enabled during state transition from TRX_OFF to PLL_ON. The center frequency calibration is additionally initiated when the PLL changes to a center frequency of another channel.

It is recommended to initiate the calibration loops manually if the PLL operates for a long time on the same channel e.g. more than 5 min or the operating temperature changes significantly. Both calibration loops can be initiated manually by setting PLL_CF_START = 1 of register PLL_CF and PLL_DCU_START = 1 of register PLL_DCU. The device must be in PLL_ON or RX_ON state to start the calibration. The completion of the center frequency tuning is indicated by a TRX24_PLL_LOCK interrupt.

Both calibration loops may be run simultaneously.

9.6.6.5 Interrupt Handling

Two different interrupts indicate the PLL status. The TRX24_PLL_LOCK interrupt indicates that the PLL has locked. The TRX24_PLL_UNLOCK interrupt indicates an unexpected unlock condition.

A TRX24_PLL_LOCK interrupt is supposed to occur in the following situations:

- State change from TRX_OFF to PLL_ON / RX_ON/ RX_AACK_ON/ TX_ARET_ON;
- Channel change in states PLL_ON / RX_ON/ RX_AACK_ON/ TX_ARET_ON;

Any other occurrences of PLL interrupts indicate erroneous behavior and require checking of the actual device status.

The state transition from BUSY_TX to PLL_ON after successful transmission does not generate a TRX24_PLL_LOCK interrupt within the settling period.

9.6.7 Automatic Filter Tuning (FTN)

The FTN is incorporated to compensate device tolerances for temperature, supply voltage variations as well as part-to-part variations of the radio transceiver. The filter-tuning result is used to correct the transfer function of the analog baseband filter and the time constant of the PLL loop-filter (refer to "General Circuit Description" on page 30).

An FTN calibration cycle is initiated automatically when entering the radio transceiver TRX_OFF state from the SLEEP or RESET state.

Although receiver and transmitter are very robust against these variations, it is recommended to initiate the FTN manually if the radio transceiver does not use the SLEEP state. A calibration cycle is to be initiated in states TRX_OFF, PLL_ON or RX_ON if necessary. This applies in particular to the High Data Rate Modes with a much higher sensitivity to variations of the BPF transfer function. The recommended calibration interval is 5 min or less.

9.7 Radio Transceiver Usage

This section describes the basic procedures to receive and transmit frames with the radio transceiver.

9.7.1 Frame Receive Procedure

A frame reception comprises of two actions: The PHY listens for a frame, receives and demodulates the frame to the Frame Buffer and signalizes its reception to the application software. The application software reads the available frame data from the Frame Buffer after or during the progress of the frame reception.

While in state RX_ON or RX_AACK_ON the radio transceiver searches for incoming frames on the selected channel. First a TRX24_RX_START interrupt indicates the detection of an IEEE 802.15.4 compliant frame assuming the appropriate interrupts are enabled. The frame reception is completed when issuing the TRX24_RX_END interrupt.

Different Frame Buffer read access scenarios are recommended for:

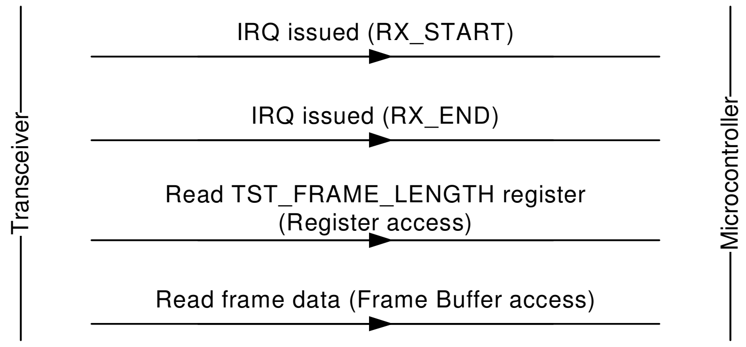
- Non-time critical applications: read access starts after the TRX24_RX_END interrupt;
- Time-critical applications: read access starts after the TRX24_RX_START interrupt;

The controller must ensure to read valid Frame Buffer contents. Reading frame data before frame reception is finished can lead to invalid data, if buffer regions are accessed which are not yet updated with the new frame.

While receiving a frame the data needs to be primarily stored in the Frame Buffer before reading it. This is ensured by accessing the first Frame Buffer byte at least 32 µs after the TRX24_RX_START interrupt.

It is recommended for operations considered to be not time-critical to wait for the TRX24_RX_END interrupt before starting a Frame Buffer read access. The following figure illustrates the frame receive procedure using the TRX24_RX_END interrupt.

Figure 9-27. Transactions between radio transceiver and microcontroller during receive



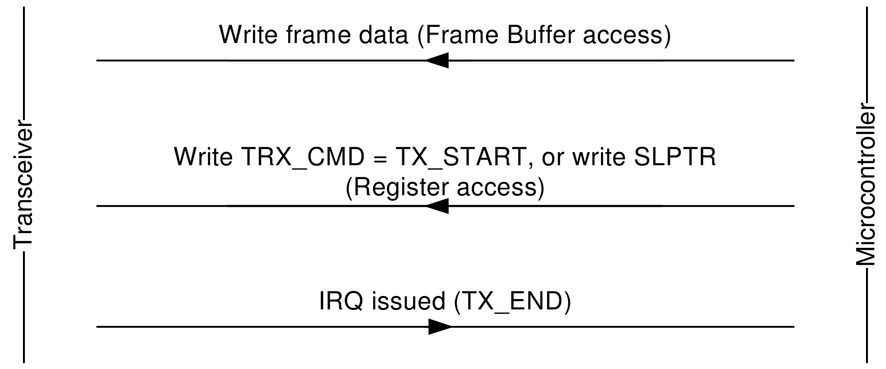
Critical protocol timing could require starting the Frame Buffer read access after the TRX24_RX_START interrupt. The first byte of the frame data can be read 32 µs after the TRX24_RX_START interrupt. The application software must ensure to read slower than the frame is received. Otherwise a Frame Buffer under-run occurs and the frame data may be not valid.

9.7.2 Frame Transmit Procedure

A frame transmission comprises of the two actions Frame Buffer write access and transmission of the Frame Buffer content. Both actions can be run in parallel if required by critical protocol timing.

Figure 9-28 on page 85 illustrates the frame transmit procedure by consecutively writing and transmitting the frame. The frame transmission is initiated writing SLPTR or writing command TX_START to register TRX_STATE after a Frame Buffer write access and while the radio transceiver is in state PLL_ON or TX_ARET_ON. The TRX24_TX_END interrupt indicates the completion of the transaction.

Figure 9-28. Transaction between radio transceiver and microcontroller during transmit

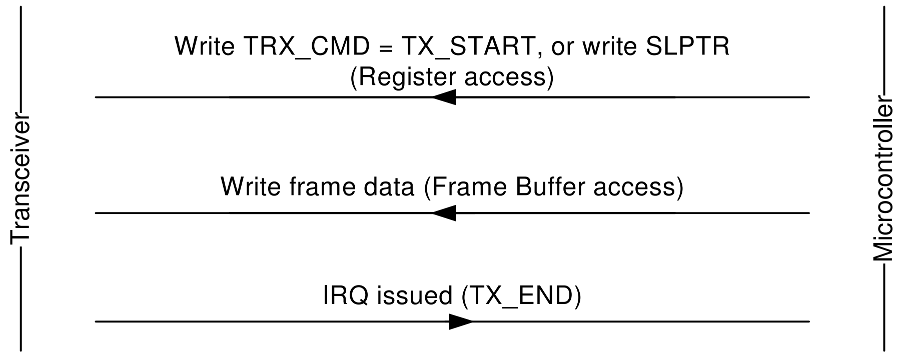

Transceiver
Write frame data (Frame Buffer access)
Write TRX_CMD = TX_START, or write SLPTR (Register access)
IRQ issued (TX_END)
Microcontroller

Alternatively a frame transmission can be started first, followed by the Frame Buffer write access (PSDU data) as shown in Figure 9-29 below. This is applicable for time critical applications.

A transmission is initiated either by writing SLPTR or by writing the TX_START command to the TRX_CMD bits of register TRX_STATE. The radio transceiver then starts transmitting the SHR which is internally generated.

This first phase requires 16 µs for PLL settling and 160 µs for SHR transmission. The PHR must be available in the Frame Buffer before this time elapses. Furthermore the Frame Buffer must be filled faster than the frame is transmitted to prevent a buffer under-run.

Figure 9-29. Time Optimized Frame Transmit Procedure


Transceiver
Write TRX_CMD = TX_START, or write SLPTR (Register access)
Write frame data (Frame Buffer access)
IRQ issued (TX_END)
Microcontroller

9.8 Radio Transceiver Extended Feature Set

9.8.1 Random Number Generator

The radio transceiver incorporates a 2-bit, noise observing, true random number generator to be used to:

- Generate random seeds for CSMA-CA algorithm (see"Extended Operating Mode" on page 43);

- Generate random values for AES key generation (see "Security Module (AES)" on page 92);

The random number is updated every t_{TR29} = 1 µs in Basic Operation Mode receive states. The values are stored in bits RND_VALUE of register PHY_RSSI.

9.8.2 High Data Rate Modes

The main features of the High Data Rate Modes are:

- High Data Rate Communication up to 2 Mb/s;
- Support of Basic and Extended Operating Mode;
- Support of other features of the Extended Feature Set;

9.8.2.1 Overview

The radio transceiver also supports alternative data rates higher than 250 kb/s for applications beyond IEEE 802.15.4 compliant networks.

The selection of a data rate does not affect the remaining functionality. Thus it is possible to run all features and operating modes of the radio transceiver in various combinations.

The data rate can be selected by writing bits OQPSK_DATA_RATE of register TRX_CTRL_2.

The High Data Rate Modes occupy the same RF channel bandwidth as the IEEE 802.15.4 – 2.4 GHz 250 kb/s standard mode. The sensitivity of the receiver is reduced due to the decreased spreading factor. The following table shows typical values of the sensitivity for different data rates.

Table 9-23. High Data Rate Sensitivity

High Data Rate	Sensitivity	Comment
250 kb/s	-100 dBm	PER ≤ 1%, PSDU length of 20 octets
500 kb/s	-9 dBm	PER ≤ 1%, PSDU length of 20 octets
1000 kb/s	-9 dBm	PER ≤ 1%, PSDU length of 20 octets
2000 kb/s	-8 dBm	PER ≤ 1%, PSDU length of 20 octets

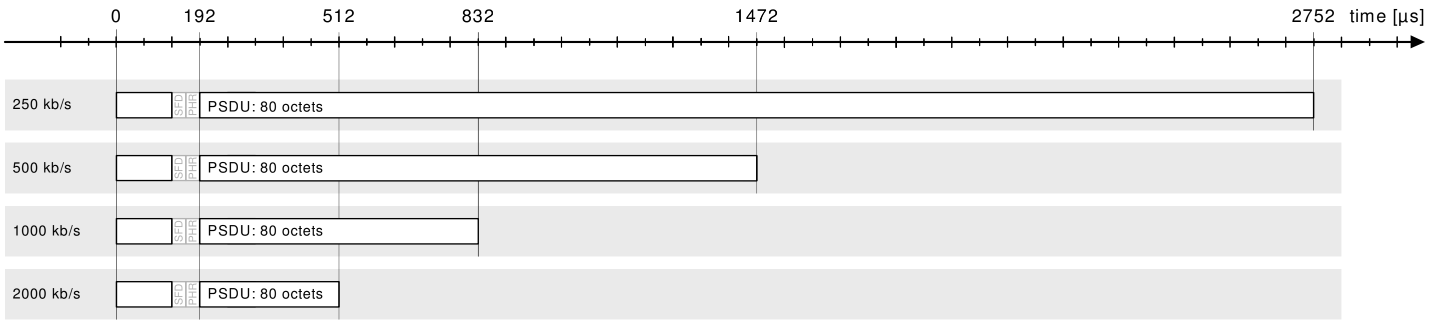
By default there is no header based signaling of the data rate within a transmitted frame. Thus nodes using a data rate other than the default IEEE 802.15.4 data rate of 250 kb/s are to be consistently configured in advance. The configurable start of frame delimiter (SFD) could be alternatively used as an indicator of the PHY data rate (see "Configurable Start-Of-Frame Delimiter (SFD)" on page 91).

9.8.2.2 High Data Rate Packet Structure

Higher data rate modulation is restricted to only the payload octets in order to allow appropriate frame synchronization. The SHR and the PHR field are transmitted with the IEEE 802.15.4 compliant data rate of 250 kb/s (refer to "Introduction – IEEE 802.15.4-2006 Frame Format" on page 61).

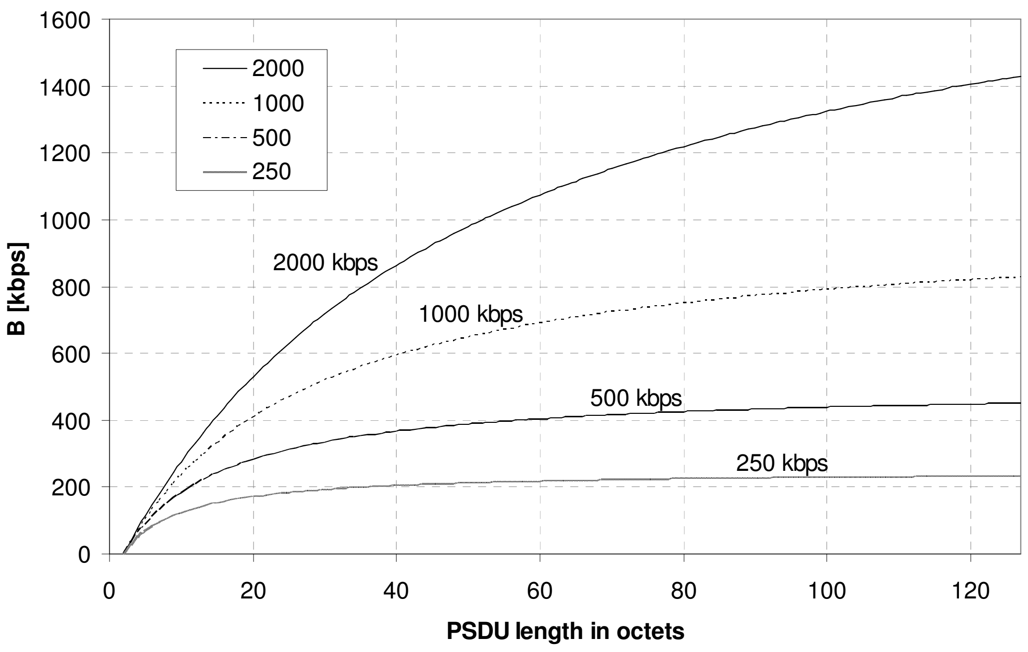
A comparison of the general packet structure for different data rates with an example PSDU length of 80 octets is shown in Figure 9-30 on page 87.

Figure 9-30. High Data Rate Frame Structure



The effective data rate is smaller than the selected data rate due to the overhead caused by the SHR, the PHR and the FCS. The overhead depends further on the length of the PSDU. A graphical representation of the effective data rate is shown in the following figure:

Figure 9-31. Effective Data Rate “B” for High Data Rate Mode



Therefore High Data Rate transmission and reception is useful for large PSDU lengths due to the higher effective data rate or to reduce the power consumption of the system. Furthermore the active on-air time using High Data Rate Modes is significantly reduced.

9.8.2.3 High Data Rate Frame Buffer Access

The Frame Buffer access to read or write frames for High Data Rate communication is similar to the procedure described in "Frame Buffer" on page 77. However the last byte in the Frame Buffer after the PSDU data is the ED value rather than the LQI value.

9.8.2.4 High Data Rate Energy Detection

According to IEEE 802.15.4 the ED measurement duration is 8 symbol periods. For frames operated at higher data rates the automated ED measurement duration is reduced to 32 µs to take the reduced frame length into account ("Energy Detection (ED)" on page 68).

9.8.2.5 High Data Rate Mode Options

Receiver Sensitivity Control

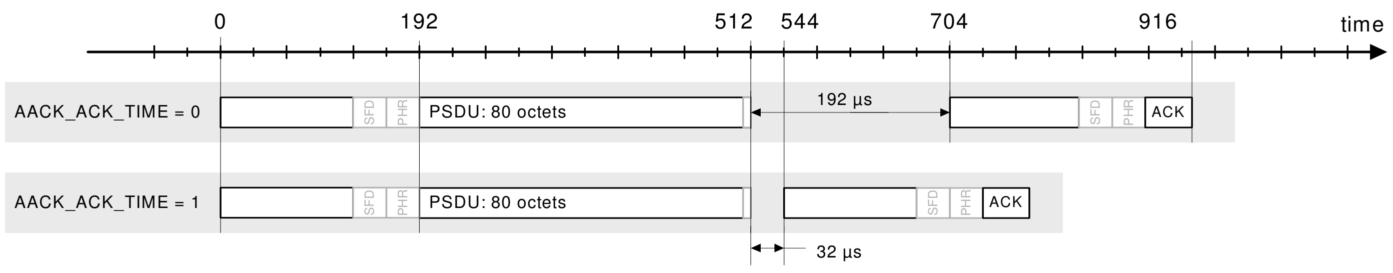
The different data rates between PPDU header (SHR and PHR) and PHY payload (PSDU) cause a different sensitivity between header and payload. This can be adjusted by defining sensitivity threshold levels of the receiver. The receiver does not receive frames with an RSSI level below the defined sensitivity threshold level (register bits RX_PDT_LEVEL > 0). Under these operating conditions the receiver current consumption is reduced by 500 µA (refer to chapter "Current Consumption Specifications" on page 509).

A description of the settings to control the sensitivity threshold with register RX_SYN can be found in section "RX_SYN – Transceiver Receiver Sensitivity Control Register" on page 118.

Reduced Acknowledgment Timing

On higher data rates the IEEE 802.15.4 compliant acknowledgment frame response time of 192 µs significantly reduces the effective data rate of the network. To minimize this influence in Extended Operating Mode RX_AACK (see section "RX_AACK_ON – Receive with Automatic ACK" on page 46), the acknowledgment frame response time can be reduced to 32 µs. Figure 9-32 below illustrates an example for a reception and acknowledgement of a frame with a data rate of 2000 kb/s and a PSDU length of 80 symbols. The PSDU length of the acknowledgment frame is 5 octets according to IEEE 802.15.4.

Figure 9-32. High Data Rate AACK Timing



The acknowledgment time is reduced from 192 µs to 32 µs if bit AACK_ACK_TIME of register XAH_CTRL_1 is set.

9.8.3 Antenna Diversity

The main features of the Antenna Diversity implementation are:

- Improves signal path robustness between nodes;
- Self-contained antenna diversity algorithm of the radio transceiver;
- Direct register based antenna selection;

9.8.3.1 Overview

The receive signal strength may vary and affect the link quality even for small changes of the antenna location due to multipath propagation effects between network nodes. These fading effects can result in an increased error floor or loss of the connection between devices.

Antenna Diversity can be applied to reduce the effects of multipath propagation and fading hence improving the reliability of a RF connection between network nodes.

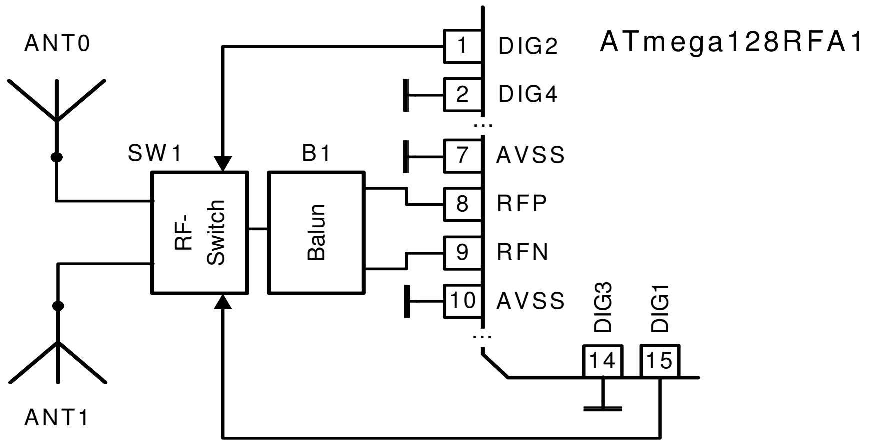
Antenna Diversity uses two antennas to switch to the most reliable RF signal path. This is done by the radio transceiver during RX_LISTEN and RX_AACK_ON state without interaction of the application software. Both antennas should be carefully separated from each other to ensure highly independent receive signals.

Antenna Diversity can be used in Basic and Extended Operating Modes and can also be combined with other features and operating modes like High Data Rate Mode and RX/TX Indication.

9.8.3.2 Antenna Diversity Application Example

A block diagram for an application using an antenna switch is shown in the following figure.

Figure 9-33. External Antenna Diversity – Block Diagram



Generally, the Antenna Diversity algorithm is enabled with bit ANT_DIV_EN=1 in register ANT_DIV. For the External Antenna Diversity the control of the antenna switch (SW1) must be enabled by bit ANT_EXT_SW_EN of register ANT_DIV. Under this condition the control pins DIG1 and DIG2 are configured as outputs. DIG1 and DIG2 are used to feed the antenna switch signal and its inverse to the differential inputs of the RF Switch (SW1).

The selected antenna is indicated by bit ANT_SEL of register ANT_DIV. The antenna selection continues searching for new frames on both antennas after the frame reception is completed. However the register bit ANT_SEL maintains its previous value (from the last received frame) until a new SHR has been found and the selection algorithm locked into one antenna again. Then the register bit ANT_SEL is updated.

The antenna defined by the ANT_CTRL bits of register ANT_DIV is selected for transmission. If for example the same antenna as selected for reception is to be used for transmission, the antenna must be set using the ANT_CTRL bits based on the value read from the ANT_SEL bit. It is recommended to read bit ANT_SEL after the TRX24_RX_START interrupt.

The autonomous search and selection allows the use of Antenna Diversity during reception even if the application software currently does not control the radio transceiver for instance in Extended Operating Mode.

An application software defined selection of a certain antenna can be done by disabling the automatic Antenna Diversity algorithm (ANT_DIV_EN = 0) and selecting one antenna using register bit ANT_CTRL.

If the radio transceiver is not in a receive or transmit state, it is recommended to disable register bit ANT_EXT_SW_EN and to set the port pins DIG1 and DIG2 to output low (DDG1 = 1, PORTG1 = 0, DDF2 = 1, PORTF2 = 0) in order to reduce the power consumption or avoid leakage current of the external RF switch especially during sleep modes.

9.8.3.3 Antenna Diversity with Extended Operation Modes

A combination of Extended Operation Mode and antenna diversity is allowed.

While the radio transceiver is in RX_AACK_ON state, it switches to an antenna with a reliable signal. The receive antenna selection is also used for transmission of an automatic acknowledge frame.

While the radio transceiver is in TX_ARET state, the selected antenna is automatically changed for every frame transmission retry.

9.8.3.4 Antenna Diversity Sensitivity Control

The detection threshold of the receiver has to be adjusted due to a different receive algorithm used by the Antenna Diversity algorithm. It is recommended to set bits PDT_THRES of register RX_CTRL to 3.

9.8.4 RX/TX Indicator

The main features are:

- RX/TX Indicator to control an external RF Front-End;
- Application software independent RF Front-End Control;
- Provide TX Timing Information;

9.8.4.1 Overview

While IEEE 802.15.4 is a low-cost, low-power standard, solutions supporting higher transmit output power are occasionally desirable. A differential control pin pair can indicate that the radio transceiver is currently in transmit mode to simplify the control of an optional external RF front-end.

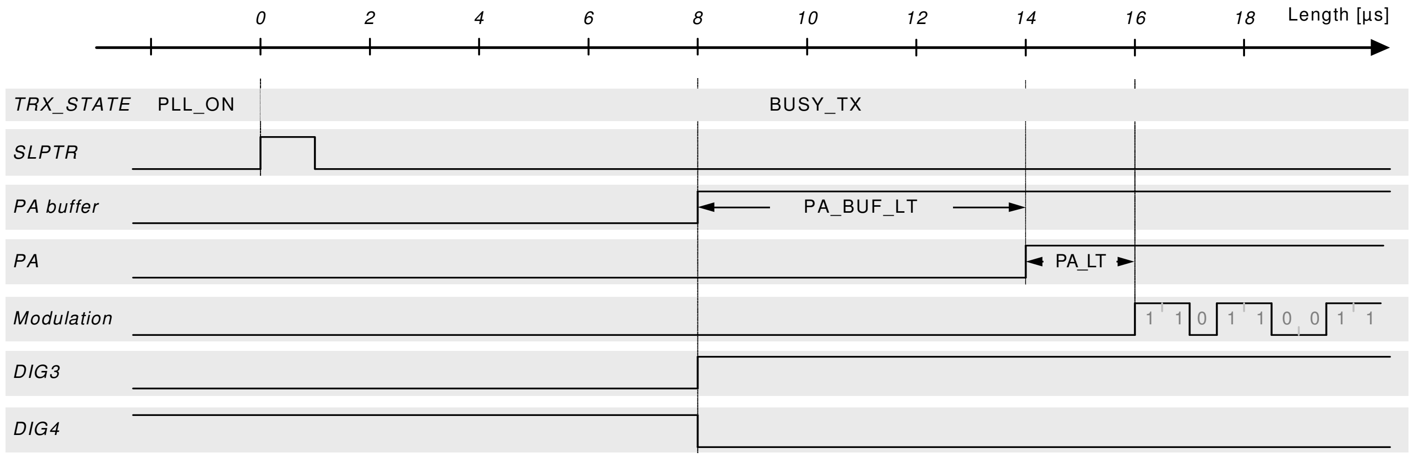
The control of an external RF front-end is done via the digital control pins DIG3/DIG4. The function of this pin pair is enabled with bit PA_EXT_EN of register TRX_CTRL_1. Pin DIG3 is set to low level and DIG4 to high level while the transmitter is turned off. The two pins change the polarity when the radio transceiver starts transmitting. This differential pin pair can be used to control PA, LNA and RF switches.

If the radio transceiver is not in a receive or transmit state, it is recommended to disable register bit PA_EXT_EN and to set the port pins DIG3 and DIG4 to output low (DDG0 = 1, PORTG0 = 0, DDF3 = 1, PORTF3 = 0) in order to reduce the power consumption or avoid leakage current of external RF switches and other building blocks especially during sleep modes.

9.8.4.2 External RF-Front End Control

The setup time of the external power amplifier (PA) relative to the internal building blocks should be adjusted when using an external RF front-end including a power amplifier to optimize the overall power spectral density (PSD) mask.

Figure 9-34. TX Power Ramping Control for RF Front-Ends



The start-up sequence of the individual building blocks of the internal transmitter is shown in the previous figure. The transmission is actually initiated by writing ‘1’ to SLPTR. The radio transceiver state changes from PLL_ON to BUSY_TX and the PLL settles to the transmit frequency within 16 µs (parameter tTR23 at page 42). The modulation starts 16 µs (parameter tTR10 at page 42) after the SLPTR=1. The PA buffer and the internal PA are enabled during this time.

The control of an external PA is done via the differential pin pair DIG3 and DIG4. DIG3 = H / DIG4 = L indicates that the transmission starts and can be used to enable an external PA. The timing of pins DIG3/DIG4 can be adjusted relative to the start of the frame and the activation of the internal PA buffer. This is controlled using the register bits PA_BUF_LT and PA_LT. For details refer to Figure 9-22 on page 77.

9.8.5 RX Frame Time Stamping

To determine the exact timing of an incoming frame e.g. for beaconing networks, the Symbol Counter should be used. SFD Time Stamping is enabled by setting bit SCTSE of the Symbol Counter Control Register SCCR0. The actual 32 Bit Symbol Counter value is captured in the SFD Time Stamp register SCTSR at the time, the SFD has been received. For details see section "SFD and Beacon Timestamp Generation" on page 135.

9.8.6 Configurable Start-Of-Frame Delimiter (SFD)

The SFD is a field indicating the end of the SHR and the start of the packet data. The length of the SFD is 1 octet (2 symbols). This octet is used for byte synchronization only and is not included in the Frame Buffer.

The value of the SFD could be changed if it is needed to operate non IEEE 802.15.4 compliant networks. An IEEE 802.15.4 compliant network node does not synchronize to frames with a different SFD value.

The register SFD_VALUE contains the one octet start-of-frame delimiter (SFD) to synchronize to a received frame. It is not recommended to set the low-order 4 bits to 0 due to the way the SHR is formed.

9.8.7 Dynamic Frame Buffer Protection

The ATmega128RFA1 continues the reception of incoming frames as long as it is in any receive state. When a frame was successfully received and stored into the Frame Buffer, the following frame will overwrite the Frame Buffer content again. To relax the timing requirements for a Frame Buffer read access the Dynamic Frame Buffer

Protection prevents that a new valid frame passes to the Frame Buffer until the buffer protection bit is cleared (RX_SAFE_MODE = 0).

A received frame is automatically protected against overwriting:

- in Basic Operating Mode, if its FCS is valid
- in Extended Operating Mode, if an TRX24_RX_END interrupt is generated

The Dynamic Frame Buffer Protection is enabled, if register bit RX_SAFE_MODE (register TRX_CTRL_2, see "TRX_CTRL_2 – Transceiver Control Register 2" on page 112) is set and the radio transceiver state is RX_ON or RX_AACK_ON.

Note that Dynamic Frame Buffer Protection only prevents write accesses from the air interface not from the application software. The application software may still modify the Frame Buffer content.

9.8.8 Security Module (AES)

The security module (AES) is characterized by:

- Hardware accelerated encryption and decryption;
- Compatible with AES-128 standard (128 bit key and data block size);
- ECB (encryption/decryption) mode and CBC (encryption) mode support;
- Stand-alone operation, independent of other blocks;

See TBD Application regarding the AES security module usage.

9.8.8.1 Overview

The security module is based on an AES-128 core according to the FIPS197 standard [5]. and provides two modes, the Electronic Code Book (ECB) and the Cipher Block Chaining (CBC). The security module works independent of other building blocks of the radio transceiver. Encryption and decryption can be performed in parallel to a frame transmission or reception.

The ECB and CBC modules including the AES core are clocked with the 16 MHz Radio Transceiver Crystal Oscillator.

Controlling the security block is possible over 5 Registers within AVR I/O space:

Table 9-24. Security Module Address Space Overview

Register Name	Description
AES_STATUS	AES status register
AES_CTRL	AES control register
AES_KEY	Access to 16 Byte key buffer
AES_STATE	Access to 16 Byte data buffer

9.8.8.2 Security Module Preparation

The use of the security module requires a configuration of the security engine before starting a security operation. The following steps are required:

Table 9-25. AES Engine Configuration Steps

Step	Description	Description
1	Key Setup	Write encryption or decryption key to KEY buffer (16 consecutive byte writes to AES_KEY)

Step	Description	Description
2	AES configuration	Select AES mode: ECB or CBC Select encryption or decryption Enable the AES Encryption Ready Interrupt AES_READY
3	Write Data	Write plaintext or cipher text to DATA buffer (16 consecutive byte writes to AES_STATE)
4	Start operation	Start AES operation
5	Wait for AES finished: 1. AES_READY IRQ or 2. polling AES_DONE bit (register AES_STATUS) or 3. wait for 24 µs	Wait until AES encryption/decryption is finished successfully
6	Read Data	Read cipher text or plaintext from DATA buffer (16 consecutive byte reads from AES_STATE)

Before starting any security operation a 16 Byte key must be written to the security engine (refer to section "Security Key Setup" below). This can be done by 16 consecutive write accesses to the I/O register AES_KEY. An internal address counter is incremented automatically with every read/ write operation. An AES encryption/ decryption run resets the internal byte counter. If the key and data buffer has not been read or written completely (all 16 Bytes), the following encryption/ decryption operation will finish with an error.

The following step selects either Electronic Code Book (ECB) or Cipher Block Chaining (CBC) as the AES_MODE. These modes are explained in more detail in section "Security Operation Modes" on page 94. Encryption or decryption must be further selected with bit AES_DIR of register AES_CTRL.

If the AES Error or AES Ready IRQ is used, the interrupt must be enabled with bit AES_IM.

Next the 128-bit plain text or cipher text data has to be provided to the AES hardware engine. The 16 data bytes must be consecutively written to the AES_STATE register. The AES_STATE register can be accessed in the same way as the key register (refer to "Security Key Setup" below).

The encryption or decryption is initiated with bit AES_REQUEST = 1.

The operation takes 24 µs and the completed encryption/ decryption is indicated by the AES_READY IRQ and the AES_DONE bit. The internal byte counter of the key and data buffer is cleared and the resulting data can be read out.

For additional information about the key and data buffer please refer to section "AES_KEY – AES Encryption and Decryption Key Buffer Register" on page 102 and "AES_STATE – AES Plain and Cipher Text Buffer Register" on page 102.

Notes: 1. Access to the security block is not possible while the radio transceiver is in state SLEEP.
2. All configurations of the security module, the SRAM content and keys are reset during SLEEP or RESET states.

9.8.8.3 Security Key Setup

The key is stored in a 16 Byte sequential buffer. To read or write the contents of the buffer, 16 consecutive read or write operations to the AES_KEY register are required.

A 16-folded read access to registers AES_KEY returns the last round key of the preceding security operation. This is the key required for the corresponding ECB decryption operation after an ECB encryption operation. However the initial AES key written to the security module in advance of an AES run (see step 1 in Table 9-25 on page 92) is not modified during an AES operation. This initial key is used for the next AES run although it cannot be read from AES_KEY.

Before accessing the Key Buffer it must be ensured, that the internal address counter is initialized correctly. This is the cases after Radio Transceiver Reset (see TRXPR – Transceiver Pin Register on page 169) or a completed AES Encryption/ Decryption operation. After an interrupted buffer read or write access, Address pointer reinitialization is recommended by a simple read access to the AES_CTRL register.

Note: 1. ECB decryption is not required for IEEE 802.15.4 or ZigBee security processing. The radio transceiver provides this functionality as an additional feature.

9.8.8.4 Security Operation Modes

9.8.8.4.1 Electronic Code Book (ECB)

ECB is the basic operating mode of the security module and is configured by the AES_CTRL register. The bit AES_MODE = 0 defines the ECB mode and bit AES_DIR selects the direction to either encryption or decryption. The data to be processed has to be written to registers AES_STATE.

A security operation can be started by writing the start command AES_REQUEST = 1 (AES_CTRL register).

The ECB encryption operation is illustrated in Figure 9-35 below. Figure 9-36 below shows the ECB decryption mode which is supported in a similar way.

Figure 9-35. ECB Mode - Encryption

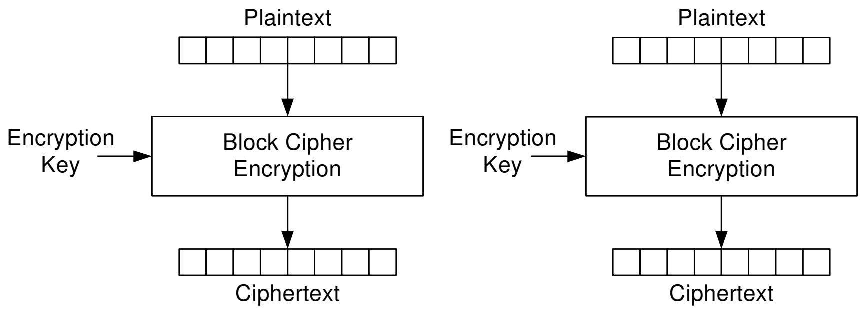
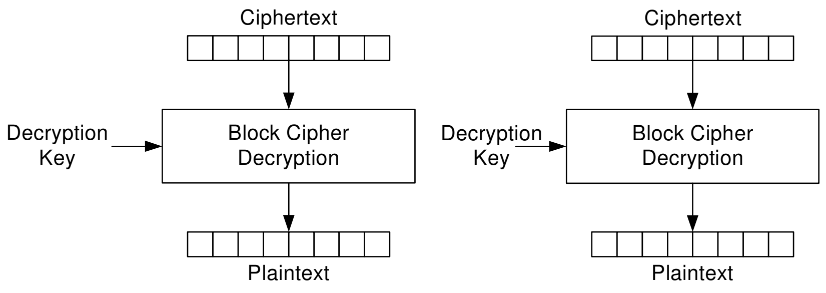


Figure 9-36. ECB Mode - Decryption



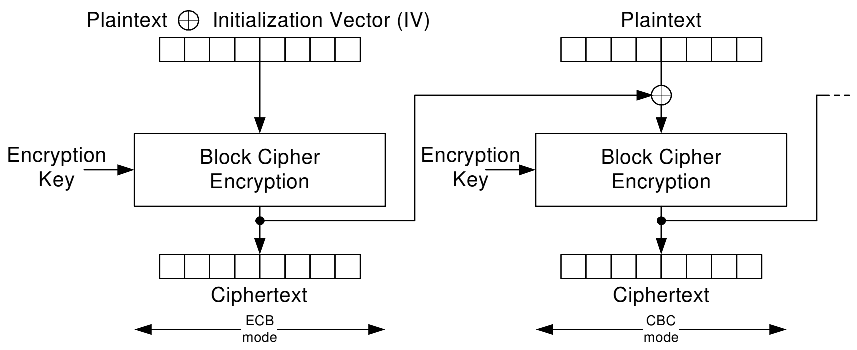
Due to the nature of AES algorithm the initial key to be used when decrypting is not the same as the one used for encryption. Instead it is the last round key. This last round key is the content of the key address space stored after running one full encryption cycle and must be saved for decryption. If the decryption key has not been saved, it has to be recomputed by first running a dummy encryption (of an arbitrary plaintext) using the original encryption key. Then the resulting round key must be fetched from the key memory and written back into the key memory as the decryption key.

ECB decryption is not used by either IEEE 802.15.4 or ZigBee frame security. Both of these standards do not directly encrypt the payload. Instead they protect the payload by applying a XOR operation between the original payload and the resulting (AES-) cipher text with a nonce (number used once). As the nonce is the same for encryption and decryption only ECB encryption is required. Decryption is performed by a XOR operation between the received cipher text and its own encryption result concluding in the original plain text payload upon success.

9.8.8.4.2 Cipher Block Chaining (CBC)

In CBC mode the result of a previous AES operation is XOR-combined with the new incoming vector forming the new plaintext to encrypt as shown in the next figure. This mode is used for the computation of a cryptographic checksum (message integrity code, MIC).

Figure 9-37. CBC Mode - Encryption



After preparing the AES key and defining the AES operation direction register bit AES_DIR, the data has to be provided to the AES engine and the CBC operation can be started.

The first CBC run has to be configured as ECB to process the initial data (plain text XOR with an initialization vector provided by the application software). All succeeding AES runs are to be configured as CBC by setting bit AES_MODE = 1 (AES_CTRL register). Bit AES_DIR (AES_CTRL register) must be set to AES_DIR = 0 to enable AES encryption. The data to be processed has to be transferred to the AES_STATE register. Setting bit AES_REQUEST = 1 (AES_CTRL register) as described in section "Security Operation Modes" on page 94 starts the first encryption. This causes the next 128 bits of plain text data to be XORed with the previous cipher text data, see Figure 9-37 above.

According to IEEE 802.15.4 the input for the very first CBC operation has to be prepared by a XOR operation of the plain text with the initialization vector (IV). The value of the initialization vector is 0. However any other initialization vector can be applied for non-compliant usage. This operation has to be prepared by the application software.

Note that the MIC algorithm of the IEEE 802.15.4-2006 standard requires CBC mode encryption only because it implements a one-way hash function.

The status of the security processing is indicated by register AES_STATUS. After a AES processing time of 24 µs the register bit AES_DONE changes to 1 (register AES_STATUS) indicating that the security operation has finished (see "Digital Interface Timing Characteristics" on page 507).

The end of the AES processing can also be indicated by the AES_READY Interrupt. The bit AES_ER of register AES_STATUS is set if the operation has finished with an error. Otherwise this bit is zero but AES_DONE is ‘1’.

9.8.8.5 AES Interrupt Handling

The AES Interrupt handling is slightly different from all other IRQ’s. If the AES_IM Bit (AES_CTRL Register) and the global interrupt enable flag is set, the AES core can generate an AES Ready Interrupt (AES_READY). If the IRQ is issued, the AES_STATUS register must be read to check the finish status of the last operation. If AES_DONE is set, the last AES operation finished successfully. If AES_ER is set, an error occurred during the last operation. The AES_ER flag is cleared automatically during the read access to the AES_STATUS register. The AES_DONE flag is cleared during the next read or write access to the AES_STATE (AES data) register.

The two status flags must be cleared before a new Interrupt can be issued.

If AES_IM is not set, the processing status can be polled by software (AES_STATUS register), but no Interrupt occurs.

9.9 Continuous Transmission Test Mode

9.9.1 Overview

The 2.4GHz transceiver offers a Continuous Transmission Test Mode to support final application / production tests as well as certification tests. In this test mode the radio transceiver transmits continuously a previously transferred frame (PRBS mode) or a continuous wave signal (CW mode).

In CW mode two different signal frequencies per channel can be transmitted:

- $f_1 = f_{CH} + 0.5$ MHz
- $f_2 = f_{CH} - 0.5$ MHz

Here f_{CH} is the channel center frequency programmed by register PHY_CC_CCA.

Note that in CW mode it is not possible to transmit a RF signal directly on the channel center frequency. PSDU data in the Frame Buffer must contain at least a valid PHR (see section "Introduction – IEEE 802.15.4-2006 Frame Format" on page 61). It is recommended to use a frame of maximum length (127 bytes) and arbitrary PSDU data for the PRBS mode. The SHR and the PHR are not transmitted. The transmission starts with the PSDU data and is repeated continuously.

9.9.2 Configuration

All register configurations shall be setup as follows before enabling Continuous Transmission Test Mode:

- TX channel setting (optional);
- TX output power setting (optional);
- Mode selection (PRBS / CW);

An access to the registers TST_CTRL_DIGI and PART_NUM enables the Continuous Transmission Test Mode.

The transmission is started by enabling the PLL (TRX_CMD = PLL_ON) and writing the TX_START command to register TRX_STATE.

Even for CW signal transmission it is required to write valid PSDU data to the Frame Buffer. For PRBS mode it is recommended to write a frame of maximum length.

The detailed programming sequence is shown in Table 9-26 below. The column R/W informs about writing (W) or reading (R) a register or the Frame Buffer.

Table 9-26. Continuous Transmission Programming Sequence

Step	Action	Register	R/W	Value	Description
1	RESET				Reset the transceiver
2	Register Access	IRQ_MASK	W	0x01	Set IRQ mask register, enable PLL_LOCK interrupt and set global AVR IRQ enable
3	Register Access	TRX_CTRL_1	W	0x00	Disable TX_AUTO_CRC_ON
4	Register Access	TRX_STATE	W	0x03	Set radio transceiver state TRX_OFF
5	Register Access	PHY_CC_CCA	W	0x33	Set IEEE 802.15.4 CHANNEL, e.g. 19
6	Register Access	PHY_TX_PWR	W	0x00	Set TX output power, e.g. to P_{max}
7	Register Access	TRX_STATUS	R	0x08	Verify TRX_OFF state
8	Register Access	TST_CTRL_DIGI	W	0x0F	Enable Continuous Transmission Test Mode – step # 1
9[1]	Register Access	TRX_CTRL_2	W	0x03	Enable High Data Rate Mode, 2 Mb/s
10[1]	Register Access	RX_CTRL	W	0xA7	Configure High Data Rate Mode
11[2]	Frame Buffer Write Access		W		Write PSDU data (even for CW mode), refer to Table 9-27 on page 98
12	Register Access	PART_NUM	W	0x54	Enable Continuous Transmission Test Mode – step # 2
13	Register Access	PART_NUM	W	0x46	Enable Continuous Transmission Test Mode – step # 3
14	Register Access	TRX_STATE	W	0x09	Enable PLL_ON state
15	Interrupt event		R	0x01	Wait for PLL_LOCK interrupt
16	Register Access	TRX_STATE	W	0x02	Initiate Transmission, enter BUSY_TX state
17	Measurement				Perform measurement
18	Register Access	TRX_CTRL_2	W	0x00	Disable Continuous Transmission Test Mode
19	RESET				Reset the transceiver

Notes: 1. Only required for CW mode, do not configure for PRBS mode.
2. Frame Buffer content varies for different modulation schemes.

The content of the Frame Buffer has to be defined for Continuous Transmission PRBS mode or CW mode. To measure the power spectral density (PSD) mask of the transmitter it is recommended to use a random sequence of maximum length for the PSDU data.

To measure CW signals it is necessary to write either 0x00 or 0xFF to the Frame Buffer. For details refer to Table 9-27 below.

Table 9-27. Frame Buffer Content for various Continuous Transmission Modulation Schemes

Step	Action	Frame Content	Comment
11	Frame Buffer Write Access	Random Sequence	modulated RF signal
		0x00 (each byte)	f_{CH} – 0.5 MHz, CW signal
		0xFF (each byte)	f_{CH} + 0.5 MHz, CW signal

9.10 Abbreviations

AACK	-	Automatic acknowledgement
ACK	-	Acknowledgement
ADC	-	Analog-to-digital converter
AD	-	Antenna diversity
AGC	-	Automated gain control
AES	-	Advanced encryption standard
ARET	-	Automatic retransmission
AVREG	-	Voltage regulator for analog building blocks
AWGN	-	Additive White Gaussian Noise
BATMON	-	Battery monitor
BBP	-	Base band processor
BPF	-	Band pass filter
CBC	-	Cipher block chaining
CRC	-	Cyclic redundancy check
CCA	-	Clear channel assessment
CSMA-CA	-	Carrier sense multiple access/Collision avoidance
CW	-	Continuous wave
DVREG	-	Voltage regulator for digital building blocks
ECB	-	Electronic code book
ED	-	Energy detection
ESD	-	Electro static discharge
EVM	-	Error vector magnitude
FCF	-	Frame control field
FCS	-	Frame check sequence
FIFO	-	First in first out
FTN	-	Filter tuning network
GPIO	-	General purpose input output

ISM	-	Industrial, scientific, and medical
LDO	-	Low-drop output
LNA	-	Low-noise amplifier
LO	-	Local oscillator
LQI	-	Link quality indicator
LSB	-	Least significant bit
MAC	-	Medium access control
MFR	-	MAC footer
MHR	-	MAC header
MSB	-	Most significant bit
MSDU	-	MAC service data unit
MPDU	-	MAC protocol data unit
MSK	-	Minimum shift keying
O-QPSK	-	Offset - quadrature phase shift keying
PA	-	Power amplifier
PAN	-	Personal area network
PCB	-	Printed circuit board
PER	-	Packet error rate
PHR	-	PHY header
PHY	-	Physical layer
PLL	-	Phase locked loop
POR	-	Power-on reset
PPF	-	Poly-phase filter
PRBS	-	Pseudo random bit sequence
PSDU	-	PHY service data unit
PSD	-	Power spectral mask
QFN	-	Quad flat no-lead package
RF	-	Radio frequency
RSSI	-	Received signal strength indicator
RX	-	Receiver
SFD	-	Start-of-frame delimiter
SHR	-	Synchronization header
SPI	-	Serial peripheral interface
SRAM	-	Static random access memory
SSBF	-	Single side band filter
TX	-	Transmitter

VCO	-	Voltage controlled oscillator
VREG	-	Voltage regulator
XOSC	-	Crystal oscillator

9.11 Reference Documents

[1] IEEE Std 802.15.4™-2006: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)

[2] IEEE Std 802.15.4™-2003: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)

[3] ANSI / ESD-STM5.1-2001: ESD Association Standard Test Method for electrostatic discharge sensitivity testing – Human Body Model (HBM).

[4] ESD-STM5.3.1-1999: ESD Association Standard Test Method for electrostatic discharge sensitivity testing – Charged Device Model (CDM).

[5] NIST FIPS PUB 197: Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, US Department of Commerce/NIST, November 26, 2001

[6] ATmega128RFA1 Software Programming Model

9.12 Register Description

9.12.1 AES_CTRL – AES Control Register

Bit	7	6	5	4	3	2	1	0	
NA ($13C)	AES_REQUEST	Res	AES_MODE	Res	AES_DIR	AES_IM	Res1	Res0	AES_CTRL
Read/Write	RW	R	RW	R	RW	RW	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register controls the operation of the security module. Do not access this register during AES operation to read the AES core status. A read or write access to the register stops the ongoing processing. To read the AES status use bit AES_DONE of register AES_STATUS. Note that the AES_CTRL register is cleared when entering the radio transceiver SLEEP state.

- **Bit 7 – AES_REQUEST - Request AES Operation.**

A write access with AES_REQUEST = 1 initiates the AES operation.

- **Bit 6 – Res - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 5 – AES_MODE - Set AES Operation Mode**

This register bit sets the AES operation mode (ECB/CBC Mode).

Table 9-28 AES_MODE Register Bits

Register Bits	Value	Description
AES_MODE	0	AES Mode is ECB (Electronic Code Book).
	1	AES Mode is CBC (Cipher Block Chaining).

- **Bit 4 – Res - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 3 – AES_DIR - Set AES Operation Direction**

This register bit sets the AES operation direction to either encryption or decryption.

Table 9-29 AES_DIR Register Bits

Register Bits	Value	Description
AES_DIR	0	AES operation is encryption.
	1	AES operation is decryption.

- **Bit 2 – AES_IM - AES Interrupt Enable**

This register bit is used to enable the AES interrupt.

- **Bit 1:0 – Res1:0 - Reserved Bit**

These bits are reserved for future use. The result of a read access is undefined. The register bits must always be written with the reset value.

9.12.2 AES_STATUS – AES Status Register

Bit	7	6	5	4	3	2	1	0	
NA ($13D)	AES_ER	Res5	Res4	Res3	Res2	Res1	Res0	AES_DONE	AES_STATUS
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This read-only register signals the status of the security module and operation. Note that the AES_STATUS register is cleared when entering the radio transceiver SLEEP state.

- **Bit 7 – AES_ER - AES Operation Finished with Error**

This register bit indicates an error during AES module run. An error occurs if accessing AES_CTRL while an AES operation is running or if AES_KEY or AES_STATE Memory is not loaded completely or less than 16 Byte read from AES_STATE.

- **Bit 6:1 – Res5:0 - Reserved**

These bits are reserved for future use.

- **Bit 0 – AES_DONE - AES Operation Finished with Success**

This register bit indicates a successfully finished operation of the AES module.

9.12.3 AES_STATE – AES Plain and Cipher Text Buffer Register

Bit	7	6	5	4	3	2	1	0	
NA ($13E)	AES_STATE7:0								AES_STATE
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The AES_STATE register accesses a 16 byte internal data buffer. The buffer is accessed by reading or writing 16 times to the same address location (AES_STATE). If the buffer is not completely read or written an error occurs when an AES operation is started. Note that the AES_STATE register is cleared when entering the radio transceiver SLEEP state.

- **Bit 7:0 – AES_STATE7:0 - AES Plain and Cipher Text Buffer**

These bits represent the data buffer for the AES operation.

9.12.4 AES_KEY – AES Encryption and Decryption Key Buffer Register

Bit	7	6	5	4	3	2	1	0	
NA ($13F)	AES_KEY7:0								AES_KEY
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The AES key register accesses a 128 Bit internal buffer that holds the Encryption or Decryption Key. The AES_KEY buffer is a 16 Byte buffer. The buffer is accessed by reading or writing 16 fold to the same address location (AES_KEY). A read access to registers AES_KEY returns the last round key of the preceding security operation. This is the key that is required for the corresponding ECB decryption operation after an ECB encryption operation. However, the initial AES key written to the security module in advance of an AES run is not modified during an AES operation. This initial key is used for the next AES run even if it cannot be read from AES_KEY register. Note that the AES_KEY register is cleared when entering the radio transceiver SLEEP state.

- **Bit 7:0 – AES_KEY7:0 - AES Encryption/Decryption Key Buffer**

These bits represent the data buffer for the AES Encryption/Decryption key.

9.12.5 TRX_STATUS – Transceiver Status Register

Bit	7	6	5	4	
NA ($141)	CCA_DONE	CCA_STATUS	TST_STATUS	TRX_STATUS4	TRX_STATUS
Read/Write	R	R	R	R	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($141)	TRX_STATUS3	TRX_STATUS2	TRX_STATUS1	TRX_STATUS0	TRX_STATUS
Read/Write	R	R	R	R	
Initial Value	0	0	0	0	

This read-only register signals the present state of the radio transceiver as well as the status of the CCA operation. A state change is initiated by writing a state transition command to the TRX_CMD bits of register TRX_STATE. The register is not accessible in SLEEP state.

- **Bit 7 – CCA_DONE - CCA Algorithm Status**

This bit indicates if a CCA request is completed. This is also indicated by a TRX24_CCA_ED_DONE interrupt. Note that register bit CCA_DONE is cleared in response to a CCA_REQUEST.

Table 9-30 CCA_DONE Register Bits

Register Bits	Value	Description
CCA_DONE	0	CCA calculation not finished
	1	CCA calculation finished

- **Bit 6 – CCA_STATUS - CCA Status Result**

The result of the CCA measurement is available in register bit CCA_STATUS after a CCA request is completed. Note that register bit CCA_STATUS is cleared in response to a CCA_REQUEST.

Table 9-31 CCA_STATUS Register Bits

Register Bits	Value	Description
CCA_STATUS	0	Channel indicated as busy.
	1	Channel indicated as idle.

- **Bit 5 – TST_STATUS - Test mode status**

This bit is reserved for internal use. It indicates the status of the test mode.

Table 9-32 TST_STATUS Register Bits

Register Bits	Value	Description
TST_STATUS	0	Test mode is disabled.
	1	Test mode is active.

- **Bit 4:0 – TRX_STATUS4:0 - Transceiver Main Status**

The register bits TRX_STATUS signal the current radio transceiver status. Do not try to initiate a further state change while the radio transceiver is in STATE_TRANSITION_IN_PROGRESS state. Values not listed in the following table are reserved.

Table 9-33 TRX_STATUS Register Bits

Register Bits	Value	Description
TRX_STATUS4:0		
	0x01	BUSY_RX
	0x02	BUSY_TX
	0x06	RX_ON
	0x08	TRX_OFF
	0x09	PLL_ON
	0x0F	SLEEP
	0x11	BUSY_RX_AACK
	0x12	BUSY_TX_ARET
	0x16	RX_AACK_ON
	0x19	TX_ARET_ON

Register Bits	Value	Description
	0x1F	STATE_TRANSITION_IN_PROGRESS

9.12.6 TRX_STATE – Transceiver State Control Register

Bit	7	6	5	4	
NA ($142)	TRAC_STATUS2	TRAC_STATUS1	TRAC_STATUS0	TRX_CMD4	TRX_STATE
Read/Write	R	R	R	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($142)	TRX_CMD3	TRX_CMD2	TRX_CMD1	TRX_CMD0	TRX_STATE
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

The states of the radio transceiver are controlled via register TRX_STATE using register bits TRX_CMD. The read-only register bits TRAC_STATUS indicate the status or result of an Extended Operating Mode transaction. A successful state transition shall be confirmed by reading register bits TRX_STATUS. This register is used for both Basic and Extended Operating Mode.

- **Bit 7:5 – TRAC_STATUS2:0 - Transaction Status**

The status of the RX_AACK and TX_ARET procedure is indicated by register bits TRAC_STATUS. TRAC_STATUS is only valid in Extended Operating Modes. Details of the algorithm and a description of the status information are given in the RX_AACK_ON and TX_ARET_ON sections of the data-sheet. Even though the reset value for register bits TRAC_STATUS is 0, the RX_AACK and TX_ARET procedures set the register bits to TRAC_STATUS = 7 (INVALID) when it is started. Not all status values are used in both RX_AACK and TX_ARET transactions. In TX_ARET the status SUCCESS_DATA_PENDING indicates a successful reception of an ACK frame with frame pending bit set to 1. In RX_AACK the status SUCCESS_WAIT_FOR_ACK indicates an ACK frame is about to sent in RX_AACK slotted acknowledgment. Slotted acknowledgment operation must be enabled with the SLOTTED_OPERATION bit of register XAH_CTRL_0. The application software must set the SLPTR bit of register TRXPWR at the next back-off slot boundary in order to initiate a transmission of the ACK frame. For details refer to IEEE 802.15.4-2006, chapter 5.5.4.1. Values not listed in the following table are reserved.

Table 9-34 TRAC_STATUS Register Bits

Register Bits	Value	Description
TRAC_STATUS2:0	0	SUCCESS (RX_AACK, TX_ARET)
	1	SUCCESS_DATA_PENDING (TX_ARET)
	2	SUCCESS_WAIT_FOR_ACK (RX_AACK)
	3	CHANNEL_ACCESS_FAILURE (TX_ARET)
	5	NO_ACK (TX_ARET)
	7	INVALID (RX_AACK, TX_ARET)

- **Bit 4:0 – TRX_CMD4:0 - State Control Command**

A write access to register bits TRX_CMD initiates a state transition of the radio transceiver towards the new state as defined by the write access. Do not try to initiate a further state change while the radio transceiver is in STATE_TRANSITION_IN_PROGRESS state (see TRX_STATUS register). FORCE_PLL_ON is not valid for the SLEEP state as well as during STATE_TRANSITION_IN_PROGRESS towards the SLEEP state. Values not listed in the following table are reserved and mapped to NOP.

Table 9-35 TRX_CMD Register Bits

Register Bits	Value	Description
TRX_CMD4:0	0x00	NOP
	0x02	TX_START
	0x03	FORCE_TRX_OFF
	0x04	FORCE_PLL_ON
	0x06	RX_ON
	0x08	TRX_OFF
	0x09	PLL_ON (TX_ON)
	0x16	RX_AACK_ON
	0x19	TX_ARET_ON

9.12.7 TRX_CTRL_0 – Reserved

Bit	7	6	5	4	3	2	1	0	
NA ($143)	Res7	Res6	Res5	Res4	Res3	Res2	Res1	Res0	TRX_CTRL_0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	1	1	0	0	1	

This register is reserved for future use.

- **Bit 7:0 – Res7:0 - Reserved**

These bits are reserved for future use.

9.12.8 TRX_CTRL_1 – Transceiver Control Register 1

Bit	7	6	5	4	
NA ($144)	PA_EXT_EN	IRQ_2_EXT_EN	TX_AUTO_CRC_ON	Res4	TRX_CTRL_1
Read/Write	RW	RW	RW	R	
Initial Value	0	0	1	0	
Bit	3	2	1	0	
NA ($144)	Res3	Res2	Res1	Res0	TRX_CTRL_1
Read/Write	R	R	R	R	
Initial Value	0	0	0	0	

The TRX_CTRL_1 register is a multi purpose register to control various operating modes and settings of the radio transceiver.

- **Bit 7 – PA_EXT_EN - External PA support enable**

This register bit enables pin DIG3 and pin DIG4 to indicate the transmit state of the radio transceiver. The control of the external RF front-end is disabled when this bit is 0. Both pins DIG3 and DIG4 are then low. The control of the external front-end is enabled when this bit is 1. DIG3 and DIG4 then indicate the state of the radio transceiver. Pin DIG3 is high and pin DIG4 is low in the state TX_BUSY. In all other states pin DIG3 is low and pin DIG4 is high. It is recommended to set PA_EXT_EN=1 only in receive or transmit states to reduce the power consumption or avoid leakage current of external RF switches or other building blocks especially during SLEEP state.

- **Bit 6 – IRQ_2_EXT_EN - Connect Frame Start IRQ to TC1**

When this bit is set to one the capture input of Timer/Counter 1 is connected to the RX frame start signal and pin DIG2 becomes an output, driving the RX frame start signal. Antenna Diversity RF switch control (ANT_EXT_SW_EN=1) shall not be used at the same time, because it shares the same device pin. The function IRQ_2_EXT_EN is available for alternate frame time stamping using Timer/Counter 1. In general the preferred method for frame time stamping is using the symbol counter.

- **Bit 5 – TX_AUTO_CRC_ON - Enable Automatic CRC Calculation**

This register bit controls the automatic FCS generation for TX operations. The automatic FCS algorithm is performed autonomously by the radio transceiver if register bit TX_AUTO_CRC_ON=1.

- **Bit 4:0 – Res4:0 - Reserved**

9.12.9 PHY_TX_PWR – Transceiver Transmit Power Control Register

Bit	7	6	5	4	
NA ($145)	PA_BUF_LT1	PA_BUF_LT0	PA_LT1	PA_LT0	PHY_TX_PWR
Read/Write	RW	RW	RW	RW	
Initial Value	1	1	0	0	
Bit	3	2	1	0	
NA ($145)	TX_PWR3	TX_PWR2	TX_PWR1	TX_PWR0	PHY_TX_PWR
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

This register controls the output power and the ramping of the transmitter.

- **Bit 7:6 – PA_BUF_LT1:0 - Power Amplifier Buffer Lead Time**

These register bits control the enable lead time of the internal PA buffer relative to the enable time of the internal PA. This time is further used to derive a control signal for an external RF front-end to switch between receive and transmit.

Table 9-36 PA_BUF_LT Register Bits

Register Bits	Value	Description
PA_BUF_LT1:0	0	0 µs
	1	2 µs
	2	4 µs
	3	6 µs

- **Bit 5:4 – PA_LT1:0 - Power Amplifier Lead Time**

These register bits control the enable lead time of the internal power amplifier relative to the beginning of the transmitted frame (SHR).

Table 9-37 PA_LT Register Bits

Register Bits	Value	Description
PA_LT1:0	0	2 µs
	1	4 µs
	2	6 µs
	3	8 µs

- **Bit 3:0 – TX_PWR3:0 - Transmit Power Setting**

These register bits determine the TX output power of the radio transceiver.

Table 9-38 TX_PWR Register Bits

Register Bits	Value	Description
TX_PWR3:0	0	3.0 dBm
	1	2.8 dBm
	2	2.3 dBm
	3	1.8 dBm
	4	1.3 dBm
	5	0.7 dBm
	6	0.0 dBm
	7	-1 dBm
	8	-2 dBm
	9	-3 dBm
	10	-4 dBm
	11	-5 dBm
	12	-7 dBm
	13	-9 dBm
	14	-12 dBm
	15	-17 dBm

9.12.10 PHY_RSSI – Receiver Signal Strength Indicator Register

Bit	7	6	5	4	
NA ($146)	RX_CRC_VALID	RND_VALUE1	RND_VALUE0	RSSI4	PHY_RSSI
Read/Write	R	R	R	R	
Initial Value	0	0	0	0	

Bit	3	2	1	0	
NA ($146)	RSSI3	RSSI2	RSSI1	RSSI0	PHY_RSSI
Read/Write	R	R	R	R	
Initial Value	0	0	0	0	

The PHY_RSSI register is a multi purpose register that indicates FCS validity, provides random numbers and shows the current RSSI value.

- **Bit 7 – RX_CRC_VALID - Received Frame CRC Status**

Reading this register bit indicates whether the last received frame has a valid FCS or not. The register bit is updated when issuing a TRX24_RX_END interrupt and remains valid until the next TRX24_RX_END interrupt is issued, caused by a new frame reception.

Table 9-39 RX_CRC_VALID Register Bits

Register Bits	Value	Description
RX_CRC_VALID	0	CRC (FCS) not valid
	1	CRC (FCS) valid

- **Bit 6:5 – RND_VALUE1:0 - Random Value**

A 2-bit random value can be retrieved by reading register bits RND_VALUE. The value can be used for random numbers for security applications. Note that the radio transceiver shall be in Basic Operating Mode receive state. The values are updated every 1 µs.

- **Bit 4:0 – RSSI4:0 - Receiver Signal Strength Indicator**

The result of the automated RSSI measurement is stored in these register bits. The value is updated every 2µs in receive states. The read value is a number between 0 and 28 indicating the received signal strength as a linear curve on a logarithmic input power scale (dBm) with a resolution of 3 dB. A RSSI value of 0 indicates a RF input power lower than RSSI_BASE_VAL (-90 dBm). A value of 28 marks a power higher or equal to -10 dBm.

Table 9-40 RSSI Register Bits

Register Bits	Value	Description
RSSI4:0	0	Minimum RSSI value: P(RF) < -90 dBm
	1	P(RF) = RSSI_BASE_VAL+3 · (RSSI-1) [dBm]
	2	...
	28	Maximum RSSI value: P(RF) ≥ -10 dBm

9.12.11 PHY_ED_LEVEL – Transceiver Energy Detection Level Register

Bit	7	6	5	4	
NA ($147)	ED_LEVEL7	ED_LEVEL6	ED_LEVEL5	ED_LEVEL4	PHY_ED_LEVEL
Read/Write	R	R	R	R	
Initial Value	1	1	1	1	

Bit	3	2	1	0	
NA ($147)	ED_LEVEL3	ED_LEVEL2	ED_LEVEL1	ED_LEVEL0	PHY_ED_LEVEL
Read/Write	R	R	R	R	
Initial Value	1	1	1	1	

This register contains the result of an Energy Detection measurement.

- **Bit 7:0 – ED_LEVEL7:0 - Energy Detection Level**

The minimum ED value (ED_LEVEL = 0) indicates a receiver power less than or equal to RSSI_BASE_VAL. The range is 84 dB with a resolution of 1 dB and an absolute accuracy of ±5 dB. A manual ED measurement can be initiated by a write access to this register. A value of 0xFF signals that no measurement has yet been started (reset value). The measurement duration is 8 symbol periods (128 µs) for a data rate of 250 kb/s. For High Data Rate Modes the automated measurement duration is reduced to 32 µs. For manually initiated ED measurements in these modes the measurement period is still 128 µs as long as the receiver is in RX_ON state. A value other than 0xFF indicates the result of the last ED measurement.

Table 9-41 ED_LEVEL Register Bits

Register Bits	Value	Description
ED_LEVEL7:0	0x00	Minimum result of last ED measurement
	0x01	P(RF) = RSSI_BASE_VAL+ED [dBm]
	0x02	...
	0x54	Maximum result of last ED measurement
	0xFF	Reset value

9.12.12 PHY_CC_CCA – Transceiver Clear Channel Assessment (CCA) Control Register

Bit	7	6	5	4	
NA ($148)	CCA_REQUEST	CCA_MODE1	CCA_MODE0	CHANNEL4	PHY_CC_CCA
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	1	0	
Bit	3	2	1	0	
NA ($148)	CHANNEL3	CHANNEL2	CHANNEL1	CHANNEL0	PHY_CC_CCA
Read/Write	RW	RW	RW	RW	
Initial Value	1	0	1	1	

This register is provided to initiate and control a CCA measurement.

- **Bit 7 – CCA_REQUEST - Manual CCA Measurement Request**

A manual CCA measurement is initiated with setting CCA_REQUEST=1. The end of the CCA measurement is indicated by the TRX24_CCA_ED_DONE interrupt. Register bits CCA_DONE and CCA_STATUS of register TRX_STATUS are updated after a CCA_REQUEST. The register bit is automatically cleared after requesting a CCA measurement with CCA_REQUEST=1.

- **Bit 6:5 – CCA_MODE1:0 - Select CCA Measurement Mode**

The CCA mode can be selected using these register bits. Note that IEEE 802.15.4-2006 CCA Mode 3 defines the logical combination of CCA Mode 1 and 2 with the logical operators AND or OR. This can be selected with CCA_MODE=0 for logical operation OR and CCA_MODE=3 for logical operation AND.

Table 9-42 CCA_MODE Register Bits

Register Bits	Value	Description
CCA_MODE1:0	0	Mode 3a, Carrier sense OR energy above threshold
	1	Mode 1, Energy above threshold
	2	Mode 2, Carrier sense only
	3	Mode 3b, Carrier sense AND energy above threshold

- **Bit 4:0 – CHANNEL4:0 - RX/TX Channel Selection**

These register bits define the RX/TX channel. The channel assignment is according to IEEE 802.15.4.

Table 9-43 CHANNEL Register Bits

Register Bits	Value	Description
CHANNEL4:0	11	2405 MHz
	12	2410 MHz
	13	2415 MHz
	14	2420 MHz
	15	2425 MHz
	16	2430 MHz
	17	2435 MHz
	18	2440 MHz
	19	2445 MHz
	20	2450 MHz
	21	2455 MHz
	22	2460 MHz
	23	2465 MHz
	24	2470 MHz
	25	2475 MHz
	26	2480 MHz

9.12.13 CCA_THRES – Transceiver CCA Threshold Setting Register

Bit	7	6	
NA ($149)	CCA_CS_THRES3	CCA_CS_THRES2	CCA_THRES
Read/Write	RW	RW	
Initial Value	1	1	
Bit	5	4	
NA ($149)	CCA_CS_THRES1	CCA_CS_THRES0	CCA_THRES
Read/Write	RW	RW	
Initial Value	0	0	
Bit	3	2	
NA ($149)	CCA_ED_THRES3	CCA_ED_THRES2	CCA_THRES
Read/Write	RW	RW	
Initial Value	0	1	
Bit	1	0	
NA ($149)	CCA_ED_THRES1	CCA_ED_THRES0	CCA_THRES
Read/Write	RW	RW	
Initial Value	1	1	

This register sets the threshold level for the Energy Detection (ED) of the Clear Channel Assessment (CCA).

- **Bit 7:4 – CCA_CS_THRES3:0 - CS Threshold Level for CCA Measurement**

These bits are reserved for internal use.

- **Bit 3:0 – CCA_ED_THRES3:0 - ED Threshold Level for CCA Measurement**

These bits define the received power threshold of the Energy above threshold algorithm. The threshold is calculated by RSSI_BASE_VAL + 2CCA_ED_THRES [dBm]. Any received power above this level is interpreted as a busy channel.

9.12.14 RX_CTRL – Transceiver Receive Control Register

Bit	7	6	5	4	
NA ($14A)	Resx7	Resx6	Resx5	Resx4	RX_CTRL
Read/Write	RW	RW	RW	RW	
Initial Value	1	0	1	1	

Bit	3	2	1	0	
NA ($14A)	PDT_THRES3	PDT_THRES2	PDT_THRES1	PDT_THRES0	RX_CTRL
Read/Write	RW	RW	RW	RW	
Initial Value	0	1	1	1	

The register controls the sensitivity of the Antenna Diversity Mode. Note that in High Data Rate modes the ACR module will always be disabled.

- **Bit 7:4 – Resx7:4 - Reserved**
- **Bit 3:0 – PDT_THRES3:0 - Receiver Sensitivity Control**

These register bits control the sensitivity of the receiver correlation unit. If the Antenna Diversity algorithm is enabled the value shall be set to PDT_THRES = 3. Otherwise it shall be set back to the reset value. Values not listed in the following table are reserved.

Table 9-44 PDT_THRES Register Bits

Register Bits	Value	Description
PDT_THRES3:0	0x7	Reset value, to be used if Antenna Diversity algorithm is disabled
	0x3	Recommended correlator threshold for Antenna Diversity operation

9.12.15 SFD_VALUE – Start of Frame Delimiter Value Register

Bit	7	6	5	4	3	2	1	0	
NA ($14B)	SFD_VALUE7:0								SFD_VALUE
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	1	0	1	0	0	1	1	1	

This register contains the one octet start-of-frame delimiter (SFD) to synchronize to a received frame. The lower 4 bits must not be all zero to avoid decoding conflicts.

- **Bit 7:0 – SFD_VALUE7:0 - Start of Frame Delimiter Value**

For compliant IEEE 802.15.4 networks set SFD_VALUE = 0xA7. This is the default value of the register. To establish non IEEE 802.15.4 compliant networks the SFD value can be changed to any other value. If enabled a RX_START interrupt is issued only if the received SFD matches the register content of SFD_VALUE and a valid PHR is received.

Table 9-45 SFD_VALUE Register Bits

Register Bits	Value	Description
SFD_VALUE7:0	0xA7	IEEE 802.15.4 compliant value of the SFD

9.12.16 TRX_CTRL_2 – Transceiver Control Register 2

Bit	7	6	5	4	
NA ($14C)	RX_SAFE_MODE	Res4	Res3	Res2	TRX_CTRL_2
Read/Write	RW	R	R	R	
Initial Value	0	0	0	0	

Bit	3	2	1	0	
NA ($14C)	Res1	Res0	OQPSK_DATA_RATE1	OQPSK_DATA_RATE0	TRX_CTRL_2
Read/Write	R	R	RW	RW	
Initial Value	0	0	0	0	

This register controls the data rate setting of the radio transceiver.

- **Bit 7 – RX_SAFE_MODE - RX Safe Mode**

If this bit is set, the next received frame will be protected and not overwritten by following frames. Set this bit to 0 to release the buffer (and set it again for further protection).

- **Bit 6:2 – Res4:0 - Reserved**
- **Bit 1:0 – OQPSK_DATA_RATE1:0 - Data Rate Selection**

A write access to these register bits sets the OQPSK PSDU data rate used by the radio transceiver. The reset value OQPSK_DATA_RATE = 0 is the PSDU data rate according to IEEE 802.15.4. All other values are used in High Data Rate Modes.

Table 9-46 OQPSK_DATA_RATE Register Bits

Register Bits	Value	Description
OQPSK_DATA_RATE1:0	0	250 kb/s (IEEE 802.15.4 compliant)
	1	500 kb/s
	2	1000 kb/s
	3	2000 kb/s

9.12.17 ANT_DIV – Antenna Diversity Control Register

Bit	7	6	5	4	
NA ($14D)	ANT_SEL	Res2	Res1	Res0	ANT_DIV
Read/Write	R	R	R	R	
Initial Value	0	0	0	0	

Bit	3	2	1	0	
NA ($14D)	ANT_DIV_EN	ANT_EXT_SW_EN	ANT_CTRL1	ANT_CTRL0	ANT_DIV
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	1	1	

This register controls the Antenna Diversity.

- **Bit 7 – ANT_SEL - Antenna Diversity Antenna Status**

This register bit signals the currently selected antenna path. The selection may be based either on the last antenna diversity cycle (ANT_DIV_EN = 1) or on the content of register bits ANT_CTRL.

Table 9-47 ANT_SEL Register Bits

Register Bits	Value	Description
ANT_SEL	0	Antenna 0
	1	Antenna 1

- **Bit 6:4 – Res2:0 - Reserved**
- **Bit 3 – ANT_DIV_EN - Enable Antenna Diversity**

If this register bit is set the Antenna Diversity algorithm is enabled. On reception of a frame the algorithm selects an antenna autonomously during SHR search. This selection is kept until

1. a new SHR search starts or

2. receive states are left or

3. a manually programming of bits ANT_CTRL occurred. If ANT_DIV_EN = 1 the bit ANT_EXT_SW_EN shall also be set to 1.

Table 9-48 ANT_DIV_EN Register Bits

Register Bits	Value	Description
ANT_DIV_EN	0	Antenna Diversity algorithm disabled
	1	Antenna Diversity algorithm enabled

- **Bit 2 – ANT_EXT_SW_EN - Enable External Antenna Switch Control**

If enabled, pin DIG1 and pin DIG2 become output pins and provide a differential control signal for an external Antenna Diversity switch. The selection of a specific antenna is done either by the automatic Antenna Diversity algorithm (ANT_DIV_EN = 1) or according to bits ANT_CTRL if the Antenna Diversity algorithm is disabled. Do not enable Antenna Diversity RF switch control (ANT_EXT_SW_EN = 1) and RX Frame Time Stamping (IRQ_2_EXT_EN = 1, see register TRX_CTRL_1) at the same time. If this bit is set the control pins DIG1/DIG2 are activated in all radio transceiver states as long as bit ANT_EXT_SW_EN is also set. If the radio transceiver is not in a receive or transmit state, it is recommended to disable bit ANT_EXT_SW_EN to reduce the power consumption or avoid leakage current of an external RF switch especially during SLEEP state. The output pins DIG1 and DIG2 are pulled-down to digital ground if bit ANT_EXT_SW_EN = 0.

Table 9-49 ANT_EXT_SW_EN Register Bits

Register Bits	Value	Description
ANT_EXT_SW_EN	0	Antenna Diversity RF switch control disabled
	1	Antenna Diversity RF switch control enabled

- **Bit 1:0 – ANT_CTRL1:0 - Static Antenna Diversity Switch Control**

These bits provide a static control of an Antenna Diversity switch. This register setting defines the selected antenna if ANT_DIV_EN is set to 0 (Antenna Diversity disabled). Register values 1 and 2 are valid for ANT_EXT_SW_EN = 1.

Table 9-50 ANT_CTRL Register Bits

Register Bits	Value	Description
ANT_CTRL1:0	0	Reserved
	1	Antenna 1: DIG1=H, DIG2=L

Register Bits	Value	Description
	2	Antenna 0: DIG1=L, DIG2=H
	3	Default value for ANT_EXT_SW_EN=0; Mandatory setting for applications not using Antenna Diversity

9.12.18 IRQ_MASK – Transceiver Interrupt Enable Register

Bit	7	6	5	4	
NA ($14E)	AWAKE_EN	TX_END_EN	AMI_EN	CCA_ED_DONE_EN	IRQ_MASK
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($14E)	RX_END_EN	RX_START_EN	PLL_UNLOCK_EN	PLL_LOCK_EN	IRQ_MASK
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

This register is used to enable or disable individual interrupts of the radio transceiver. An interrupt is enabled if the corresponding bit is set to 1. All interrupts are disabled after the power up sequence or reset. If an interrupt is enabled it is recommended to read the interrupt status register IRQ_STATUS first to clear the history.

- **Bit 7 – AWAKE_EN - Awake Interrupt Enable**
- **Bit 6 – TX_END_EN - TX_END Interrupt Enable**
- **Bit 5 – AMI_EN - Address Match Interrupt Enable**
- **Bit 4 – CCA_ED_DONE_EN - End of ED Measurement Interrupt Enable**
- **Bit 3 – RX_END_EN - RX_END Interrupt Enable**
- **Bit 2 – RX_START_EN - RX_START Interrupt Enable**
- **Bit 1 – PLL_UNLOCK_EN - PLL Unlock Interrupt Enable**
- **Bit 0 – PLL_LOCK_EN - PLL Lock Interrupt Enable**

9.12.19 IRQ_STATUS – Transceiver Interrupt Status Register

Bit	7	6	5	4	
NA ($14F)	AWAKE	TX_END	AMI	CCA_ED_DONE	IRQ_STATUS
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($14F)	RX_END	RX_START	PLL_UNLOCK	PLL_LOCK	IRQ_STATUS
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

This register contains the status of the pending interrupt requests. An interrupt is pending if the associated bit has a value of one. Such a pending interrupts can be manually cleared by writing a 1 to that register bit. Interrupts are automatically cleared when the corresponding interrupt service routine is being executed.

- **Bit 7 – AWAKE - Awake Interrupt Status**

- **Bit 6 – TX_END - TX_END Interrupt Status**
- **Bit 5 – AMI - Address Match Interrupt Status**
- **Bit 4 – CCA_ED_DONE - End of ED Measurement Interrupt Status**
- **Bit 3 – RX_END - RX_END Interrupt Status**
- **Bit 2 – RX_START - RX_START Interrupt Status**
- **Bit 1 – PLL_UNLOCK - PLL Unlock Interrupt Status**
- **Bit 0 – PLL_LOCK - PLL Lock Interrupt Status**

9.12.20 VREG_CTRL – Voltage Regulator Control and Status Register

Bit	7	6	5	4	
NA ($150)	AVREG_EXT	AVDD_OK	Resx5	Resx4	VREG_CTRL
Read/Write	RW	R	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($150)	Resx3	DVDD_OK	Resx1	Resx0	VREG_CTRL
Read/Write	RW	R	RW	RW	
Initial Value	0	0	0	0	

This register controls the use of the voltage regulators and indicates their status.

- **Bit 7 – AVREG_EXT - Use External AVDD Regulator**

If set, this register bit disables the internal analog voltage regulator to apply an external regulated 1.8V supply for the analog building blocks.

Table 9-51 AVREG_EXT Register Bits

Register Bits	Value	Description
AVREG_EXT	0	Internal AVDD voltage regulator for the analog section is enabled.
	1	Internal AVDD voltage regulator is disabled; use external regulated 1.8V supply voltage for the analog section.

- **Bit 6 – AVDD_OK - AVDD Supply Voltage Valid**

This register bit indicates if the internal 1.8V regulated voltage supply AVDD has settled. The bit is set to logic high if AVREG_EXT = 1.

Table 9-52 AVDD_OK Register Bits

Register Bits	Value	Description
AVDD_OK	0	Analog voltage regulator disabled or supply voltage not stable
	1	Analog supply voltage has settled

- **Bit 5:3 – Resx5:3 - Reserved**
- **Bit 2 – DVDD_OK - DVDD Supply Voltage Valid**

This register bit indicates if the internal 1.8V regulated voltage supply DVDD has settled. The bit is set to logic high if DVREG_EXT = 1.

Table 9-53 DVDD_OK Register Bits

Register Bits	Value	Description
DVDD_OK	0	Digital voltage regulator disabled or supply voltage not stable
	1	Digital supply voltage has settled

- **Bit 1:0 – DVREG_TRIM1:0 - Reserved**

Table 9-54 DVREG_TRIM Register Bits

Register Bits	Value	Description
DVREG_TRIM1:0	0	1.80V
	1	1.75V
	2	1.84V
	3	1.88V

9.12.21 BATMON – Battery Monitor Control and Status Register

Bit	7	6	5	4	
NA ($151)	BAT_LOW	BAT_LOW_EN	BATMON_OK	BATMON_HR	BATMON
Read/Write	RW	RW	R	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($151)	BATMON_VTH3	BATMON_VTH2	BATMON_VTH1	BATMON_VTH0	BATMON
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	1	0	

This register configures the battery monitor to observe the supply voltage at EVDD. The status of the EVDD supply voltage is accessible by reading bit BATMON_OK with respect to the actual BATMON settings. Furthermore the Battery Monitor Interrupt can be controlled with the bits BAT_LOW and BAT_LOW_EN similar to the function of the IRQ_STATUS and IRQ_MASK register for other radio transceiver interrupts.

- **Bit 7 – BAT_LOW - Battery Monitor Interrupt Status**

A BATMON Interrupt is pending if this bit is set. Writing one to this bit if it has been at one will clear the interrupt.

- **Bit 6 – BAT_LOW_EN - Battery Monitor Interrupt Enable**

The Battery Monitor Interrupt is enabled if this bit is set to one. The Battery Monitor will not generate an interrupt if this bit is zero.

- **Bit 5 – BATMON_OK - Battery Monitor Status**

The register bit BATMON_OK indicates the level of the external supply voltage with respect to the programmed threshold BATMON_VTH.

Table 9-55 BATMON_OK Register Bits

Register Bits	Value	Description
BATMON_OK	0	The battery voltage is below the threshold.
	1	The battery voltage is above the threshold.

- **Bit 4 – BATMON_HR - Battery Monitor Voltage Range**

This bit sets the range and resolution of the battery monitor.

Table 9-56 BATMON_HR Register Bits

Register Bits	Value	Description
BATMON_HR	0	Enables the low range, see BATMON_VTH
	1	Enables the high range, see BATMON_VTH

- **Bit 3:0 – BATMON_VTH3:0 - Battery Monitor Threshold Voltage**

The threshold values for the battery monitor are set by these register bits according to the following table.

Table 9-57 BATMON_VTH Register Bits

Register Bits	Value	Description
BATMON_VTH3:0	0x0	2.550V (BATMON_HR=1) 1.70V (BATMON_HR=0)
	0x1	2.625V (BATMON_HR=1) 1.75V (BATMON_HR=0)
	0x2	2.700V (BATMON_HR=1) 1.80V (BATMON_HR=0)
	0x3	2.775V (BATMON_HR=1) 1.85V (BATMON_HR=0)
	0x4	2.850V (BATMON_HR=1) 1.90V (BATMON_HR=0)
	0x5	2.925V (BATMON_HR=1) 1.95V (BATMON_HR=0)
	0x6	3.000V (BATMON_HR=1) 2.00V (BATMON_HR=0)
	0x7	3.075V (BATMON_HR=1) 2.05V (BATMON_HR=0)
	0x8	3.150V (BATMON_HR=1) 2.10V (BATMON_HR=0)
	0x9	3.225V (BATMON_HR=1) 2.15V (BATMON_HR=0)
	0xA	3.300V (BATMON_HR=1) 2.20V (BATMON_HR=0)
	0xB	3.375V (BATMON_HR=1) 2.25V (BATMON_HR=0)
	0xC	3.450V (BATMON_HR=1) 2.30V (BATMON_HR=0)
	0xD	3.525V (BATMON_HR=1) 2.35V (BATMON_HR=0)
	0xE	3.600V (BATMON_HR=1) 2.40V (BATMON_HR=0)
	0xF	3.675V (BATMON_HR=1) 2.45V (BATMON_HR=0)

9.12.22 XOSC_CTRL – Crystal Oscillator Control Register

Bit	7	6	5	4	
NA ($152)	XTAL_MODE3	XTAL_MODE2	XTAL_MODE1	XTAL_MODE0	XOSC_CTRL
Read/Write	RW	RW	RW	RW	
Initial Value	1	1	1	1	

Bit	3	2	1	0	
NA ($152)	XTAL_TRIM3	XTAL_TRIM2	XTAL_TRIM1	XTAL_TRIM0	XOSC_CTRL
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

This register controls the operation of the 16MHz crystal oscillator.

- **Bit 7:4 – XTAL_MODE3:0 - Crystal Oscillator Operating Mode**

These register bits set the operating mode of the 16 MHz crystal oscillator. For normal operation the default value is set to XTAL_MODE = 0xF after reset. For use with an external clock source it is recommended to set XTAL_MODE = 0x4.

Table 9-58 XTAL_MODE Register Bits

Register Bits	Value	Description
XTAL_MODE3:0	0x4	Internal crystal oscillator disabled; use external reference frequency.
	0xF	Internal crystal oscillator enabled; amplitude regulation of oscillation enabled.

- **Bit 3:0 – XTAL_TRIM3:0 - Crystal Oscillator Load Capacitance Trimming**

These register bits control two internal capacitance arrays connected to pins XTAL1 and XTAL2. A capacitance value in the range from 0 pF to 4.5 pF is selectable with a resolution of 0.3 pF.

Table 9-59 XTAL_TRIM Register Bits

Register Bits	Value	Description
XTAL_TRIM3:0	0x0	0.0 pF, trimming capacitors disconnected
	0x1	0.3 pF, trimming capacitor switched on
	0x2	...
	0xF	4.5 pF, trimming capacitor switched on

9.12.23 RX_SYN – Transceiver Receiver Sensitivity Control Register

Bit	7	6	
NA ($155)	RX_PDT_DIS	Res2	RX_SYN
Read/Write	RW	R	
Initial Value	0	0	
Bit	5	4	
NA ($155)	Res1	Res0	RX_SYN
Read/Write	R	R	
Initial Value	0	0	
Bit	3	2	
NA ($155)	RX_PDT_LEVEL3	RX_PDT_LEVEL2	RX_SYN
Read/Write	RW	RW	
Initial Value	0	0	
Bit	1	0	
NA ($155)	RX_PDT_LEVEL1	RX_PDT_LEVEL0	RX_SYN
Read/Write	RW	RW	
Initial Value	0	0	

This register controls the sensitivity threshold of the receiver.

- **Bit 7 – RX_PDT_DIS - Prevent Frame Reception**

RX_PDT_DIS = 1 prevents the reception of a frame even if the radio transceiver is in receive modes. An ongoing frame reception is not affected. This operation mode is independent of the setting of register bits RX_PDT_LEVEL.

- **Bit 6:4 – Res2:0 - Reserved**
- **Bit 3:0 – RX_PDT_LEVEL3:0 - Reduce Receiver Sensitivity**

These register bits reduce the receiver sensitivity such that frames with a RSSI level below the RX_PDT_LEVEL threshold level are not received (RX_PDT_LEVEL>0). The threshold level can be calculated according to the following formula: RX_THRES > RSSI_BASE_VAL+3·(RX_PDT_LEVEL-1), for RX_PDT_LEVEL>0. If register bits RX_PDT_LEVEL>0 the current consumption of the receiver in states RX_ON and RX_AACK_ON is reduced by 500 µA. If register bits RX_PDT_LEVEL=0 (reset value) all frames with a valid SHR and PHR are received, independently of their signal strength. Examples for certain register settings are given in the following table.

Table 9-60 RX_PDT_LEVEL Register Bits

Register Bits	Value	Description
RX_PDT_LEVEL3:0	0x0	RX_THRES ≤ RSSI_BASE_VAL (Reset value); RSSI value not considered
	0x1	RX_THRES > RSSI_BASE_VAL + 0 · 3; RSSI > -90 dBm
	0x2	...
	0xE	RX_THRES > RSSI_BASE_VAL + 13 · 3; RSSI > -51 dBm
	0xF	RX_THRES > RSSI_BASE_VAL + 14 · 3; RSSI > -48 dBm

9.12.24 XAH_CTRL_1 – Transceiver Acknowledgment Frame Control Register 1

Bit	7	6	5	4	
NA ($157)	Res1	Res0	AACK_FLTR_RES_FT	AACK_UPLD_RES_FT	XAH_CTRL_1
Read/Write	R	R	RW	RW	
Initial Value	0	0	0	0	

Bit	3	2	1	0	
NA ($157)	Res	AACK_ACK_TIME	AACK_PROM_MODE	Res	XAH_CTRL_1
Read/Write	R	RW	RW	R	
Initial Value	0	0	0	0	

This register is a multi-purpose control register for various RX_AACK settings.

- **Bit 7:6 – Res1:0 - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 5 – AACK_FLTR_RES_FT - Filter Reserved Frames**

This register bit shall only be set if AACK_UPLD_RES_FT = 1. If AACK_FLTR_RES_FT = 1 reserved frame types are filtered similar to data frames as specified in IEEE 802.15.4-2006. Reserved frame types are explained in IEEE 802.15.4 section 7.2.1.1.1. If AACK_FLTR_RES_FT = 0 a received, reserved frame is only checked for a valid FCS.

- **Bit 4 – AACK_UPLD_RES_FT - Process Reserved Frames**

If AACK_UPLD_RES_FT = 1 received frames indicated as reserved are further processed. A RX_END interrupt is generated if the FCS of those frames is valid. In conjunction with the configuration bit AACK_FLTR_RES_FT set, these frames are handled like IEEE 802.15.4 compliant data frames during RX_AACK transaction. An AMI interrupt is issued if the address in the received frame matches the node address. That means if a reserved frame passes the third level filter rules, an acknowledgment frame is generated and transmitted if it was requested by the received frame. If this is not wanted bit AACK_DIS_ACK in register CSMA_SEED_1 has to be set.

- **Bit 3 – Res - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 2 – AACK_ACK_TIME - Reduce Acknowledgment Time**

According to IEEE 802.15.4, section 7.5.6.4.2 the transmission of an acknowledgment frame shall commence 12 symbols (aTurnaroundTime) after the reception of the last symbol of a data or MAC command frame. This is achieved with the reset value of the register bit AACK_ACK_TIME. If AACK_ACK_TIME = 1 an acknowledgment frame is alternatively sent already 2 symbol periods (32 µs) after the reception of the last symbol of a data or MAC command frame. This may be applied to proprietary networks or networks using the High Data Rate Modes to increase battery lifetime and to improve the overall data throughput. This setting affects also to acknowledgment frame response time for slotted acknowledgment operation.

Table 9-61 AACK_ACK_TIME Register Bits

Register Bits	Value	Description
AACK_ACK_TIME	0	12 symbols acknowledgment time
	1	2 symbols acknowledgment time

- **Bit 1 – AACK_PROM_MODE - Enable Promiscuous Mode**

This register bit enables the promiscuous mode within the RX_AACK mode; refer to IEEE 802.15.4-2006 chapter 7.5.6.5. If this bit is set, every incoming frame with a valid PHR finishes with a RX_END interrupt even if the third level filter rules do not match or the FCS is not valid. The bit RX_CRC_VALID of register PHY_RSSI is set accordingly. If this bit is set and a frame passes the third level filter rules, an acknowledgment frame is generated and transmitted unless disabled by bit AACK_DIS_ACK of register CSMA_SEED_1.

- **Bit 0 – Res - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

9.12.25 FTN_CTRL – Transceiver Filter Tuning Control Register

Bit	7	6	5	4	
NA ($158)	FTN_START	Resx6	Resx5	Resx4	FTN_CTRL
Read/Write	RW	RW	RW	RW	
Initial Value	0	1	0	1	

Bit	3	2	1	0	
NA ($158)	Resx3	Resx2	Resx1	Resx0	FTN_CTRL
Read/Write	RW	RW	RW	RW	
Initial Value	1	0	0	0	

This register controls the operation of the calibration loop of the filter tuning network.

- **Bit 7 – FTN_START - Start Calibration Loop of Filter Tuning Network**

FTN_START = 1 initiates the calibration of the filter tuning network. When the calibration cycle has finished after at most 25 µs the register bit is automatically reset to 0.

- **Bit 6:0 – Resx6:0 - Reserved**

9.12.26 PLL_CF – Transceiver Center Frequency Calibration Control Register

Bit	7	6	
NA ($15A)	PLL_CF_START	Resx6	PLL_CF
Read/Write	RW	RW	
Initial Value	0	1	
Bit	5	4	
NA ($15A)	Resx5	Resx4	PLL_CF
Read/Write	RW	RW	
Initial Value	0	1	
Bit	3	2	
NA ($15A)	Resx3	Resx2	PLL_CF
Read/Write	RW	RW	
Initial Value	0	1	
Bit	1	0	
NA ($15A)	Resx1	Resx0	PLL_CF
Read/Write	RW	RW	
Initial Value	1	1	

This register controls the operation of the center frequency calibration loop.

- **Bit 7 – PLL_CF_START - Start Center Frequency Calibration**

PLL_CF_START = 1 initiates the center frequency calibration. The calibration cycle has finished after 35 µs (typical). The register bit is cleared immediately after finishing the calibration.

- **Bit 6:0 – Resx6:0 - Reserved**

9.12.27 PLL_DCU – Transceiver Delay Cell Calibration Control Register

Bit	7	6	5	4	
NA ($15B)	PLL_DCU_START	Resx6	Resx5	Resx4	PLL_DCU
Read/Write	RW	R	RW	RW	
Initial Value	0	0	1	0	

Bit	3	2	1	0	
NA ($15B)	Resx3	Resx2	Resx1	Resx0	PLL_DCU
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

This register controls the operation of the calibration loop of the delay cell.

- **Bit 7 – PLL_DCU_START - Start Delay Cell Calibration**

PLL_DCU_START = 1 initiates the delay cell calibration. The calibration cycle has finished after at most 6 µs. The register bit is cleared immediately after finishing the calibration.

- **Bit 6:0 – Resx6:0 - Reserved**

9.12.28 PART_NUM – Device Identification Register (Part Number)

Bit	7	6	5	4	3	2	1	0	
NA ($15C)	PART_NUM7:0								PART_NUM
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	1	0	0	0	0	0	1	1	

This register contains the part number of the device.

- **Bit 7:0 – PART_NUM7:0 - Part Number**

These bits decode the part number of the device according to the following table.

Table 9-62 PART_NUM Register Bits

Register Bits	Value	Description
PART_NUM7:0	0x83	ATmega128RFA1 part number

9.12.29 VERSION_NUM – Device Identification Register (Version Number)

Bit	7	6	5	4	3	2	1	0	
NA ($15D)	VERSION_NUM7:0								VERSION_NUM
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the version number of the device. The device identification overwrites the Reset value.

- **Bit 7:0 – VERSION_NUM7:0 - Version Number**

These bits decode the version number of the device according to the following table.

Table 9-63 VERSION_NUM Register Bits

Register Bits	Value	Description
VERSION_NUM7:0	2	Revision AB
	3	Revision C
	4	Revision D

9.12.30 MAN_ID_0 – Device Identification Register (Manufacture ID Low Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($15E)	MAN_ID_07:00								MAN_ID_0
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	1	1	1	1	1	

Bits [7:0] of the 32-bit JEDEC manufacturer ID are stored in this register. Bits [15:8] are stored in register MAN_ID_1. The highest 16 bits of the JEDEC ID are not stored in registers.

- **Bit 7:0 – MAN_ID_07:00 - Manufacturer ID (Low Byte)**

These bits contain bits [7:0] of the 32-bit JEDEC manufacturer ID.

Table 9-64 MAN_ID_0 Register Bits

Register Bits	Value	Description
MAN_ID_07:00	0x1f	Atmel JEDEC manufacturer ID, bits [7:0] of 32 bit manufacturer ID: 00 00 00 1F

9.12.31 MAN_ID_1 – Device Identification Register (Manufacture ID High Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($15F)	MAN_ID_17:10								MAN_ID_1
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

Bits [15:8] of the 32-bit JEDEC manufacturer ID are stored in this register. Bits [7:0] are stored in register MAN_ID_0. The highest 16 bits of the JEDEC ID are not stored in registers.

- **Bit 7:0 – MAN_ID_17:10 - Manufacturer ID (High Byte)**

These bits contain bits [15:8] of the 32-bit JEDEC manufacturer ID.

Table 9-65 MAN_ID_1 Register Bits

Register Bits	Value	Description
MAN_ID_17:10	0x00	Atmel JEDEC manufacturer ID, bits [15:8] of 32 bit manufacturer ID: 00 00 00 1F

9.12.32 SHORT_ADDR_0 – Transceiver MAC Short Address Register (Low Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($160)	SHORT_ADDR_07:00								SHORT_ADDR_0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	1	1	1	1	1	1	1	1	

This register contains the lower 8 bits of the MAC short address for Frame Filter address recognition.

- **Bit 7:0 – SHORT_ADDR_07:00 - MAC Short Address**

These bits contain the bits [7:0] of the MAC short address.

9.12.33 SHORT_ADDR_1 – Transceiver MAC Short Address Register (High Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($161)	SHORT_ADDR_17:10								SHORT_ADDR_1
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	1	1	1	1	1	1	1	1	

This register contains the upper 8 bits of the MAC short address for Frame Filter address recognition.

- **Bit 7:0 – SHORT_ADDR_17:10 - MAC Short Address**

These bits contain the bits [15:8] of the MAC short address.

9.12.34 PAN_ID_0 – Transceiver Personal Area Network ID Register (Low Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($162)	PAN_ID_07:00								PAN_ID_0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	1	1	1	1	1	1	1	1	

This register contains the lower 8 bits of the MAC PAN ID for Frame Filter address recognition.

- **Bit 7:0 – PAN_ID_07:00 - MAC Personal Area Network ID**

These bits contain the bits [7:0] of the MAC PAN ID.

9.12.35 PAN_ID_1 – Transceiver Personal Area Network ID Register (High Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($163)	PAN_ID_17:10								PAN_ID_1
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	1	1	1	1	1	1	1	1	

This register contains the upper 8 bits of the MAC PAN ID for Frame Filter address recognition.

- **Bit 7:0 – PAN_ID_17:10 - MAC Personal Area Network ID**

These bits contain the bits [15:8] of the MAC PAN ID.

9.12.36 IEEE_ADDR_0 – Transceiver MAC IEEE Address Register 0

Bit	7	6	5	4	3	2	1	0	
NA ($164)	IEEE_ADDR_07:00								IEEE_ADDR_0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [7:0] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_07:00 - MAC IEEE Address**

These bits map to the bits [7:0] of the 64 bit MAC IEEE address.

9.12.37 IEEE_ADDR_1 – Transceiver MAC IEEE Address Register 1

Bit	7	6	5	4	3	2	1	0	
NA ($165)	IEEE_ADDR_17:10								IEEE_ADDR_1
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [15:8] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_17:10 - MAC IEEE Address**

These bits map to the bits [15:8] of the 64 bit MAC IEEE address.

9.12.38 IEEE_ADDR_2 – Transceiver MAC IEEE Address Register 2

Bit	7	6	5	4	3	2	1	0	
NA ($166)	IEEE_ADDR_27:20								IEEE_ADDR_2
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [23:16] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_27:20 - MAC IEEE Address**

These bits map to the bits [23:16] of the 64 bit MAC IEEE address.

9.12.39 IEEE_ADDR_3 – Transceiver MAC IEEE Address Register 3

Bit	7	6	5	4	3	2	1	0	
NA ($167)	IEEE_ADDR_37:30								IEEE_ADDR_3
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [31:24] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_37:30 - MAC IEEE Address**

These bits map to the bits [31:24] of the 64 bit MAC IEEE address.

9.12.40 IEEE_ADDR_4 – Transceiver MAC IEEE Address Register 4

Bit	7	6	5	4	3	2	1	0	
NA ($168)	IEEE_ADDR_47:40								IEEE_ADDR_4
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [39:32] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_47:40 - MAC IEEE Address**

These bits map to the bits [39:32] of the 64 bit MAC IEEE address.

9.12.41 IEEE_ADDR_5 – Transceiver MAC IEEE Address Register 5

Bit	7	6	5	4	3	2	1	0	
NA ($169)	IEEE_ADDR_57:50								IEEE_ADDR_5
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [47:40] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_57:50 - MAC IEEE Address**

These bits map to the bits [47:40] of the 64 bit MAC IEEE address.

9.12.42 IEEE_ADDR_6 – Transceiver MAC IEEE Address Register 6

Bit	7	6	5	4	3	2	1	0	
NA ($16A)	IEEE_ADDR_67:60								IEEE_ADDR_6
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [55:48] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_67:60 - MAC IEEE Address**

These bits map to the bits [55:48] of the 64 bit MAC IEEE address.

9.12.43 IEEE_ADDR_7 – Transceiver MAC IEEE Address Register 7

Bit	7	6	5	4	3	2	1	0	
NA ($16B)	IEEE_ADDR_77:70								IEEE_ADDR_7
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the bits [63:56] of the MAC IEEE address for Frame Filter address recognition.

- **Bit 7:0 – IEEE_ADDR_77:70 - MAC IEEE Address**

These bits map to the bits [63:56] of the 64 bit MAC IEEE address.

9.12.44 XAH_CTRL_0 – Transceiver Extended Operating Mode Control Register

Bit	7	6	
NA ($16C)	MAX_FRAME_RETRIES3	MAX_FRAME_RETRIES2	XAH_CTRL_0
Read/Write	RW	RW	
Initial Value	0	0	
Bit	5	4	
NA ($16C)	MAX_FRAME_RETRIES1	MAX_FRAME_RETRIES0	XAH_CTRL_0
Read/Write	RW	RW	
Initial Value	1	1	
Bit	3	2	
NA ($16C)	MAX_CSMA_RETRIES2	MAX_CSMA_RETRIES1	XAH_CTRL_0
Read/Write	RW	RW	
Initial Value	1	0	
Bit	1	0	
NA ($16C)	MAX_CSMA_RETRIES0	SLOTTED_OPERATION	XAH_CTRL_0
Read/Write	RW	RW	
Initial Value	0	0	

This register is used to control various settings of the Extended Operating Mode.

- **Bit 7:4 – MAX_FRAME_RETRIES3:0 - Maximum Number of Frame Retransmission Attempts**

The setting of MAX_FRAME_RETRIES in TX_ARET mode specifies the number of attempts to retransmit a frame when it was not acknowledged by the recipient. The transaction gets canceled if the number of attempts exceeds MAX_FRAME_RETRIES.

Table 9-66 MAX_FRAME_RETRIES Register Bits

Register Bits	Value	Description
MAX_FRAME_RETRIES3:0	0x0	Retransmission of frame is not attempted.
	0x1	Retransmission of frame is attempted once.
	0x2	...
	0xF	Retransmission of frame is attempted 15 times.

- **Bit 3:1 – MAX_CSMA_RETRIES2:0 - Maximum Number of CSMA-CA Procedure Repetition Attempts**

MAX_CSMA_RETRIES specifies the number of retries in TX_ARET mode to repeat the CSMA-CA procedure before the transaction gets canceled. According to IEEE 802.15.4 the valid range of MAX_CSMA_RETRIES is 0 to 5. A value of MAX_CSMA_RETRIES = 7 initiates an immediate frame transmission without performing CSMA-CA. This may especially be required for slotted acknowledgment operation. MAX_CSMA_RETRIES = 6 is reserved.

Table 9-67 MAX_CSMA_RETRIES Register Bits

Register Bits	Value	Description
MAX_CSMA_RETRIES2:0	0x0	No repetition of CSMA-CA procedure
	0x1	One repetition of CSMA-CA procedure
	0x2	...
	0x5	Five repetitions (highest IEEE 802.15.4 compliant value)
	0x6	Reserved
	0x7	Immediate frame re-transmission without performing CSMA-CA

- **Bit 0 – SLOTTED_OPERATION - Set Slotted Acknowledgment**

When using RX_AACK mode in networks operating in beacon or slotted mode according to IEEE 802.15.4-2006, chapter 5.5.1 the register bit SLOTTED_OPERATION indicates that acknowledgment frames are to be sent on back-off slot boundaries (slotted acknowledgment). If this register bit is set the acknowledgment frame transmission has to be initiated by the application software using bit SLPTR of register TRXPR. This waiting state is signaled in sub register TRAC_STATUS of register TRX_STATE with value SUCCESS_WAIT_FOR_ACK.

Table 9-68 SLOTTED_OPERATION Register Bits

Register Bits	Value	Description
SLOTTED_OPERATION	0	The radio transceiver operates in unslotted mode. An acknowledgment frame is automatically sent if requested.
	1	The transmission of an acknowledgment frame has to be controlled by the microcontroller.

9.12.45 CSMA_SEED_0 – Transceiver CSMA-CA Random Number Generator Seed Register

Bit	7	6	5	4	3	2	1	0	
NA ($16D)	CSMA_SEED_07:00								CSMA_SEED_0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	1	1	1	0	1	0	1	0	

This register contains the lower 8 bits of the CSMA_SEED. The upper 3 bits are part of register CSMA_SEED_1. CSMA_SEED is the seed for the random number generation that determines the length of the back-off period in the CSMA-CA algorithm. It is recommended to initialize registers CSMA_SEED by random values. This can be done using the bits RND_VALUE of register PHY_RSSI.

- **Bit 7:0 – CSMA_SEED_07:00 - Seed Value for CSMA Random Number Generator**

These bits contain the bits [7:0] of the CSMA_SEED.

9.12.46 CSMA_SEED_1 – Transceiver Acknowledgment Frame Control Register 2

Bit	7	6	
NA ($16E)	AACK_FVN_MODE1	AACK_FVN_MODE0	CSMA_SEED_1
Read/Write	RW	RW	
Initial Value	0	1	
Bit	5	4	
NA ($16E)	AACK_SET_PD	AACK_DIS_ACK	CSMA_SEED_1
Read/Write	RW	RW	
Initial Value	0	0	
Bit	3	2	
NA ($16E)	AACK_I_AM_COORD	CSMA_SEED_12	CSMA_SEED_1
Read/Write	RW	RW	
Initial Value	0	0	
Bit	1	0	
NA ($16E)	CSMA_SEED_11	CSMA_SEED_10	CSMA_SEED_1
Read/Write	RW	RW	
Initial Value	1	0	

This register is a control register for RX_AACK and contains a part of the CSMA_SEED for the CSMA-CA algorithm.

- **Bit 7:6 – AACK_FVN_MODE1:0 - Acknowledgment Frame Filter Mode**

The frame control field of the MAC header (MHR) contains a frame version subfield. The setting of AACK_FVN_MODE specifies the frame filtering behavior of the radio transceiver. According to the content of these register bits the radio transceiver passes frames with a specific frame version number, number group or independent of the frame version number. Thus the register bits AACK_FVN_MODE define the maximum acceptable frame version. Received frames with a higher frame version number than configured do not pass the address filter and are not acknowledged.

Table 9-69 AACK_FVN_MODE Register Bits

Register Bits	Value	Description
AACK_FVN_MODE1:0	0	Acknowledge frames with version number 0
	1	Acknowledge frames with version number 0 or 1
	2	Acknowledge frames with version number 0 or 1 or 2
	3	Acknowledge frames independent of frame version number

- **Bit 5 – AACK_SET_PD - Set Frame Pending Sub-field**

The content of AACK_SET_PD bit is copied into the frame pending subfield of the acknowledgment frame if the acknowledgment is the answer to a data request MAC command frame. If in addition the bits AACK_FVN_MODE of this register are configured to accept frames with a frame version other than 0 or 1, the content of register bit AACK_SET_PD is also copied into the frame pending subfield of the acknowledgment frame for any MAC command frame with a frame version of 2 or 3 that have the security enabled subfield set to 1. This is done in the assumption that a future version of the IEEE 802.15.4 standard might change the length or structure of the auxiliary security header, so that it is not possible to safely detect whether the MAC command frame is actually a data request command or not.

- **Bit 4 – AACK_DIS_ACK - Disable Acknowledgment Frame Transmission**

If this bit is set no acknowledgment frames are transmitted in RX_AACK Extended Operating Mode even if requested.

- **Bit 3 – AACK_I_AM_COORD - Set Personal Area Network Coordinator**

This register bit has to be set if the node is a PAN coordinator. It is used for address filtering in RX_AACK.

- **Bit 2:0 – CSMA_SEED_12:10 - Seed Value for CSMA Random Number Generator**

These bits contain the bits [10:8] of the CSMA_SEED. The lower part is defined in register CSMA_SEED_0. See register CSMA_SEED_0 for details.

9.12.47 CSMA_BE – Transceiver CSMA-CA Back-off Exponent Control Register

Bit	7	6	5	4	
NA ($16F)	MAX_BE3	MAX_BE2	MAX_BE1	MAX_BE0	CSMA_BE
Read/Write	RW	RW	RW	RW	
Initial Value	0	1	0	1	
Bit	3	2	1	0	
NA ($16F)	MIN_BE3	MIN_BE2	MIN_BE1	MIN_BE0	CSMA_BE
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	1	1	

This register controls the back-off exponent for the CSMA-CA procedure.

- **Bit 7:4 – MAX_BE3:0 - Maximum Back-off Exponent**

These register bits define the maximum back-off exponent used in the CSMA-CA algorithm to generate a pseudo random number for back off the CCA. For details refer to IEEE 802.15.4-2006, section 7.5.1.4. Valid values are 3 to 8.

Table 9-70 MAX_BE Register Bits

Register Bits	Value	Description
MAX_BE3:0	1	This value is not valid for the maximum back-off exponent.
	2	This value is not valid for the maximum back-off exponent.
	3	Minimum, IEEE compliant value for the maximum back-off exponent.
	4	...
	8	Maximum, IEEE compliant value for the maximum back-off exponent.

- **Bit 3:0 – MIN_BE3:0 - Minimum Back-off Exponent**

These register bits define the minimum back-off exponent used in the CSMA-CA algorithm to generate a pseudo random number for back off the CCA. For details refer to IEEE 802.15.4-2006, section 7.5.1.4. Valid values are MAX_BE, MAX_BE-1), ..., 0. If MIN_BE = 0 and MAX_BE = 0 the CCA back off period is always set to 0.

Table 9-71 MIN_BE Register Bits

Register Bits	Value	Description
MIN_BE3:0	0	Minimum value of minimum back-off exponent.
	1	...
	8	Maximum value of minimum back-off exponent. MIN_BE must be smaller or equal to MAX_BE.

9.12.48 TST_CTRL_DIGI – Transceiver Digital Test Control Register

Bit	7	6	
NA ($176)	Resx7	Resx6	TST_CTRL_DIGI
Read/Write	RW	RW	
Initial Value	0	0	
Bit	5	4	
NA ($176)	Resx5	Resx4	TST_CTRL_DIGI
Read/Write	RW	RW	
Initial Value	0	0	
Bit	3	2	
NA ($176)	TST_CTRL_DIG3	TST_CTRL_DIG2	TST_CTRL_DIGI
Read/Write	RW	RW	
Initial Value	0	0	
Bit	1	0	
NA ($176)	TST_CTRL_DIG1	TST_CTRL_DIG0	TST_CTRL_DIGI
Read/Write	RW	RW	
Initial Value	0	0	

This register takes part in the activation sequence of the continuous transmission test mode. Other functionality of this register is reserved for internal use.

- **Bit 7:4 – Resx7:4 - Reserved**
- **Bit 3:0 – TST_CTRL_DIG3:0 - Digital Test Controller Register**

This sub-register selects a test controller function. All values not listed int the following table are reserved for internal use.

Table 9-72 TST_CTRL_DIG Register Bits

Register Bits	Value	Description
TST_CTRL_DIG3:0	0	NORMAL (no test is active)
	15	TST_CONT_TX (continuous transmit)

9.12.49 TST_RX_LENGTH – Transceiver Received Frame Length Register

Bit	7	6	5	4	
NA ($17B)	RX_LENGTH7	RX_LENGTH6	RX_LENGTH5	RX_LENGTH4	TST_RX_LENGTH
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

Bit	3	2	1	0	
NA ($17B)	RX_LENGTH3	RX_LENGTH2	RX_LENGTH1	RX_LENGTH0	TST_RX_LENGTH
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

This register contains the frame length information of a received frame. This information is not stored in the frame buffer. The frame length information is written to this register after the last received octet.

- **Bit 7:0 – RX_LENGTH7:0 - Received Frame Length**

These bits contain the length of the last received frame.

9.12.50 TRXFBST – Start of frame buffer

Bit	7	6	5	4	3	2	1	0	
NA ($180)	TRXFBST7:0								TRXFBST
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

First byte of the 128 byte long frame buffer of the TRX24.

- **Bit 7:0 – TRXFBST7:0 - Frame Buffer Start Byte**

9.12.51 TRXFBEND – End of frame buffer

Bit	7	6	5	4	3	2	1	0	
NA ($1FF)	TRXFBEND7:0								TRXFBEND
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register is the last byte of the 128 byte long frame buffer of the radio transceiver.

- **Bit 7:0 – TRXFBEND7:0 - Frame Buffer End Byte**

10 MAC Symbol Counter

10.1 Main Features

The MAC symbol counter provides symbol timing information for IEEE 802.15.4 wireless networks. The counter time base can be derived from the 16 MHz crystal or the RTC (32.768 kHz crystal on TOSC) during operation. In deep-sleep mode the counter operates from the RTC clock. The module provides the following features:

- **Backoff slot counter with interrupt generation**
- **Counter clock source selection between XTAL1 (16 MHz) and TOSC1 (RTC)**
- **Automatic RTC clock selection for sleep mode operation and automatic fallback**
- **3 independent compare units with relative and absolute compare mode and interrupt generation (support for slotted operation and superframe handling)**
- **Low-power, deep-sleep mode operation and system wake up with all symbol counter interrupt events**
- **Automatic SFD and incoming beacon timestamping**
- **Manual beacon timestamping**
- **Manual timer synchronization within a 16 µs symbol period by resetting clock prescaler and backoff slot counter**
- **Atomic read/write access for 32 bit registers**

10.2 Clock source selection and Sleep/Active mode operation

The symbol counter can be sourced by the transceiver clock or by the asynchronous Real Time Clock (RTC) oscillator. If the transceiver goes from active mode into sleep mode, the symbol counter clock source is switched to the RTC clock automatically. A clock source change is indicated in the bit SCCKSEL of Register "SCCR0 – Symbol Counter Control Register 0" on page 143 . The bit SCCKSEL can not be written if the radio tranceiver is in SLEEP mode.

After wake up, the counter switches back to the clock source which was selected before going to sleep mode. Switching the clock source from RTC to 16 MHz resets the 16 MHz clock prescaler. This makes sure, that after switching back the clock source, the symbol counter starts counting with a full 16 µs symbol period.

The clock source can be selected with bit SCCKSEL in the SCCR0 Register

10.3 32 bit Register Access (Atomic Read/Write)

All 32 bit registers support atomic read or write operation. That means reading or writing the least significant *xxx*LL byte (the register name ends in LL) updates or captures the complete 32 bit value.

Read Access:	1. Reading the LL-Byte captures the 32 bit value in a temporary register
	2. Read the upper 3 bytes
Write Access:	1. Write the upper 3 byte
	2. Writing the LL-Byte stores the 32 bit value in the counter registers

The same temporary register is used for all 32 bit register of the MAX symbol counter.

10.4 Symbol Counter (32 bit, SCCNT)

The symbol counter is a 32 bit counter which can be sourced by a 62.5 kHz clock, derived from the 16 MHz system clock or from the RTC (32.768 kHz). If sourced by the RTC, a special control circuitry ensures that the counter error does not exceed one symbol period.

The symbol counter can be set or read from the controller. Reading must start with the least significant byte. If the least significant byte is accessed, all 32 bit of the counter are captured. A read access to SCCNTLL requires a maximum of three AVR clocks. Reading the upper three bytes of the counter requires two CPU clock cycles for each byte.

Writing to the counter should start with the most significant byte. Writing the least significant byte initiates the counter update and the new 32 bit counter value is loaded into the counter with the next available counter clock edge. This can take up to 16 µs beginning from the low byte write operation, if the counter is sourced by the RTC.

If the counter clock is derived from the 16 MHz clock system, the new counter value is stored immediately.

During the counter update cycle, the counter busy flag SCBSY in the SCSR register is set to “1”. As long as this bit is “1”, no further read/write access to the counter should be initiated. The same applies if the AVR is forced to any sleep mode with disabled AVR clock, right after writing to the SCCNT register. If the counter busy flag is not checked before going to sleep, it is possible that the counter register is not updated correctly.

The symbol counter overflow is indicated by a overflow interrupt. The interrupt is generated when the counter turns from 0xFFFFFFFF to 0x00000000.

10.5 Symbol Counter SFD Timestamp Register (32 bit, SCTSR, Read Only)

The SFD timestamp register stores the symbol counter value at the time, the SFD has been detected. The Register value becomes valid if a valid frame length byte (frame length > 0) has been detected, but it is not checked if the received frame is valid (CRC check). Timestamping must be enabled in the control register (Bit SCTSE of Register SCCR0). A read access to SCTSRLL requires a maximum of three AVR clocks. Reading the upper three bytes of the timestamp requires two CPU clock cycles for each byte.

Note that there is no separate interrupt provided for timestamping. Instead the TRX24_RX_START interrupt can be used (see "Interrupt Vectors in ATmega128RFA1" on page 211).

10.6 Symbol Counter Beacon Timestamp Register (32 bit, SCBTSR)

If timestamping is enabled in the SCCR register, the beacon timestamp register is updated with the SFD timestamp at the end of the received frame, if the received frame was a beacon frame with valid FCS and:

- Source PAN identifier == {PAN_ID_1, PAN_ID_0}

or

- {PAN_ID_1, PAN_ID_0} == 0xFFFF

PAN_ID_0 and PAN_ID_1 are register of the radio transceiver, see "PAN_ID_0 – Transceiver Personal Area Network ID Register (Low Byte)" on page 124.

Beacon timestamps can also be generated manually. Writing “1” to SCMBTS of Register SCCR0 captures the current symbol counter value and stores it in the beacon timestamp register. The bit is cleared automatically afterwards.

It is also possible to manually set the register in order to provide a distinct starting value for the relative compare modes (see next section).

10.7 Compare Unit (3x 32 bit, SCOCR1, SCOCR2, SCOCR3)

The compare unit contains 3 independent 32 bit compare modules and is used to compare the current counter value with the value stored in the compare register, and optionally the beacon timestamp register. There are two possible modes available which can be selected separately for all three compare modules:

1. Absolute Compare: In this mode the value stored in the compare register is compared directly with the symbol counter value (SCCNT == SCOCRx). If the values are equal an interrupt is generated.

2. Relative Compare: This mode allows the compare between the current symbol counter value and the compare value plus the beacon timestamp value (SCCNT == SCBTSR + SCOCRx). This mode can be used to generate an interrupt at a time offset relative to the value stored in the beacon timestamp register.

Note that a beacon timestamp is valid after a valid FCS. The relative compare must exceed the beacon length, otherwise no relative compare interrupt will occure.

10.8 Interrupt Control Registers

The interrupt status and mask registers control the interrupt generation. Each interrupt can be enabled in SCIRQM (Symbol Counter IRQ Mask Register). If an interrupt occurs, the appropriate interrupt flag within the interrupt status register is set regardless of the interrupt mask register setting. If the appropriate interrupt is enabled, an interrupt is generated.

The interrupt flags can be cleared either by:

1. Entering the respective interrupt handler, or
2. Writing “one” to the according interrupt flag in the interrupt status register.

All Interrupts can be used to wakeup the controller from any sleep state.

10.9 Backoff Slot Counter

The backoff slot counter can be used to provide accurate MAC protocol timing. The counter is sourced by the transceiver clock and works only if the transceiver clock is running. If the transceiver is disabled or in sleep mode the counter is also disabled.

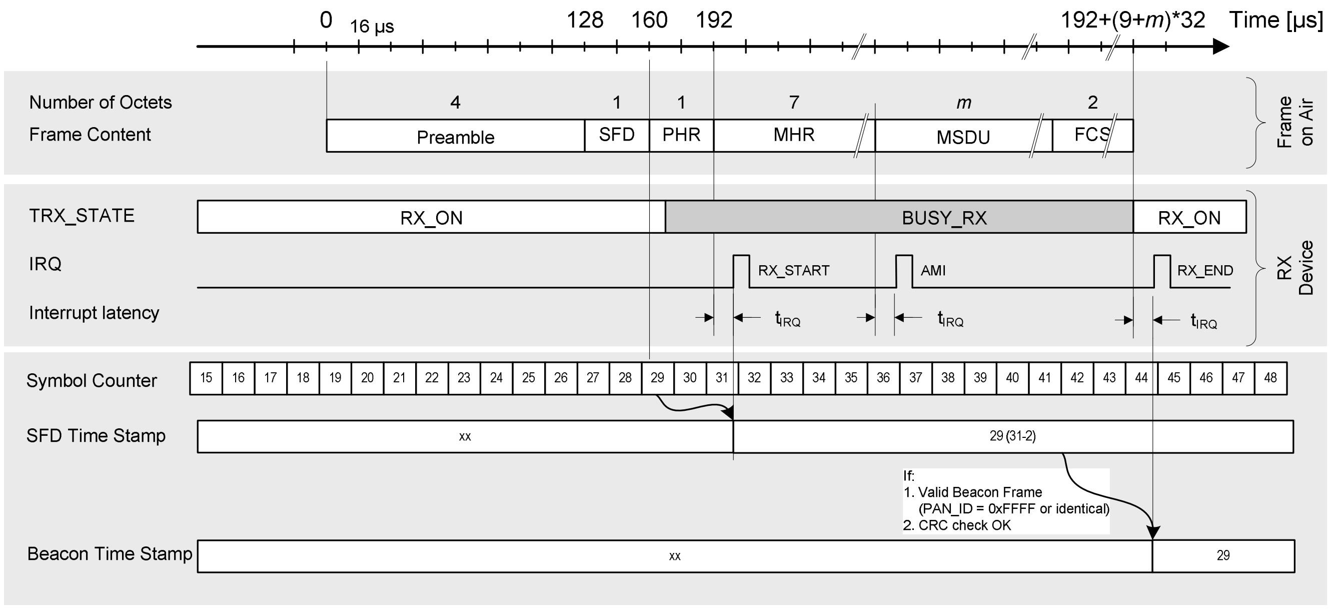
The counter generates periodic Interrupts every 20 symbols, i.e. every 320 µs.

10.10 Symbol Counter Usage

10.10.1 SFD and Beacon Timestamp Generation

The SFD timestamp register is updated with the symbol counter value at the time the SFD value has been received completely. For an incoming frame, the register is valid after the RX_START IRQ was issued until the next RX_START IRQ. SFD timestamps are generated for all incoming frames with valid SFD and length field even if the PSDU is corrupted (invalid FCS).

Figure 10-1. SFD and Beacon Timestamp Generation



Note that Figure 10-1 contains no exact timing information; it is for visualization only.

The beacon timestamp register is updated with the SFD timestamp value at the end of the frame (RX_END IRQ), if the received frame was a beacon frame with valid FCS and expected source PAN identifier or { PAN_ID_1, PAN_ID_0} = 0xFFFF.

The register value is valid until a new beacon frame has been received or the beacon timestamp is updated manually. A manual beacon timestamp can be generated by writing “1” to SCMBTS of the SCCR0 register.

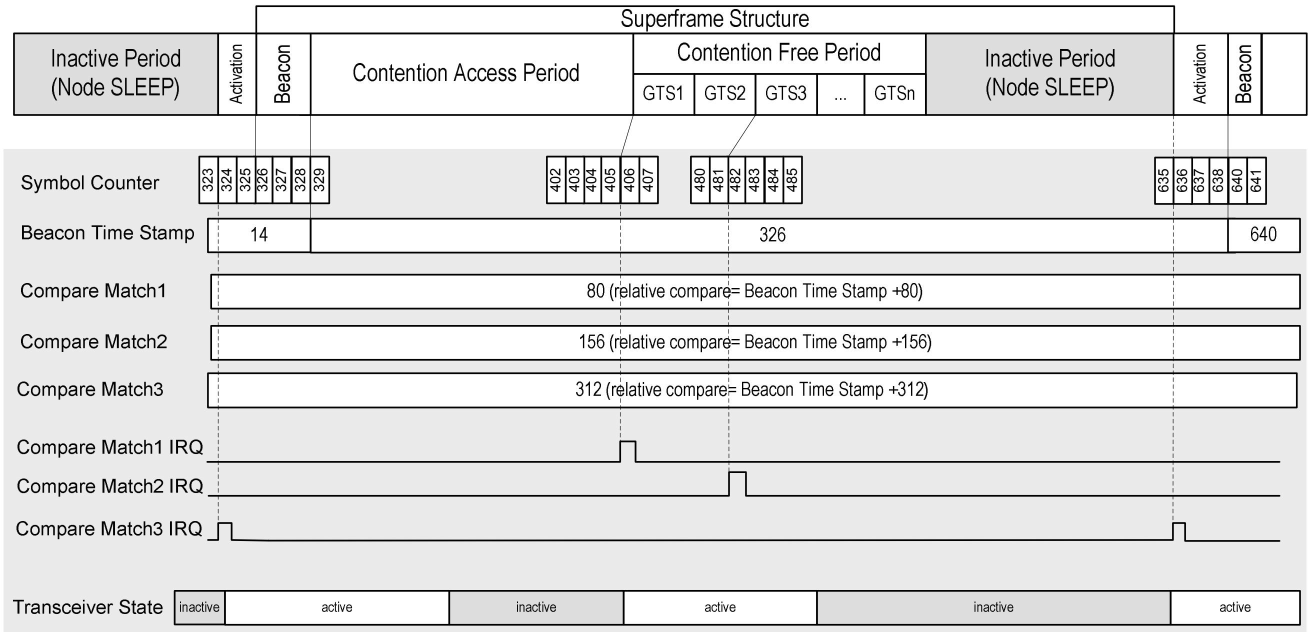
10.10.2 Relative Compare Mode for Superframe Access Timing

The IEEE 802.15.4 describes a superframe structure which contains different time slots where a device can access the channel.

The Symbol Counter together with the three compare units provide support for waking up the device at the right time to receive the beacon for superframe synchronization and at certain times within the superframe.

A typical superframe timing scenario using the symbol counter relative compare mode is shown in Figure 10-2 on page 137. The Symbol Counter values in the figure do not reflect realistic time intervals but demonstrate the principle of operation.

Figure 10-2. Relative Compare Mode



The compare match registers are programmed with symbol intervals relative to the beacon frame SFD timestamp. For instance the SCCMP1 is programmed to 80, because the first Granted Time Slot (GTS1) is expected 80 symbols after the beacon frame. Register SCCMP2 is programmed to 156 to meet GTS3 156 symbols after the beacon frame. SCCMP3 is programmed to 312. This is the time interval where the beacon of the next superframe is expected. Because it requires some time to activate the transceiver and there is also some timing drift possible, the compare interrupt must be programmed to wake up some symbols in advance to make sure the next beacon is not missed.

If the controller receives a compare match wake up event it is activating the transceiver. After the frame operations are finished, the system can go back to sleep until the next compare match event occurs.

10.11 Register Description

10.11.1 SCCNTHH – Symbol Counter Register HH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E4)	SCCNTHH7:0								SCCNTHH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the most significant byte of the 32 bit Symbol Counter.

- **Bit 7:0 – SCCNTHH7:0 - Symbol Counter Register HH-Byte**

10.11.2 SCCNTHL – Symbol Counter Register HL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E3)	SCCNTHL7:0								SCCNTHL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second most significant byte of the 32 bit Symbol Counter.

- **Bit 7:0 – SCCNTHL7:0 - Symbol Counter Register HL-Byte**

10.11.3 SCCNTLH – Symbol Counter Register LH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E2)	SCCNTLH7:0								SCCNTLH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second least significant byte of the 32 bit Symbol Counter.

- **Bit 7:0 – SCCNTLH7:0 - Symbol Counter Register LH-Byte**

10.11.4 SCCNTLL – Symbol Counter Register LL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E1)	SCCNTLL7:0								SCCNTLL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the least significant byte of the 32 bit Symbol Counter.

- **Bit 7:0 – SCCNTLL7:0 - Symbol Counter Register LL-Byte**

10.11.5 SCTSRHH – Symbol Counter Frame Timestamp Register HH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($EC)	SCTSRHH7:0								SCTSRHH
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the most significant byte of the 32 bit frame (SFD) timestamp register

- **Bit 7:0 – SCTSRHH7:0 - Symbol Counter Frame Timestamp Register HH-Byte**

10.11.6 SCTSRHL – Symbol Counter Frame Timestamp Register HL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($EB)	SCTSRHL7:0								SCTSRHL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second most significant byte of the 32 bit Frame (SFD) Timestamp Register

- **Bit 7:0 – SCTSRHL7:0 - Symbol Counter Frame Timestamp Register HL-Byte**

10.11.7 SCTSRLH – Symbol Counter Frame Timestamp Register LH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($EA)	SCTSRLH7:0								SCTSRLH
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second least significant byte of the 32 bit Frame (SFD) Timestamp Register

- **Bit 7:0 – SCTSRLH7:0 - Symbol Counter Frame Timestamp Register LH-Byte**

10.11.8 SCTSRLL – Symbol Counter Frame Timestamp Register LL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E9)	SCTSRLL7:0								SCTSRLL
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the least significant byte of the 32 bit Frame (SFD) Timestamp Register

- **Bit 7:0 – SCTSRLL7:0 - Symbol Counter Frame Timestamp Register LL-Byte**

10.11.9 SCBTSRHH – Symbol Counter Beacon Timestamp Register HH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E8)	SCBTSRHH7:0								SCBTSRHH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the most significant byte of the 32 bit Beacon Timestamp Register. The Beacon Timestamp Register is updated with the contents of the Frame Timestamp Register if the received frame was a valid beacon frame with matching source PAN identifier or register {PAN_ID_1, PAN_ID_0} = 0xFFFF.

- **Bit 7:0 – SCBTSRHH7:0 - Symbol Counter Beacon Timestamp Register HH-Byte**

10.11.10 SCBTSRHL – Symbol Counter Beacon Timestamp Register HL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E7)	SCBTSRHL7:0								SCBTSRHL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second most significant byte of the 32 bit Beacon Timestamp Register.

- **Bit 7:0 – SCBTSRHL7:0 - Symbol Counter Beacon Timestamp Register HL-Byte**

10.11.11 SCBTSRLH – Symbol Counter Beacon Timestamp Register LH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E6)	SCBTSRLH7:0								SCBTSRLH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second least significant byte of the 32 bit Beacon Timestamp Register.

- **Bit 7:0 – SCBTSRLH7:0 - Symbol Counter Beacon Timestamp Register LH-Byte**

10.11.12 SCBTSRLL – Symbol Counter Beacon Timestamp Register LL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($E5)	SCBTSRLL7:0								SCBTSRLL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the least significant byte of the 32 bit Beacon Timestamp Register.

- **Bit 7:0 – SCBTSRLL7:0 - Symbol Counter Beacon Timestamp Register LL-Byte**

10.11.13 SCOCR1HH – Symbol Counter Output Compare Register 1 HH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F8)	SCOCR1HH7:0								SCOCR1HH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the most significant byte of the 32 bit compare value for the first compare unit

- **Bit 7:0 – SCOCR1HH7:0 - Symbol Counter Output Compare Register 1 HH-Byte**

10.11.14 SCOCR1HL – Symbol Counter Output Compare Register 1 HL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F7)	SCOCR1HL7:0								SCOCR1HL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second most significant byte of the 32 bit compare value for the first compare unit

- **Bit 7:0 – SCOCR1HL7:0 - Symbol Counter Output Compare Register 1 HL-Byte**

10.11.15 SCOCR1LH – Symbol Counter Output Compare Register 1 LH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F6)	SCOCR1LH7:0								SCOCR1LH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second least significant byte of the 32 bit compare value for the first compare unit

- **Bit 7:0 – SCOCR1LH7:0 - Symbol Counter Output Compare Register 1 LH-Byte**

10.11.16 SCOCR1LL – Symbol Counter Output Compare Register 1 LL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F5)	SCOCR1LL7:0								SCOCR1LL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the least significant byte of the 32 bit compare value for the first compare unit

- **Bit 7:0 – SCOCR1LL7:0 - Symbol Counter Output Compare Register 1 LL-Byte**

10.11.17 SCOCR2HH – Symbol Counter Output Compare Register 2 HH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F4)	SCOCR2HH7:0								SCOCR2HH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the most significant byte of the 32 bit compare value for the second compare unit

- **Bit 7:0 – SCOCR2HH7:0 - Symbol Counter Output Compare Register 2 HH-Byte**

10.11.18 SCOCR2HL – Symbol Counter Output Compare Register 2 HL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F3)	SCOCR2HL7:0								SCOCR2HL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second most significant byte of the 32 bit compare value for the second compare unit

- **Bit 7:0 – SCOCR2HL7:0 - Symbol Counter Output Compare Register 2 HL-Byte**

10.11.19 SCOCR2LH – Symbol Counter Output Compare Register 2 LH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F2)	SCOCR2LH7:0								SCOCR2LH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second least significant byte of the 32 bit compare value for the second compare unit

- **Bit 7:0 – SCOCR2LH7:0 - Symbol Counter Output Compare Register 2 LH-Byte**

10.11.20 SCOCR2LL – Symbol Counter Output Compare Register 2 LL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F1)	SCOCR2LL7:0								SCOCR2LL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the least significant byte of the 32 bit compare value for the second compare unit

- **Bit 7:0 – SCOCR2LL7:0 - Symbol Counter Output Compare Register 2 LL-Byte**

10.11.21 SCOCR3HH – Symbol Counter Output Compare Register 3 HH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($F0)	SCOCR3HH7:0								SCOCR3HH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the most significant byte of the 32 bit compare value for the third compare unit

- **Bit 7:0 – SCOCR3HH7:0 - Symbol Counter Output Compare Register 3 HH-Byte**

10.11.22 SCOCR3HL – Symbol Counter Output Compare Register 3 HL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($EF)	SCOCR3HL7:0								SCOCR3HL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second most significant byte of the 32 bit compare value for the third compare unit

- **Bit 7:0 – SCOCR3HL7:0 - Symbol Counter Output Compare Register 3 HL-Byte**

10.11.23 SCOCR3LH – Symbol Counter Output Compare Register 3 LH-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($EE)	SCOCR3LH7:0								SCOCR3LH
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the second least significant byte of the 32 bit compare value for the third compare unit

- **Bit 7:0 – SCOCR3LH7:0 - Symbol Counter Output Compare Register 3 LH-Byte**

10.11.24 SCOCR3LL – Symbol Counter Output Compare Register 3 LL-Byte

Bit	7	6	5	4	3	2	1	0	
NA ($ED)	SCOCR3LL7:0								SCOCR3LL
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register contains the least significant byte of the 32 bit compare value for the third compare unit

- **Bit 7:0 – SCOCR3LL7:0 - Symbol Counter Output Compare Register 3 LL-Byte**

10.11.25 SCCR0 – Symbol Counter Control Register 0

Bit	7	6	5	4	3	2	1	0	
NA ($DC)	SCRES	SCMBTS	SCEN	SCCKSEL	SCTSE	SCCMP3	SCCMP2	SCCMP1	SCCR0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The Control Register 0 is used to setup the operating mode of the symbol counter and the compare units

- **Bit 7 – SCRES - Symbol Counter Synchronization**

If this bit is set to 1, the 16 MHz clock prescaler as well as the backoff slot counter is cleared. This function can be used to align the symbol timing within one 16 µs symbol period and to restart the backoff slot counter with a complete 320 µs period. This feature works only if the symbol counter module operates with the 16 MHz clock from XTAL1. After switching to RTC clock source, the symbol period synchronization is lost. This bit is cleared automatically.

- **Bit 6 – SCMBTS - Manual Beacon Timestamp**

With this bit a manual beacon timestamp can be generated. If set to 1, the current symbol counter value is stored into the beacon timestamp register. The bit is cleared afterwards. The manual beacon timestamping can be used in conjunction with the relative compare mode of the three compare units to generate compare match interrupts without having a beacon frame received.

- **Bit 5 – SCEN - Symbol Counter enable**

This bit activates the symbol counter module. If the bit is not set, the counter, backoff slot counter and the compare unit are disabled and disconnected from the clock. In this way the power consumption can be reduced. All registers can be accessed, but write access to the counter register SCCNT is not possible.

- **Bit 4 – SCCKSEL - Symbol Counter Clock Source select**

With this bit the clock source for the symbol counter can be selected. If the bit is one, the RTC clock from TOSC1 is selected, otherwise the symbol counter operates with the clock from XTAL1. During transceiver sleep modes the clock falls back to the RTC clock source, regardless of the selected clock. After wakeup, it switches back to the previosly selected clock source.

- **Bit 3 – SCTSE - Symbol Counter Automatic Timestamping enable**

This bit enables automatic SFD and Beacon Timestamping. If the bit is zero, no automatic timestamp capturing is possible. Only manual beacon timestamping can be used.

- **Bit 2 – SCCMP3 - Symbol Counter Compare Unit 3 Mode select**

This bit enables the relative compare mode for compare unit 3. If enabled, the counter value is compared against the content of the beacon timestamp register plus the content of the compare register 3 (SCCNT == SCBTS+SCOCR3). Otherwise, the counter is compared against the copare register 3 (SCCNT == SCOCR3).

- **Bit 1 – SCCMP2 - Symbol Counter Compare Unit 2 Mode select**

This bit enables the relative compare mode for compare unit 2. If enabled, the counter value is compared against the content of the beacon timestamp register plus the content of the compare register 2 (SCCNT == SCBTS+SCOCR2). Otherwise, the counter is compared against the copare register 2 (SCCNT == SCOCR2).

- **Bit 0 – SCCMP1 - Symbol Counter Compare Unit 1 Mode select**

This bit enables the relative compare mode for compare unit 1. If enabled, the counter value is compared against the content of the beacon timestamp register plus the content of the compare register 1 (SCCNT == SCBTS+SCOCR1). Otherwise, the counter is compared against the copare register 1 (SCCNT == SCOCR1).

10.11.26 SCCR1 – Symbol Counter Control Register 1

Bit	7	6	5	4	3	2	1	0	
NA ($DD)	Res6	Res5	Res4	Resx4	Resx3	Resx2	Resx1	SCENBO	SCCR1
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

This register is used to enable the backoff slot counter.

- **Bit 7:5 – Res6:4 - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 4:1 – Resx4:1 - Reserved**
- **Bit 0 – SCENBO - Backoff Slot Counter enable**

If this bit is set, the backoff slot counter starts working. To enable the corresponding IRQ the SCIRQM register must be updated.

10.11.27 SCSR – Symbol Counter Status Register

Bit	7	6	5	4	3	2	1	0	
NA ($DE)	Res6	Res5	Res4	Res3	Res2	Res1	Res0	SCBSY	SCSR
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:1 – Res6:0 - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 0 – SCBSY - Symbol Counter busy**

This bit is set if a write operation to the symbol counter register is pending. This bit is set after writing the counter low byte (SCCNTLL) until the symbol counter is updated with the new value. This update process can take up to 16 µs and during this time no read or write access to the 32 bit counter register should occure.

10.11.28 SCIRQS – Symbol Counter Interrupt Status Register

Bit	7	6	5	4	3	2	1	0	
NA ($E0)	Res2	Res1	Res0	IRQSBO	IRQSOF	IRQSCP3	IRQSCP2	IRQSCP1	SCIRQS
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The Interrupt Status Register indicates pending interrupt requests. If the corresponding interrupt mask bit is set, an interrupt service routine is called and the status bit is cleared automatically. It is also possible to clear the status bit by writing "1" to the selected bit.

- **Bit 7:5 – Res2:0 - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 4 – IRQSBO - Backoff Slot Counter IRQ**

This interrupt is generated every 320 µs, that means every 20 symbols.

- **Bit 3 – IRQSOF - Symbol Counter Overflow IRQ**

This interrupt is generated when the 32 bit counter turns from 0xFFFFFFF to 0x00000000.

- **Bit 2 – IRQSCP3 - Compare Unit 3 Compare Match IRQ**

This interrupt indicates a compare match on compare unit 3.

- **Bit 1 – IRQSCP2 - Compare Unit 2 Compare Match IRQ**

This interrupt indicates a compare match on compare unit 2.

- **Bit 0 – IRQSCP1 - Compare Unit 1 Compare Match IRQ**

This interrupt indicates a compare match on compare unit 1.

10.11.29 SCIRQM – Symbol Counter Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
NA ($DF)	Res2	Res1	Res0	IRQMBO	IRQMOF	IRQMCP3	IRQMCP2	IRQMCP1	SCIRQM
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The Interrupt Mask Register is used to enable corresponding interrupts. After reset all interrupts are disabled. Disabled interrupts are still captured in the interrupt status register SCIRQS, but no interrupt is requested. Before enabling an interrupt, the corresponding interrupt status bit should be cleared by writing a 1. If the status bit is set and the IRQ gets enabled, the IRQ handler is called immediatly.

- **Bit 7:5 – Res2:0 - Reserved Bit**

This bit is reserved for future use. The result of a read access is undefined. The register bit must always be written with the reset value.

- **Bit 4 – IRQMBO - Backoff Slot Counter IRQ enable**

This bit enables the SCNT_BACKOFF interrupt.

- **Bit 3 – IRQMOF - Symbol Counter Overflow IRQ enable**

This bit enables the SCNT_OVFL interrupt.

- **Bit 2 – IRQMCP3 - Symbol Counter Compare Match 3 IRQ enable**

This bit enables the SCNT_CMP3 interrupt.

- **Bit 1 – IRQMCP2 - Symbol Counter Compare Match 2 IRQ enable**

This bit enables the SCNT_CMP2 interrupt.

- **Bit 0 – IRQMCP1 - Symbol Counter Compare Match 1 IRQ enable**

This bit enables the SCNT_CMP1 interrupt.

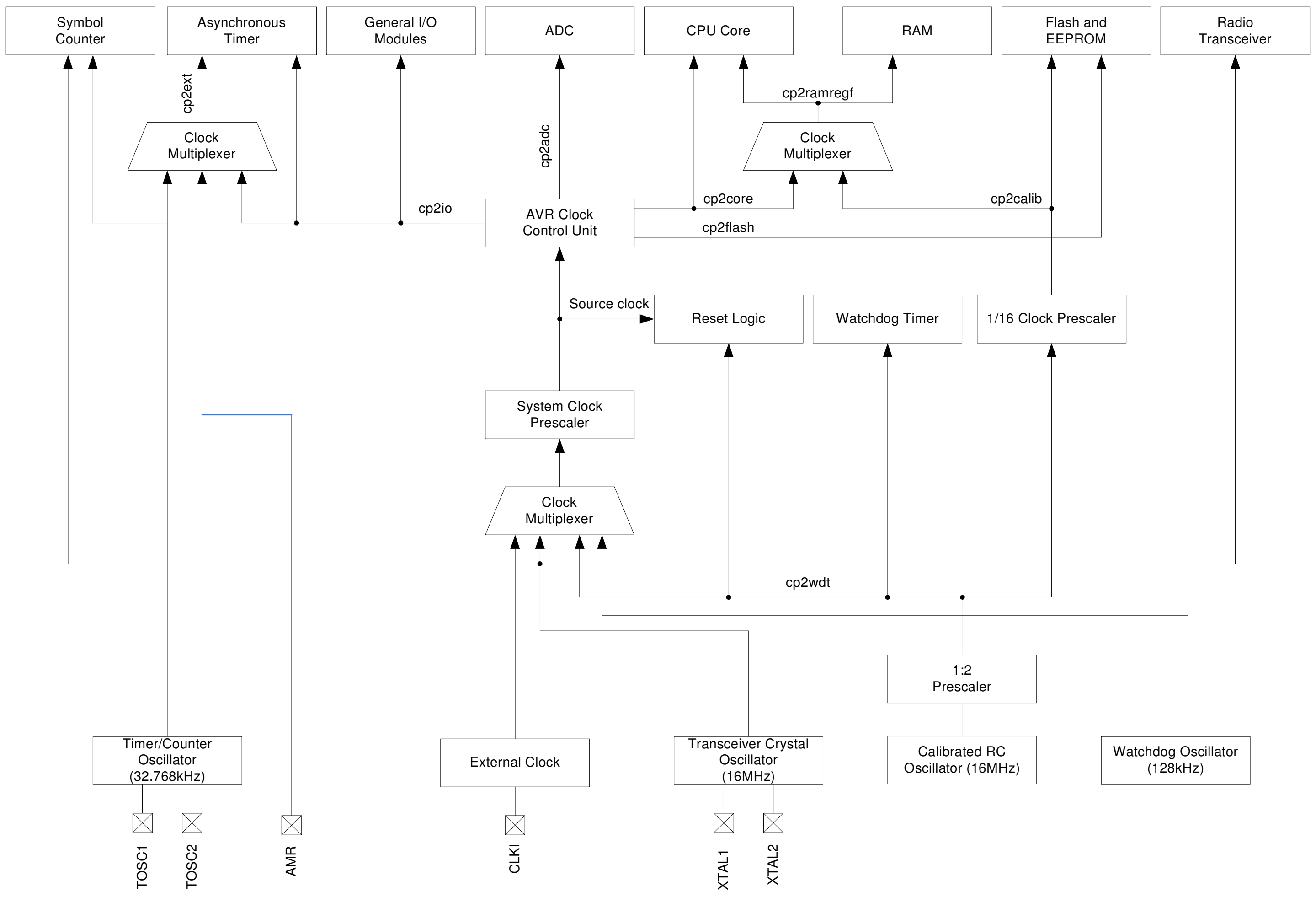
11 System Clock and Clock Options

This section describes the clock options for the AVR microcontroller.

11.1 Overview

Figure 11-1 below 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 chapter "Power Management and Sleep Modes" on page 156. The clock systems are detailed below.

Figure 11-1. Clock Distribution



11.2 Clock Systems and their Distribution

11.2.1 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.

11.2.2 I/O Clock – $clk_{I/O}$

The I/O clock is used by the majority of the I/O modules, like Timer/Counters, SPI, and USART. 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. Also note that start condition detection in the 2-wire serial interface (TWI) module is carried out asynchronously when $clk_{I/O}$ is halted. Similar the TWI address recognition in all sleep modes also occurs asynchronously.

11.2.3 Flash Clock – clk_{FLASH}

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

11.2.4 Asynchronous Timer Clock – clk_{ASY}

The Asynchronous Timer clock allows the Asynchronous Timer/Counter to be clocked directly from an external clock or an external 32 kHz clock crystal. The dedicated clock domain allows using this Timer/Counter as a real-time counter even if the device is in sleep mode.

11.2.5 ADC Clock – clkADC

The ADC is provided with a dedicated clock domain. This allows halting the CPU and I/O clocks in order to reduce noise generated by digital circuitry. This gives more accurate ADC conversion results.

11.3 Clock Sources

The device has the following clock source options, selectable by Flash Fuse bits as shown below. The clock from the selected source is input to the AVR clock generator, and routed to the appropriate modules.

Table 11-1. Device Clocking Options Select[1]

Device Clocking Option	CKSEL3:0
Transceiver clock	1111 – 0110
Reserved	0101 - 0100
Internal 128 kHz RC Oscillator	0011
Calibrated Internal RC Oscillator	0010
External Clock	0000
Reserved	0001

Notes: 1. For all fuses “1” means unprogrammed while “0” means programmed.

11.3.1 Default Clock Source

The device is shipped with internal RC oscillator at 16.0 MHz, the 1:2 prescaler enabled and with the fuse CKDIV8 programmed, resulting in 1.0 MHz system clock. The startup time is set to maximum time. (CKSEL = "0010", SUT = "10", CKDIV8 = "0"). The default setting ensures that all users can make their desired clock source setting using any available programming interface.

11.3.2 Clock Start-up Sequence

Any clock source needs a minimum number of oscillating cycles before it can be considered stable.

To ensure sufficient startup time, the device issues an internal reset with a time-out delay (t_{TOUT}) after the device reset is released by all other reset sources. Section "Power-on Reset" on page 177 describes the start conditions for the internal reset. The delay (t_{TOUT}) is timed from the Watchdog Oscillator and the number of cycles in the delay is set by the SUTx and CKSELx fuse bits. The selectable delays are shown in Table 11-2 below. The frequency of the Watchdog Oscillator is voltage dependent as shown in section "Typical Characteristics" on page 510.

Table 11-2. Number of Watchdog Oscillator Cycles

Typ Time-out	Number of Cycles
0 ms	0
4.0 ms	512
64 ms	8K (8,192)

Main purpose of the delay is to keep the AVR in reset until it is supplied with a stable V_{DEVDD}. The delay will not monitor the actual voltage and it will be required to select a delay longer than the DEVDD rise time. If this is not possible, an internal or external Brown-Out Detection (BOD) circuit should be used. A BOD circuit will ensure sufficient V_{DEVDD} before it releases the reset, and the time-out delay can be disabled. Disabling the time-out delay without utilizing a Brown-Out Detection circuit is not recommended.

The oscillator is required to oscillate for a minimum number of cycles before the clock is considered stable. An internal ripple counter monitors the oscillator output clock, and keeps the internal reset active for a given number of clock cycles. The reset is then released and the device will start to execute. The recommended oscillator start-up time is dependent on the clock type, and varies from 6 cycles for an externally applied clock to 32K cycles for a low frequency crystal.

The start-up sequence for the clock includes both the time-out delay and the start-up time when the device starts up from reset. When starting up from Power-save or Power-down mode, DEVDD is assumed to be at a sufficient level and only the start-up time is included.

11.4 Calibrated Internal RC Oscillator

By default, the Internal RC Oscillator provides an approximate 16 MHz clock. The RC oscillator is voltage and temperature dependent, but can be very accurately calibrated by the user. See chapter "Clock Characteristics" on page 502 and "Internal Oscillator Speed" on page 510 for more details. The device is shipped with the CKDIV8 Fuse and the 1:2 system clock prescaler programmed. See section "System Clock Prescaler" on page 152 for more details.

This clock may be selected as the system clock by programming the CKSEL Fuses as shown in Table 11-3 on page 150. If selected, it will operate with no external components. During reset, hardware loads the pre-programmed calibration value into the OSCCAL Register and thereby automatically calibrates the RC Oscillator. The accuracy of this calibration is shown as Factory calibration in section "Clock Characteristics" on page 502.

By changing the OSCCAL register (see "OSCCAL – Oscillator Calibration Value" on page 153) from Software, it is possible to get a higher calibration accuracy than by using the factory calibration. The accuracy of this calibration is shown as User calibration in section "Clock Characteristics" on page 502.

When this Oscillator is used as the chip clock, the Watchdog Oscillator will still be used for the Watchdog Timer and for the Reset Time-out. For more information on the pre-programmed calibration value, see the section "Calibration Byte" on page 467.

Table 11-3. Internal Calibrated RC Oscillator Operating Modes(1)(2)

Frequency Range (MHz)	CKSEL3:0
9.6 ... 22.4	0010

Notes: 1. The device is shipped with this option selected.

When this Oscillator is selected, start-up times are determined by the SUT Fuses as shown in the following table.

Table 11-4. Start-up times for the internal calibrated RC Oscillator clock selection

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset	SUT1:0
BOD enabled	6 CK	14CK	00
Fast rising power	6 CK	14CK + 4.0 ms	01
Slowly rising power	6 CK	14CK + 64 ms(1)	10
Reserved			11

Notes: 1. The device is shipped with this option selected

11.5 128 kHz Internal Oscillator

The 128 kHz Internal Oscillator is an ultra-low power RC oscillator providing a clock of approximate 128 kHz nominal frequency. This clock may be selected as the system clock by programming the CKSEL Fuses to “0011” as shown in the following table.

Table 11-5. 128 kHz Internal Oscillator Operating Modes(1)

Nominal Frequency	CKSEL3:0
128 kHz	0011

Notes: 1. Note that the 128 kHz oscillator is a very low power clock source, and is not designed for high accuracy

When this clock source is selected, start-up times are determined by the SUT Fuses as shown in the following table.

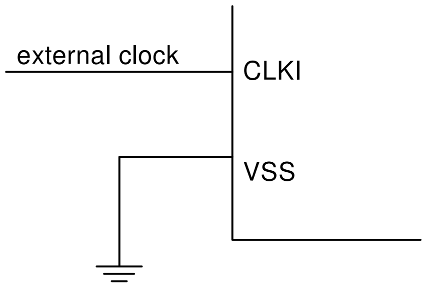
Table 11-6. Start-up Times for the 128 kHz Internal Oscillator

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset	SUT1:0
BOD enabled	6 CK	14CK	00
Fast rising power	6 CK	14CK + 4.1 ms	01
Slowly rising power	6 CK	14CK + 64 ms	10
Reserved			11

11.6 External Clock

To drive the device from an external clock source, CLKI should be used as shown in Figure 11-2 on page 151. To run the device on an external clock, the CKSEL Fuses must be programmed to “0000”.

Figure 11-2. External Clock Drive Configuration



When this clock source is selected, start-up times are determined by the SUT Fuses as shown in Table 11-8 below.

Table 11-7. External Clock Frequency

Nominal Frequency	CKSEL3:0
0 – 16 MHz	0000

Table 11-8. Start-up Times for the External Clock Selection

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset	SUT1:0
BOD enabled	6 CK	14 CK	00
Fast rising power	6 CK	14 CK + 4.0 ms	01
Slowly rising power	6 CK	14 CK + 64 ms	10
Reserved			11

When applying an external clock, it is required to avoid sudden changes in the applied clock frequency to ensure stable operation of the microcontroller unit (MCU). A variation in frequency of more than 2% from one clock cycle to the next can lead to unpredictable behavior. If changes of more than 2% are required, ensure that the MCU is kept in Reset during the changes.

Note that the System Clock Prescaler can be used to implement run-time changes of the internal clock frequency while still ensuring stable operation. Refer to section "System Clock Prescaler" on page 152 for details.

11.7 Transceiver Crystal Oscillator

The integrated crystal oscillator for the radio transceiver generates a low-jitter 16MHz clock frequency. See section "Crystal Oscillator (XOSC)" on page 80 for details about the operation of this oscillator. The AVR core and the radio transceiver operate synchronously on the same clock if this oscillator is selected. If the transceiver crystal oscillator is selected as AVR core clock, it remains enabled even if the radio transceiver is in SLEEP mode or its power reduction bit PRTRX24 is set.

Table 11-9. Transceiver Crystal Clock Operating Mode

Frequency Range (MHz)	CKSEL3:0[(1)]
16	1111 - 0110

Notes: 1. All CKSEL fuse values have the same significance.

Table 11-10. Start-up Times for the Transceiver Oscillator Clock Selection

Power Conditions	Start-up Time from Power-down and Power-save	Additional Delay from Reset	CKSEL0	SUT1:0
fast rising power	258 CK	14CK + 4.1 ms	0	00
slowly rising power	258 CK	14CK + 65 ms	0	01
BOD enabled	1K CK	14CK + 0 ms	0	10
fast rising power	1K CK	14CK + 4.1 ms	0	11
slowly rising power	1K CK	14CK + 65 ms	1	00
BOD enabled	16K CK	14CK + 0 ms	1	01
fast rising power	16K CK	14CK + 4.1 ms	1	10
slowly rising power	16K CK	14CK + 65 ms	1	11

11.8 Clock Output Buffer

The device can output the system clock on the CLKO pin. To enable the output, the CKOUT Fuse has to be programmed. This mode is suitable when the chip clock is used to drive other circuits on the system. The clock also will be output during reset, and the normal operation of I/O pin will be overridden when the fuse is programmed. Any clock source, including the internal RC Oscillator, can be selected when the clock is output on CLKO. If the System Clock Prescaler is used, it is the divided system clock that is output.

Special attention is required to prevent unwanted radiation from the connected PCB clock trace. Proper filtering can help to suppress higher harmonics.

11.9 Timer/Counter Oscillator

The device can operate the Timer/Counter2 from the 32.768 kHz crystal oscillator or an external clock source. See section "Application Circuits" on page 493 for the watch crystal connection.

11.10 System Clock Prescaler

The ATmega128RFA1 has a system clock prescaler, and the system clock can be divided by setting the "CLKPR – Clock Prescale Register". This feature can be used to decrease the system clock frequency and the power consumption when the requirement for processing power is low. This can be used with all clock source options, and it will affect the clock frequency of the CPU and all synchronous peripherals. The clocks $clk_{I/O}$, clk_{ADC}, clk_{CPU}, and clk_{FLASH} are divided by a factor as shown in CLKPR – Clock Prescale Register on page 154.

The prescaler clock division factor of the internal RC-Oscillator is different from all other clock sources, see register description CLKPR – Clock Prescale Register on page 154

Flash, EEPROM, Fuse- and Lock-bit programming is not allowed while using RC-Oscillator with CLKPS=0xF (clk_{CPU} = 16MHz).

When switching between prescaler settings, the System Clock Prescaler ensures that no glitches occur in the clock system. It also ensures that no intermediate frequency is higher than neither the clock frequency corresponding to the previous setting nor the clock frequency corresponding to the new setting.

The prescaler is implemented as a ripple counter running at the frequency of the undivided clock, which may be faster than the CPU's clock frequency. Hence, it is not possible to determine the state of the prescaler - even if it were readable. The exact time it takes to switch from one clock division to another cannot be exactly predicted. From the time the CLKPS values are written, it takes between $t_1 + t_2$ and $t_1 + 2t_2$ before the new clock frequency is active. In this interval 2 active clock edges are produced. Here t_1 is the previous clock period and t_2 is the clock period corresponding to the new prescaler setting.

To avoid unintentional changes of clock frequency, a special write procedure must be followed to change the CLKPS bits:

1. Write the Clock Prescaler Change Enable (CLKPCE) bit to one and all other bits in CLKPR to zero.
2. Within four cycles, write the desired value to CLKPS while writing a zero to CLKPCE.

Interrupts must be disabled when changing prescaler settings to make sure the write procedure is not interrupted.

It is not required to change the prescaler setting of an existing software package written for an 8MHz internal RC oscillator. The change of the prescaler (additional 1:2 divider) is compensated by doubling the RC oscillator frequency of the ATmega128RFA1.

11.11 Register Description

11.11.1 OSCCAL – Oscillator Calibration Value

Bit	7	6	5	4	3	2	1	0	
NA ($66)	CAL7	CAL6	CAL5	CAL4	CAL3	CAL2	CAL1	CAL0	OSCCAL
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Oscillator Calibration Register is used to trim the Calibrated Internal RC Oscillator to remove process variations from the oscillator frequency. A preprogrammed calibration value is automatically written to this register during chip reset, giving the Factory calibrated frequency. The application software can write this register to change the oscillator frequency. The oscillator can be calibrated to frequencies as specified in the section "Electrical Characteristics". 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. The calibration to very high frequencies can cause EEPROM or Flash erase/write failures. The CAL7 bit determines the range of operation for the oscillator. Setting this bit to 0 gives the lowest frequency range, setting this bit to 1 gives the highest frequency range. The two frequency ranges are overlapping, in other words a setting of OSCCAL = 0x7F gives a higher frequency than OSCCAL = 0x80. The CAL6..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 0x7F gives the highest frequency in the range.

- **Bit 7:0 – CAL7:0 - Oscillator Calibration Tuning Value**

Table 11-11 CAL Register Bits

Register Bits	Value	Description
CAL7:0	0x00	Calibration value for lowest oscillator frequency
	0x7f	End value of low frequency range calibration
	0x80	Start value of high frequency range calibration

Register Bits	Value	Description
	0xff	Calibration value for highest oscillator frequency

11.11.2 CLKPR – Clock Prescale Register

Bit	7	6	5	4	3	2	1	0	
NA ($61)	CLKPCE	Res2	Res1	Res0	CLKPS3	CLKPS2	CLKPS1	CLKPS0	CLKPR
Read/Write	RW	R	R	R	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – CLKPCE - Clock Prescaler Change Enable**

The CLKPCE bit must be written to logic one to enable change of the CLKPS bits. The CLKPCE bit is only updated when the other bits in CLKPR are simultaneously written to zero. CLKPCE is cleared by hardware four cycles after it is written or when CLKPS bits are written. Rewriting the CLKPCE bit within this time-out period does neither extend the time-out period, nor clear the CLKPCE bit.

- **Bit 6:4 – Res2:0 - Reserved**
- **Bit 3:0 – CLKPS3:0 - Clock Prescaler Select Bits**

These bits define the division factor between the selected clock source and the internal system clock. These bits can be written run-time to vary the clock frequency to suit the application requirements. As the divider divides the master clock input to the MCU, the speed of all synchronous peripherals is reduced when a division factor is used. The division factors are given in the following table. Note that the factor is different when using the internal 16MHz RC oscillator as the clock source. The CKDIV8 Fuse determines the initial value of the CLKPS bits. If CKDIV8 is not programmed, the CLKPS bits will be reset to 0000. If CKDIV8 is programmed, CLKPS bits are reset to 0011 giving a division factor of 8 at start up. This feature should be used if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. Note that any value can be written to the CLKPS bits regardless of the CKDIV8 Fuse setting. The Application software must ensure that a sufficient division factor is chosen if the selected clock source has a higher frequency than the maximum frequency of the device at the present operating conditions. The device is shipped with the CKDIV8 Fuse programmed.

Table 11-12 CLKPS Register Bits

Register Bits	Value	Description
CLKPS3:0	0x0	Division factor 1 / RC-Oscillator 2
	0x1	Division factor 2 / RC-Oscillator 4
	0x2	Division factor 4 / RC-Oscillator 8
	0x3	Division factor 8 / RC-Oscillator 16
	0x4	Division factor 16 / RC-Oscillator 32
	0x5	Division factor 32 / RC-Oscillator 64
	0x6	Division factor 64 / RC-Oscillator 128
	0x7	Division factor 128 / RC-Oscillator 256
	0x8	Division factor 256 / RC-Oscillator 512
	0x9	Reserved
	0xA	Reserved

Register Bits	Value	Description
	0xB	Reserved
	0xC	Reserved
	0xD	Reserved
	0xE	Reserved
	0xF	Division factor 1 only permitted for RC-Oscillator. Flash and EEPROM programming is not allowed.

12 Power Management and Sleep Modes

Sleep modes enable the application to shut down unused modules in the MCU, thereby saving power. The AVR microcontroller and the RF transceiver provide various sleep modes allowing the user to tailor the power consumption to the application's requirements.

12.1 Deep-Sleep Mode

When the microcontroller goes into Power-down or Power-save modes while the transceiver is in SLEEP state the device enters the Deep-Sleep mode.

Sending the microcontroller to Power-down or Power-save is not allowed during the wake-up phase of the transceiver. The TRX24_AWAKE interrupt shall be used to wait for the transceiver is operational.

The DVDD voltage regulator and the associated power chain will be switched off. Remaining running logic will then be supplied from the Low Leakage Voltage Regulator. Even the AVDD regulator will switched off. See chapter "Radio Transceiver" on page 160 how to disable the radio transceiver.

The SRAM blocks use the data retention mode to preserve its content while saving leakage power. The Low Leakage Voltage Regulator has only limited driving capabilities, see section "Supply Voltage and Leakage Control" on page 161 for details. Therefore the remaining running logic must be clocked with low frequencies only.

The Deep-Sleep mode can be finished by a wake-up source shown by the Table 12-1 below. Then DVDD voltage regulator and the associated power chain will be switched on. If the power-chain is completely enabled the standard AVR wake-up procedure continues (for details see chapter "Power-chain" on page 161).

Note that the wake-up time from Deep-sleep mode is significantly longer than the wake-up time from the Power-down or Power-save mode because the entire power-chain will be restarted.

Additionally note that if the ADC is enabled and/or running a conversion, while entering Deep-sleep mode, the ADC supply voltage is switched off. Therefore the ADC must be disabled before entering Deep-sleep mode to avoid an undefined ADC operation.

12.2 AVR Microcontroller Sleep Modes

In chapter "System Clock and Clock Options" on page 147 the different clock systems in the ATmega128RFA1, and their distribution were presented. Figure 11-1 on page 147 is helpful in selecting an appropriate sleep mode. The following table shows the different sleep modes and their wake-up sources.

Table 12-1. Active Clock Domains and Wake-up Sources in the Different Sleep Modes

	Active Clock Domains					Oscillators		Wake-up Sources								
Sleep Mode	**clkCPU**	**clkFLASH**	**clkIO**	**clkADC**	**clkASY**	**Main Clock-source Enabled**	**Timer Oscillator Enabled**	**INT7:0 and Pin Change**	**TWI Address Match**	**Timer/Counter2**	**SPM/EEPROM Ready**	**ADC**	**WDT Interrupt**	**Other I/O**	**Symbol Counter**	**Transceiver**
Idle			X	X	X	X	X[2]	X	X	X	X	X	X	X	X	X
ADCNRM				X	X	X	X[2]	X[3]	X	X[2]	X	X	X		X	X

	Active Clock Domains					Oscillators		Wake-up Sources								
Power-down								X[(3)]	X				X		X	X
Power-save					X		X[(2)]	X[(3)]	X	X			X		X	X
Standby [(1)]						X		X[(3)]	X				X		X	X
Extended Standby					X[(2)]	X	X[(2)]	X[(3)]	X	X			X		X	X

Notes: 1. Only recommended with external crystal or resonator selected as clock source.

2. If Timer/Counter2 is running in asynchronous mode.

3. For INT7:4, only level interrupt.

To enter any of the sleep modes, the SE bit in in the SMCR register (see "SMCR – Sleep Mode Control Register" on page 166) must be written to logic one and a SLEEP instruction must be executed. The SM2, SM1, and SM0 bits in the SMCR Register select which sleep mode will be activated by the SLEEP instruction. See chapter "Register Description" on page 166 for a summary.

If an enabled interrupt occurs while the MCU is in a sleep mode, the MCU wakes up. The MCU is then halted for four cycles in addition to the start-up time, executes the interrupt routine, and resumes execution from the instruction following SLEEP. The contents of the Register File and SRAM are unaltered when the device wakes up from sleep. Note that SRAM data retention must be enabled in some sleep modes to preserve the memory contents (see section "SRAM with Data Retention" on page 163). If a reset occurs during sleep mode, the MCU wakes up and executes from the Reset Vector.

12.2.1 Idle Mode

When the SM2:0 bits are written to 000 in the SMCR register, the SLEEP instruction makes the MCU enter Idle mode, stopping the CPU but allowing the SPI, USART, Analog Comparator, ADC, 2-wire Serial Interface, Timer/Counters, Watchdog, and the interrupt system to continue operating. This sleep mode basically halts clk_{CPU} and clk_{FLASH}, while allowing the other clocks to run.

Idle mode enables the MCU to wake up from external triggered interrupts as well as internal ones like the Timer Overflow and USART Transmit Complete interrupts. If wake-up from the Analog Comparator interrupt is not required, the Analog Comparator can be powered down by setting the ACD bit in the Analog Comparator Control and Status Register – ACSR. This will reduce power consumption in Idle mode. If the ADC is enabled, a conversion starts automatically when this mode is entered.

12.2.2 ADC Noise Reduction Mode

When the SM2:0 bits are written to 001, the SLEEP instruction makes the MCU enter ADC Noise Reduction mode (ADCNRM), stopping the CPU but allowing the ADC, the external interrupts, 2-wire Serial Interface address match, Timer/Counter2 and the Watchdog to continue operating (if enabled). This sleep mode basically halts $clk_{I/O}$, clk_{CPU}, and clk_{FLASH}, while allowing the other clocks to run.

This improves the noise environment for the ADC, enabling higher resolution measurements. If the ADC is enabled, a conversion starts automatically when this mode is entered. Apart form the ADC Conversion Complete interrupt, only an External Reset, a Watchdog System Reset, a Watchdog interrupt, a Brown-out Reset, a 2-wire serial interface interrupt, a Timer/Counter2 interrupt, an SPM/EEPROM ready interrupt, an external level interrupt on INT7:4 or a pin change interrupt can wakeup the MCU from ADC Noise Reduction mode.

12.2.3 Power-down Mode

When the SM2:0 bits are written to 010, the SLEEP instruction makes the MCU enter Power-down mode. In this mode, the 16 MHz crystal oscillator is stopped (if selected by CKSEL fuses), while the external interrupts, the 2-wire Serial Interface, and the Watchdog continue operating (if enabled). Only an External Reset, a Watchdog Reset, a Brown-out Reset, 2-wire Serial Interface address match, an external level interrupt on INT7:4, an external interrupt on INT3:0, a pin change interrupt, or a symbol counter interrupt can wake up the MCU. This sleep mode basically halts all generated clocks, allowing operation of asynchronous modules only.

Note that if a level triggered interrupt is used for wake-up from Power-down mode, the changed level must be held for some time to wake up the MCU. Refer to section "External Interrupts" on page 218 for details.

When waking up from Power-down mode, there is a delay from the wake-up condition occurs until the wake-up becomes effective. This allows the clock to restart and become stable after have been stopped. The wake-up period is defined by the same CKSEL Fuses that define the Reset Time-out period, as described in chapter "System Clock and Clock Options" on page 147.

12.2.4 Power-save Mode

When the SM2:0 bits are written to 011, the SLEEP instruction makes the MCU enter Power-save mode. This mode is identical to Power-down, with one exception:

If Timer/Counter2 is enabled, it will keep running during sleep. The device can wake up from either Timer Overflow or Output Compare event from Timer/Counter2 if the corresponding Timer/Counter2 interrupt enable bits are set in TIMSK2, and the Global Interrupt Enable bit in SREG is set. If Timer/Counter2 is not running, Power-down mode is recommended instead of Power-save mode.

The Timer/Counter2 can be clocked both synchronously and asynchronously in Power-save mode. If the Timer/Counter2 is not using the asynchronous clock, the Timer/Counter Oscillator is stopped during sleep. If the Timer/Counter2 is not using the synchronous clock, the clock source is stopped during sleep. Note that even if the synchronous clock is running in Power-save, this clock is only available for the Timer/Counter2. Timer/Counter2 operation is described in detail in section "8-bit Timer/Counter2 with PWM and Asynchronous Operation" on page 309.

12.2.5 Standby Mode

When the SM2:0 bits are 110 and the crystal oscillator of the radio transceiver is selected, the SLEEP instruction makes the MCU enter Standby mode. This mode is identical to Power-down with the exception that the Oscillator is kept running. From Standby mode, the device wakes up in six clock cycles.

12.2.6 Extended Standby Mode

When the SM2:0 bits are 111 and the crystal oscillator of the radio transceiver is selected, the SLEEP instruction makes the MCU enter Extended Standby mode. This mode is identical to Power-save mode with the exception that the oscillator is kept running. From Extended Standby mode, the device wakes up in six clock cycles.

12.3 Power Reduction Register

The Power Reduction Register (PRR), see "PRR0 – Power Reduction Register0" on page 167, "PRR1 – Power Reduction Register 1" on page 168 and "PRR2 – Power Reduction Register 2" on page 168, provide a method to stop the clock to individual peripherals to reduce power consumption. The current state of the peripheral is frozen and the I/O registers can not be read or written. Resources used by the peripheral when stopping the clock will remain occupied. Hence the peripheral unit should in most cases be disabled before stopping the clock. Waking up a module, which is done by clearing the bit in PRR, puts the module in the same state as before the shutdown. Exceptions are the SRAM blocks and the radio transceiver. The SRAM is shut down by a DRT switch and the radio transceiver is in reset state if its respective power reduction bit is set.

Module shutdown can be used in Idle mode and Active mode to significantly reduce the overall power consumption. See chapter "Typical Characteristics" on page 510 for examples. In all other sleep modes, the clock is already stopped.

12.4 Minimizing Power Consumption

There are several issues to consider when trying minimizing the power consumption in an AVR controlled system. In general, sleep modes should be used as much as possible, and the sleep mode should be selected so that as few as possible of the device's functions are operating. All functions not needed should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

12.4.1 Analog to Digital Converter

If enabled, the ADC will be enabled in all sleep modes. To save power, the ADC should be disabled before entering any sleep mode. Refer to chapter "ADC – Analog to Digital Converter" on page 410 for details on ADC operation.

12.4.2 Analog Comparator

When entering Idle mode, the Analog Comparator should be disabled if not used. When entering ADC Noise Reduction mode the Analog Comparator should also be disabled. In other sleep modes, the Analog Comparator is automatically disabled. However, if the Analog Comparator is set up to use the Internal Voltage Reference as input, the Analog Comparator should be disabled in all sleep modes. Otherwise, the Internal Voltage Reference will be enabled, independent of sleep mode. Refer to "AC – Analog Comparator" on page 407 for details on how to configure the Analog Comparator.

12.4.3 Brown-out Detector

If the Brown-out Detector is enabled by the BODLEVEL Fuses, it will be disabled in Deep-sleep mode. Refer to "Brown-out Detection" on page 178 for details on how to configure the Brown-out Detector. It is recommended to enable the Brown-out Detector.

12.4.4 Internal Voltage Reference

The Internal Voltage Reference will be enabled when needed by the Brown-out Detection, the Analog Comparator or the ADC. If these modules are disabled as described in the sections above, the internal voltage reference will be disabled and not consume power. When turned on again, the user must allow the reference to start up before the output is used. If the reference is kept on in sleep mode, the output can be used immediately. Refer to "Internal Voltage Reference" on page 179 for details on the start-up time.

12.4.5 Watchdog Timer

If the Watchdog Timer is not needed in the application, the module should be turned off. If the Watchdog Timer is enabled, it will be enabled in all sleep modes, and hence, always consume power. In the deeper sleep modes, this will contribute significantly to the total current consumption. Refer to "Watchdog Timer" on page 180 for details on how to configure the Watchdog Timer.

12.4.6 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. The most important is then to ensure that no pins drive resistive loads. In sleep modes where both the I/O clock ($clk_{I/O}$) and the ADC clock (clk_{ADC}) are stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed. In some cases, the input logic is needed for detecting wake-up conditions, and it will then be enabled. Refer to the section "I/O-Ports" on page 186 for details on which pins are enabled. If the input buffer is enabled and the input signal is left floating or have an analog signal level close to DEVDD/2, the input buffer will use excessive power.
For analog input pins, the digital input buffer should be disabled at all times. An analog signal level close to DEVDD/2 on an input pin can cause significant current even in active mode. Digital input buffers can be disabled by writing to the Digital Input Disable Registers DIDR1 and DIDR0. Refer to "DIDR1 – Digital Input Disable Register 1" on page 409 and "DIDR0 – Digital Input Disable Register 0" on page 433 for details.

12.4.7 On-chip Debug System

If the On-chip debug system is enabled by the OCDEN Fuse and the chip enters sleep mode, the main clock source is enabled, and hence, always consumes power. In the deeper sleep modes, this will contribute significantly to the total current consumption. There are three alternative ways to disable the OCD system:

- Disable the OCDEN Fuse.
- Disable the JTAGEN Fuse.
- Write one to the JTD bit in MCUCR.

12.4.8 Symbol Counter

The Symbol Counter acts as a separate counter, which uses either the 16MHz clock from XTAL1/XTAL2 crystal pins or the clock from PG3/PG4 low frequency crystal pins. If the Symbol Counter module is not used, it should be disabled, see section "MAC Symbol Counter" on page 133.

12.4.9 Radio Transceiver

The radio transceiver module is automatically starting its state machine after power on. While the CPU is in any sleep mode, the radio transceiver remains active. This enables the radio transceiver to wakeup the MCU if a pending action is over (frame received or transmission completed). The radio transceiver will be inactive during sleep, if either the its power reduction bit PRTRX24 in register PRR1 is set or it is send into SLEEP mode, see "PRR1 – Power Reduction Register 1" on page 168 for details.

The radio transceiver is derived from a stand alone solution that was partly controlled by external pins. Now the radio transceiver is fully controlled by individual register bits.

The radio transceiver has a separate reset signal. A radio transceiver reset is initiated by setting bit TRXRST in register TRXPR. This bit is self-resetting.

The radio transceiver signal SLPTR can be controlled by the bit SLPTR in register TRXPR and is used to set the radio transceiver into SLEEP mode (assuming TRX_STATE is TRX_OFF). This bit has a multiple function, see section "Low-Power 2.4 GHz Transceiver" on page 29 for a detailed description of the radio transceiver.

12.5 Supply Voltage and Leakage Control

For battery applications using deep sleep periods, the leakage current defines the system life time. Due to the typical strong temperature dependency of the leakage current, major contributors to the leakage budget are turned off:

- Analog and digital voltage regulator,
- Non-volatile memory (NVM),
- SRAM,
- Digital signal processor of the radio transceiver including AES engine.

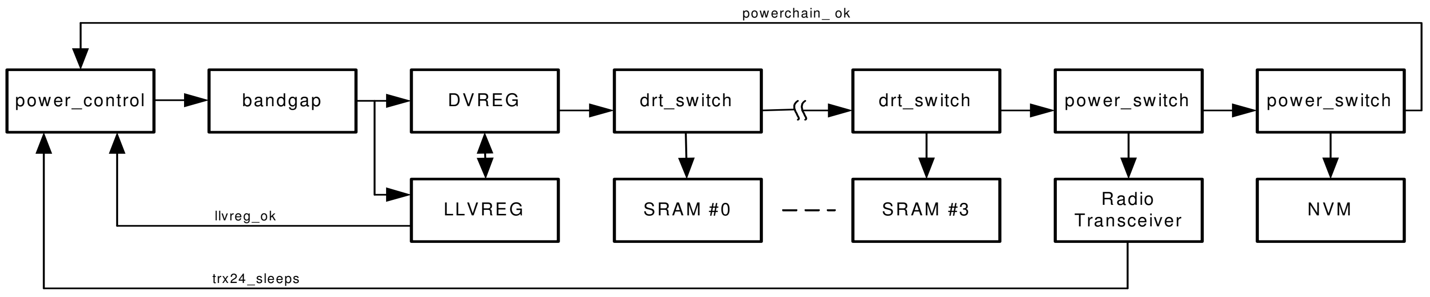
If the CPU uses one of the sleep modes “power-down” or “power-save”, the above mentioned blocks will be switched off by power switches. When the CPU wakes up, the blocks are switched on again. There are some additional exceptions (internal voltage regulator, SRAM, radio transceiver), see section "Power-chain" below .

The supply voltage control is mainly hidden to the application, it is not necessary to configure the supply voltage control. Nevertheless some configurations can be done in order to get the maximum effect and the lowest sleep current, for details see section "SRAM with Data Retention" on page 163.

12.5.1 Power-chain

The following figure shows the major dependencies of the power-chain and how the power switches are situated inside the chain.

Figure 12-1. Power-chain connections



Startup and Wakeup from deep sleep

After power-on reset (POR) or wakeup from deep sleep the power switches of the blocks will be enabled one after another (power-chained) to decrease current peaks. The blocks will be enabled in the following order:

1. Bandgap reference and voltage regulator,
2. Digital voltage regulator (DVREG) and low leakage voltage regulator (LLVREG),
3. SRAM block #0 (lower 4k bytes),
4. SRAM block #1,
5. SRAM block #2,

6. SRAM block #3 (upper 4k bytes),
7. Radio transceiver including AES engine,
8. Non-volatile memory.

If the power-chain is completely enabled the standard AVR wake-up procedure continues.

Figure 12-2 shows the chained startup procedure after power up. A module is only switched on if it is not deselected by power reduction register (PRR1 or PRR2). This is possible for SRAM blocks and radio transceiver power switch. At the end of the startup, the pin RSTON is enabled.

Figure 12-3 shows a similar procedure, the startup from deep sleep.

For further timing information see "Power Management Electrical Characteristics" on page 503.

Figure 12-2. Timing visualization of power up

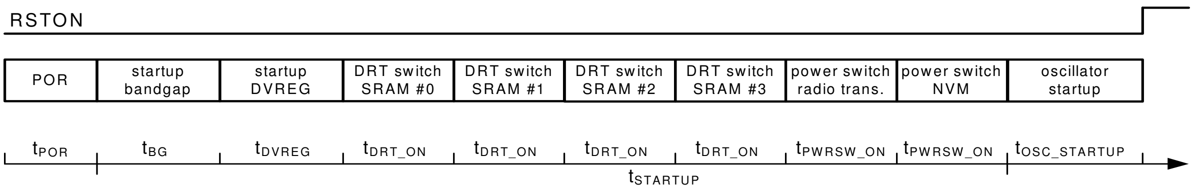
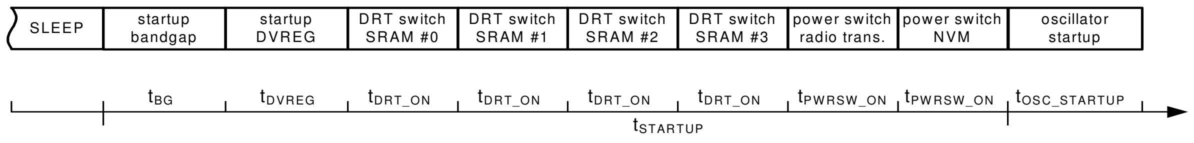


Figure 12-3. Timing visualization of wakeup from deep sleep



Sleep

Six sleep modes are defined for the CPU. Disabling the power-chain and thus switching off of the above mentioned blocks makes only sense for the modes “power-down” and “power-save”. Also an enabled radio transceiver prevents the power-chain from being disabled.

In order to disable the power-chain, one of the following conditions must fit:

- The radio transceiver has to be disabled (power reduction register PRR1 bit PRTRX24).
- The radio transceiver is sent into SLEEP mode (register TRXPR bit SLPTR).

The SRAM blocks may be configured separately to decrease their leakage current (see section "SRAM with Data Retention" on page 163).

The following table shows the different implemented sleep modes and the behavior of the power-chain depending on the current state of the radio transceiver.

Table 12-2. Power states of microcontroller and radio transceiver

AVR State	Radio Transceiver State	Powerchain
ON	ON	ON
ON	off (SLEEP or power reduction)	ON
off (1…6)	ON	ON
off (1,4…6)	off (SLEEP or power reduction)	ON

AVR State	Radio Transceiver State	Powerchain
off [2,3] DEEP SLEEP	off (SLEEP or power reduction)	off [7]

Notes:
1. Idle
2. Power Down
3. Power Save
4. ADC Noise Reduction Mode
5. Standby
6. Extended Standby
7.

12.5.2 SRAM with Data Retention

It is necessary to prevent any data loss of the SRAM when setting the CPU in one of the sleep modes. For that purpose the SRAM blocks will not be completely switched off if the power-chain is disabled. Instead the supply voltage for any individual SRAM block is decreased to reduce its leakage current but guaranteeing its data retention.

The SRAM memory is divided into four separate blocks. Each block can be fully switched off by setting the correspondent bit (PRRAM0 ... PRRAM3) in register PRR2 (see "PRR2 – Power Reduction Register 2" on page 168). This enables the application software to switch off unused SRAM memory to save power and to reduce leakage currents.

Every SRAM block can be enabled again by resetting the respective bit (PRRAM0 ... PRRAM3) of register PRR2. For each SRAM block *n* the bit DRTSWOK of the corresponding register DRTRAM*n* shows the state of the DRT switch (logic high means SRAM block can be accessed).

If the power-chain is switched off during deep-sleep modes, the content of the SRAM blocks must be sustained. To provide data retention and lowest leakage current, a data retention block controls the SRAM behavior during deep-sleep. Since the leakage current is dramatically depending from the voltage of the SRAM, the supply voltage can be decreased by enabling the data retention mode DRT.

Every SRAM block *n* is controlled by its assigned register DRTRAM*n*. The bit ENDRT enables the data retention mode during deep-sleep. If this bit is zero, the respective SRAM block is completely switched off.

Table 12-3. SRAM behavior while in deep-sleep mode

ENDRT	Power-chain	SRAM supply voltage
1	ON	1.8V (DVDD)
0	ON	1.8V (DVDD)
1	off	Reduced
0	off	Disconnected

The lower 4-bit of the register DRTRAM*n* are reserved and should not be changed. The reset value of the DRT voltage settings are preprogrammed during the manufacturing process and need not to be changed.

12.5.3 Voltage Regulators (AVREG, DVREG)

The main features of the Voltage Regulator blocks are:

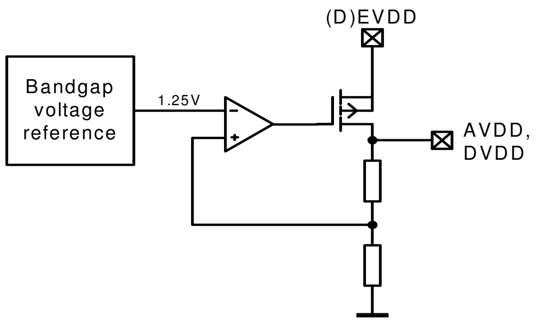
- Bandgap stabilized 1.8V supply for analog and digital domain;

- Low dropout (LDO) voltage regulator;
- Configurable to use an external voltage regulator;

The internal voltage regulators supply a stabilized voltage to the ATmega128RFA1. The AVREG provides the regulated 1.8V supply voltage for the analog section and the DVREG supplies the 1.8V supply voltage for the digital section. The DVREG is enabled during startup and is switched off if the power-chain is disabled. The AVREG is enabled only on request by either the A/D converter or the radio transceiver.

A simplified schematic of the internal voltage regulator is shown in Figure 12-4 below.

Figure 12-4. Simplified Schematic of AVREG/DVREG



The voltage regulators require bypass capacitors for stable operation. The value of the bypass capacitors determines their settling time. The bypass capacitors shall be placed as close as possible to the pins and shall be connected to ground with the shortest possible traces.

The voltage regulators can be configured with the register VREG_CTRL. It is recommended to use the internal regulators but it is also possible to supply the low voltage domains by an external voltage supply. For this configuration the internal regulators need to be switched off by setting the register bits to the values AVREG_EXT = 1 and DVREG_EXT = 1 (see "VREG_CTRL – Voltage Regulator Control and Status Register" on page 115). A regulated external supply voltage of 1.8V must then be connected to the pins 13, 14 (DVDD) and pin 29 (AVDD). When turning on the external supply ensure a sufficiently long stabilization time before interacting with the ATmega128RFA1.

The status bits AVDD_OK = 1 and DVDD_OK = 1 of register VREG_CTRL indicate an enabled and stable internal supply voltage. Reading value 0 indicates that the internal supply voltage is disabled or not yet settled to the final value.

Note that disabling the digital voltage regulator by DVREG_EXT bit is for internal use only.

12.5.4 Low Leakage Voltage Regulator (LLVREG)

The main digital voltage regulator (DVREG) will be switched off during the deep sleep modes “power-down” and “power-save”. The Low Leakage Voltage Regulator will then keep the digital supply voltage to provide data retention. No application software control is required.

During the active power states, when the main voltage regulator supplies the chip, the Low Leakage Voltage Regulator is digitally calibrated. Its output voltage is adjusted to match the output voltage of the main regulator. This fixed calibration result is stored and used when the chip enters a power-down state where the main regulator is switched off.

Because the calibration setting is fixed, temperature and load current variations during the following Deep Sleep period are not regulated out. Thus the output voltage may drift away from the target value. However the design guarantees that for allowed operating conditions the output voltage will stay within valid limits. After every wake-up a new calibration cycle is initiated.

The output driving capability of the Low Leakage Voltage Regulator is limited. Its main purpose is to provide the leakage current of the connected analog and digital blocks.

At least one full calibration cycle of the Low Leakage Voltage Regulator has to be completed before the power-chain can be disabled. Therefore if the CPU uses one of the deep sleep modes “power down” or “power save”, the power-chain is not disabled before the Low Leakage Voltage Regulator completed this first calibration cycle.

By default the LLVREG automatically starts the calibration after finishing the power-on reset and the wake-up/start-up procedures (see section "Low Leakage Voltage Regulator Control" below for a detailed description of the Low Leakage Voltage Regulator).

12.5.5 Low Leakage Voltage Regulator Control

The three register LLCR, LLDRL and LLDRH allow the software to monitor the calibration process and to modify or correct the calibration results. The automatic calibration is the normal operation mode. It is an internal process that does not require any software interaction. Nevertheless the calibration is transparent for the user through LLCR, LLDRL and LLDRH (control and data register respectively).

Before the device can enter the sleep mode “power down” or “power save” the first calibration cycle of the Low Leakage Voltage Regulator must be completed to get valid data in LLDRL and LLDRH. The cycle time $\mathbf{t}_{\mathbf{LLVREG\ CALIB}}$ is not fixed. It depends on the temperature, manufacturing process and the frequency of the 128 kHz RC oscillator (independent of the Watchdog setting).

Systems that require very short power-up times may temporarily disable the calibration process by setting bit LLENCAL to 0.

The output voltage of the Low Leakage Voltage Regulator in sleep mode will be the most accurate if constantly calibrated to compensate for any environmental changes (e.g. temperature). However these changes may be slow enough to skip the calibration during some power-up cycles (e.g. calibrate only every 10th power-up time and use the old calibration results during all other times).

After the completion of the power-up process the calibration will start automatically if bit LLENCAL in the control register LLCR is 1 (default). The completion of a calibration cycle is indicated by the bit LLDONE in that same register. After the first cycle the calibration will continue to run until either the device goes into a sleep mode (“power down” or “power save”) or by setting the LLENCAL bit to 0. The output voltage of the Low Leakage Voltage Regulator is then defined by the values in the data register LLDRL and LLDRH and by the bits LLTCO and LLSHORT of the control register.

Write access to the three register is granted when the bit LLENCAL is set to 0. The application software can then modify the calibration results. Higher values in the data register generate lower output voltages in the sleep modes. In general it is not recommended nor required to alter the automatically generated calibration result.

The write access to the three register must follow a certain scheme to be successful. The registers are implemented in the I/O clock domain while the logic of the Low Leakage Voltage Regulator runs with 64 kHz (clock output of the 128 kHz RC oscillator divided by 2). It takes at least two 64 kHz clock cycles before the data written to the register take effect in the regulator circuit. The write access from the software must be aware of this process. Furthermore the value of LLDRH must be written first followed by LLDRL. Otherwise the LLDRH write access will be ignored. The following Assembler code fragment shows a working example. Note the polling of bit 3 *LLCAL* of the LLCR register to verify the completion of the synchronization process.

Assembly Code Example

```
      …
      clr r20
      IOST LLDRH,r18    ; write LLDRH first
      IOST LLDRL,r19    ; write LLDRL second
      IOST LLCR,r20     ; bit 0 cleared = disable automatic calibration
   ; poll LLCAL bit of LLCR to check if automatic calibration is
   ; turned of
   wait_calib:
      IOLD r20,LLCR
      sbrc r20,3
      rjmp wait_calib  ; not executed if bit 3 of LLCR is cleared
      …
```

12.6 Register Description

12.6.1 SMCR – Sleep Mode Control Register

Bit	7	6	5	4	3	2	1	0	
$33 ($53)	Res3	Res2	Res1	Res0	SM2	SM1	SM0	SE	SMCR
Read/Write	R	R	R	R	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The Sleep Mode Control Register contains control bits for power management.

- **Bit 7:4 – Res3:0 - Reserved**
- **Bit 3:1 – SM2:0 - Sleep Mode Select bit 2**

These bits select between the five available sleep modes. Standby modes are only recommended for use with external crystals or resonators.

Table 12-4 SM Register Bits

Register Bits	Value	Description
SM2:0	0x00	Idle
	0x01	ADC Noise Reduction (If Available)
	0x02	Power Down
	0x03	Power Save
	0x04	Reserved
	0x05	Reserved
	0x06	Standby
	0x07	Extended Standby

- **Bit 0 – SE - Sleep Enable**

The SE bit must be written to logic one to make the MCU enter the sleep mode when the SLEEP instruction is executed. To avoid the MCU entering the sleep mode unless it is the programmers purpose, it is recommended to write the Sleep Enable (SE) bit to one just before the execution of the SLEEP instruction and to clear it immediately after waking up.

12.6.2 PRR0 – Power Reduction Register0

Bit	7	6	5	4	3	2	1	0	
NA ($64)	PRTWI	PRTIM2	PRTIM0	PRPGA	PRTIM1	PRSPI	PRUSART0	PRADC	PRR0
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 – PRTWI - Power Reduction TWI**

Writing a logic one to this bit shuts down the TWI by stopping the clock to the module. When waking up the TWI again, the TWI should be re initialized to ensure proper operation.

- **Bit 6 – PRTIM2 - Power Reduction Timer/Counter2**

Writing a logic one to this bit shuts down the Timer/Counter2 module. When the Timer/Counter2 is enabled, operation will continue like before the shutdown.

- **Bit 5 – PRTIM0 - Power Reduction Timer/Counter0**

Writing a logic one to this bit shuts down the Timer/Counter0 module. When the Timer/Counter0 is enabled, operation will continue like before the shutdown.

- **Bit 4 – PRPGA - Power Reduction PGA**

Writing a logic one to this bit reduced the power consumption of the programmable gain amplifier. The block is not turned off. Only the current levels in the amplifiers are reduced. Reducing the PGA current levels is only recommended for slow ADC clock frequencies. A new ADC conversion using the PGA should be delayed by a default start-up time after changing (setting or resetting) this bit.

- **Bit 3 – PRTIM1 - Power Reduction Timer/Counter1**

Writing a logic one to this bit shuts down the Timer/Counter1 module. When the Timer/Counter1 is enabled, operation will continue like before the shutdown.

- **Bit 2 – PRSPI - Power Reduction Serial Peripheral Interface**

Writing a logic one to this bit shuts down the Serial Peripheral Interface by stopping the clock to the module. When waking up the SPI again, the SPI should be re initialized to ensure proper operation.

- **Bit 1 – PRUSART0 - Power Reduction USART**

Writing a logic one to this bit shuts down the USART0 by stopping the clock to the module. When waking up the USART0 again, the USART0 should be reinitialized to ensure proper operation.

- **Bit 0 – PRADC - Power Reduction ADC**

Writing a logic one to this bit shuts down the ADC. The ADC must be disabled (reset ADEN bit in register ADCSRA) before shut down. The analog comparator cannot use the ADC input MUX when the ADC is shut down.

12.6.3 PRR1 – Power Reduction Register 1

Bit	7	6	5	4	3	2	1	0	
NA ($65)	Res	PRTRX24	PRTIM5	PRTIM4	PRTIM3			PRUSART1	PRR1
Read/Write	R	RW	RW	RW	RW			RW	
Initial Value	0	0	0	0	0			0	

- **Bit 7 – Res - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 6 – PRTRX24 - Power Reduction Transceiver**

Writing a logic one to this bit shuts down the transceiver (disconnect from the power supply). In power-down and power-save modes the power-chain will be disabled when this bit is one. Writing a logic zero to this bit will re-enable the transceiver.

- **Bit 5 – PRTIM5 - Power Reduction Timer/Counter5**

Writing a logic one to this bit shuts down the Timer/Counter5 module. When the Timer/Counter5 is enabled, operation will continue like before the shutdown.

- **Bit 4 – PRTIM4 - Power Reduction Timer/Counter4**

Writing a logic one to this bit shuts down the Timer/Counter4 module. When the Timer/Counter4 is enabled, operation will continue like before the shutdown.

- **Bit 3 – PRTIM3 - Power Reduction Timer/Counter3**

Writing a logic one to this bit shuts down the Timer/Counter3 module. When the Timer/Counter3 is enabled, operation will continue like before the shutdown.

- **Bit 0 – PRUSART1 - Power Reduction USART1**

Writing a logic one to this bit shuts down the USART1 by stopping the clock to the module. When waking up the USART1 again, the USART1 should be reinitialized to ensure proper operation.

12.6.4 PRR2 – Power Reduction Register 2

Bit	7	6	5	4	3	2	1	0	
NA ($63)	Res3	Res2	Res1	Res0	PRRAM3	PRRAM2	PRRAM1	PRRAM0	PRR2
Read/Write	R	R	R	R	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The Power Reduction Register PRR2 allows to individually disable all four SRAM blocks. Setting any PRRAM3:0 bit to one will completely switch off (disconnect from the power supply) the corresponding SRAM block. This enables the application to disable unused SRAM memory to save power. Every SRAM block can be re-enabled by reseting the appropriate PRRAM3:0 bit.

- **Bit 7:4 – Res3:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 3 – PRRAM3 - Power Reduction SRAM 3**

Setting this bit to one will disable the SRAM block 3. Setting this bit to zero will enable the SRAM block 3.

- **Bit 2 – PRRAM2 - Power Reduction SRAM 2**

Setting this bit to one will disable the SRAM block 2. Setting this bit to zero will enable the SRAM block 2.

- **Bit 1 – PRRAM1 - Power Reduction SRAM 1**

Setting this bit to one will disable the SRAM block 1. Setting this bit to zero will enable the SRAM block 1.

- **Bit 0 – PRRAM0 - Power Reduction SRAM 0**

Setting this bit to one will disable the SRAM block 0. Setting this bit to zero will enable the SRAM block 0.

12.6.5 TRXPR – Transceiver Pin Register

Bit	7	6	5	4	3	2	1	0	
NA ($139)	Res3	Res2	Res1	Res0	Resx3	Resx2	SLPTR	TRXRST	TRXPR
Read/Write	R	R	R	R	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The register TRXPR allows to control basic actions of the radio transceiver like reset or state transitions. The register bit functionality is inherited from the external pins of the stand-alone radio transceiver.

- **Bit 7:4 – Res3:0 - Reserved**
- **Bit 3:2 – Resx3:2 - Reserved**
- **Bit 1 – SLPTR - Multi-purpose Transceiver Control Bit**

The bit SLPTR is a multi-functional bit to control transceiver state transitions. Dependent on the radio transceiver state, a rising edge of bit SLPTR causes the following state transitions: TRX_OFF => SLEEP (level sensitive), PLL_ON => BUSY_TX. Whereas the falling edge of bit SLPTR causes the following state transition: SLEEP => TRX_OFF (level sensitive). When the radio transceiver is in TRX_OFF state the microcontroller forces the transceiver to SLEEP by setting SLPTR = H. The Transceiver awakes when the microcontroller releases the bit SLPTR. In states PLL_ON and TX_ARET_ON, bit SLPTR is used as trigger input to initiate a TX transaction. Here SLPTR is sensitive on rising edge only. After initiating a state change by a rising edge at Bit SLPTR in radio transceiver states TRX_OFF, RX_ON or RX_AACK_ON, the radio transceiver remains in the new state as long as the pin is logical high and returns to the preceding state with the falling edge.

- **Bit 0 – TRXRST - Force Transceiver Reset**

The RESET state is used to set back the state machine and to reset all registers of the transceiver to their default values. A reset forces the radio transceiver into the TRX_OFF state and resets all transceiver register to their default values. A reset is initiated with bit TRXRST = H. The bit is cleared automatically During transceiver reset the microcontroller has to set the radio transceiver control bit SLPTR to the default value.

12.6.6 DRTRAM0 – Data Retention Configuration Register of SRAM 0

Bit	7	6	5	4	3	2	1	0	
NA ($135)	Res1	Res0	DRTSWOK	ENDRT	Resx3	Resx2	Resx1	Resx0	DRTRAM0
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DRTRAM0 register controls the behavior of SRAM block 0 in the power-states "power-save" and "power-down". To prevent any data loss the SRAM will not completely disconnected from the power supply. Reserved bits will be overwritten during chip reset by the factory calibration and should not be modified.

- **Bit 7:6 – Res1:0 - Reserved**
- **Bit 5 – DRTSWOK - DRT Switch OK**

This bit indicates the status of the SRAM power-switch. A logical one indicates that the SRAM supply voltage is fully available and the memory may be accessed normally.

- **Bit 4 – ENDRT - Enable SRAM Data Retention**

During "Deep-Sleep" each SRAM block will either be switched off or provides data retention of its memory content. This bit must set to one if data retention mode should be used. Otherwise the SRAM is switched off (disconnected from the power supply) and all its data are lost.

- **Bit 3:0 – Resx3:0 - Reserved**

12.6.7 DRTRAM1 – Data Retention Configuration Register of SRAM 1

Bit	7	6	5	4	3	2	1	0	
NA ($134)	Res1	Res0	DRTSWOK	ENDRT	Resx3	Resx2	Resx1	Resx0	DRTRAM1
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DRTRAM1 register controls the behavior of SRAM block 1 in the power-states "power-save" and "power-down". To prevent any data loss the SRAM will not completely disconnected from the power supply. Reserved bits will be overwritten during chip reset by the factory calibration and should not be modified.

- **Bit 7:6 – Res1:0 - Reserved**
- **Bit 5 – DRTSWOK - DRT Switch OK**

This bit indicates the status of the SRAM power-switch. A logical one indicates that the SRAM supply voltage is fully available and the memory may be accessed normally.

- **Bit 4 – ENDRT - Enable SRAM Data Retention**

During "Deep-Sleep" each SRAM block will either be switched off or provides data retention of its memory content. This bit must set to one if data retention mode should be used. Otherwise the SRAM is switched off (disconnected from the power supply) and all its data are lost.

- **Bit 3:0 – Resx3:0 - Reserved**

12.6.8 DRTRAM2 – Data Retention Configuration Register of SRAM 2

Bit	7	6	5	4	3	2	1	0	
NA ($133)	Resx7	Res	DRTSWOK	ENDRT	Resx3	Resx2	Resx1	Resx0	DRTRAM2
Read/Write	RW	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DRTRAM2 register controls the behavior of SRAM block 2 in the power-states "power-save" and "power-down". To prevent any data loss the SRAM will not completely disconnected from the power supply. Reserved bits will be overwritten during chip reset by the factory calibration and should not be modified.

- **Bit 7 – Resx7 - Reserved**
- **Bit 6 – Res - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 5 – DRTSWOK - DRT Switch OK**

This bit indicates the status of the SRAM power-switch. A logical one indicates that the SRAM supply voltage is fully available and the memory may be accessed normally.

- **Bit 4 – ENDRT - Enable SRAM Data Retention**

During "Deep-Sleep" each SRAM block will either be switched off or provides data retention of its memory content. This bit must set to one if data retention mode should be used. Otherwise the SRAM is switched off (disconnected from the power supply) and all its data are lost.

- **Bit 3:0 – Resx3:0 - Reserved**

12.6.9 DRTRAM3 – Data Retention Configuration Register of SRAM 3

Bit	7	6	5	4	3	2	1	0	
NA ($132)	Res1	Res0	DRTSWOK	ENDRT	Resx3	Resx2	Resx1	Resx0	DRTRAM3
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DRTRAM3 register controls the behavior of SRAM block 3 in the power-states "power-save" and "power-down". To prevent any data loss the SRAM will not completely disconnected from the power supply. Reserved bits will be overwritten during chip reset by the factory calibration and should not be modified.

- **Bit 7:6 – Res1:0 - Reserved**
- **Bit 5 – DRTSWOK - DRT Switch OK**

This bit indicates the status of the SRAM power-switch. A logical one indicates that the SRAM supply voltage is fully available and the memory may be accessed normally.

- **Bit 4 – ENDRT - Enable SRAM Data Retention**

During "Deep-Sleep" each SRAM block will either be switched off or provides data retention of its memory content. This bit must set to one if data retention mode should be used. Otherwise the SRAM is switched off (disconnected from the power supply) and all its data are lost.

- **Bit 3:0 – Resx3:0 - Reserved**

12.6.10 LLCR – Low Leakage Voltage Regulator Control Register

Bit	7	6	5	4	3	2	1	0	
NA ($12F)	Res1	Res0	LLDONE	LLCOMP	LLCAL	LLTCO	LLSHORT	LLENCAL	LLCR
Read/Write	R	R	R	R	R	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	1	

This register allows to monitor and to control the calibration process of the low-leakage voltage regulator. The automatic calibration is the normal operation mode. However, certain circumstances may require to disable this automatic process for instance to save power-up time. The results of the automatic calibration can also be modified when required by the application for instance to get a higher or lower output voltage.

- **Bit 7:6 – Res1:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 5 – LLDONE - Calibration Done**

This bit indicates the last state of the calibration algorithm. The data register contents is updated with new calibration data after the bit changed to 1. The bit will only be high for one 64kHz clock period, because a new calibration loop is started automatically.

- **Bit 4 – LLCOMP - Comparator Output**

This bit indicates the output state of the comparator of the low-leakage voltage regulator. In this way the calibration progress can be directly monitored for debug purposes. The state of the bit changes at most every 64kHz clock period.

- **Bit 3 – LLCAL - Calibration Active**

This bit indicates that the automatic calibration is in progress. The analog part of the calibration circuit is powered up if the bit is 1.

- **Bit 2 – LLTCO - Temperature Coefficient of Current Source**

This bit shows the status of the selection of the temperature coefficient. The state of the bit is updated in the course of the automatic calibration. A valid value is present after the LLDONE bit is 1 for the first time. Write access is only enabled when the automatic calibration is turned off (LLENCAL is 0). This bit should not be changed without further information.

- **Bit 1 – LLSHORT - Short Lower Calibration Circuit**

This bit shows the status of the short switch for the lower calibration circuit. The state of the bit is updated in the course of the automatic calibration. A valid value is present after the LLDONE bit is 1 for the first time. If this bit is set to 1 register LLDRL has no function. Write access is only possible when the automatic calibration is turned off (LLENCAL is 0). This bit should not be changed without further information.

- **Bit 0 – LLENCAL - Enable Automatic Calibration**

This bit enables the automatic calibration. The automatic calibration runs if the state of the bit is 1. Write access to the two data register and the bits LLSHORT and LLTCO is then denied. If the state of LLENCAL is 0 then the calibration algorithm is stopped and the output voltage of the low-leakage voltage regulator is defined by the values in the two data register LLDRL and LLDRH and by the bits LLSHORT and LLTCO.

12.6.11 LLDRH – Low Leakage Voltage Regulator Data Register (High-Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($131)	Res2	Res1	Res0	LLDRH4	LLDRH3	LLDRH2	LLDRH1	LLDRH0	LLDRH
Read/Write	R	R	R	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The high-byte of the calibration data can be accessed through this register. Write access is only enabled when the bit LLENCAL of the LLCR register is 0. Then the data bits LLDRH4:0 directly control the output voltage of the low-leakage voltage regulator. Higher numbers generate lower voltages. If the bit LLENCAL is 1 then the results of the automatic calibration are stored.

- **Bit 7:5 – Res2:0 - Reserved**

These bits are reserved for future use.

- **Bit 4:0 – LLDRH4:0 - High-Byte Data Register Bits**

Value of the high-byte calibration result

Table 12-5 LLDRH Register Bits

Register Bits	Value	Description
LLDRH4:0	0x00	Calibration limit for fast process corner/high output voltage
	0x10	Calibration limit for slow process corner/low output voltage

12.6.12 LLDRL – Low Leakage Voltage Regulator Data Register (Low-Byte)

Bit	7	6	5	4	3	2	1	0	
NA ($130)	Res3	Res2	Res1	Res0	LLDRL3	LLDRL2	LLDRL1	LLDRL0	LLDRL
Read/Write	R	R	R	R	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The low-byte of the calibration data can be accessed through this register. Write access is only enabled when the bit LLENCAL of the LLCR register is 0. Then the data bits LLDRL3:0 directly control the output voltage of the low-leakage voltage regulator. Higher numbers generate lower voltages. The contents of this register is meaningless when the bit LLSHORT of the LLCR register is 1. If the bit LLENCAL is 1 then the results of the automatic calibration are stored.

- **Bit 7:4 – Res3:0 - Reserved**

These bits are reserved for future use.

- **Bit 3:0 – LLDRL3:0 - Low-Byte Data Register Bits**

Value of the low-byte calibration result

Table 12-6 LLDRL Register Bits

Register Bits	Value	Description
LLDRL3:0	0x00	Calibration limit for fast process corner/high output voltage
	0x08	Calibration limit for slow process corner/low output voltage

12.6.13 DPDS0 – Port Driver Strength Register 0

Bit	7	6	5	4	3	2	1	0	
NA ($136)	PFDRV1	PFDRV0	PEDRV1	PEDRV0	PDDRV1	PDDRV0	PBDRV1	PBDRV0	DPDS0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The output driver strength can be set individually for each digital I/O port. The following tables show output current levels for a typical supply voltage of DEVDD = 3.3V. Refer to section "Electrical Characteristics" for details.

- **Bit 7:6 – PFDRV1:0 - Driver Strength Port F**

Table 12-7 PFDRV Register Bits

Register Bits	Value	Description
PFDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

- **Bit 5:4 – PEDRV1:0 - Driver Strength Port E**

Table 12-8 PEDRV Register Bits

Register Bits	Value	Description
PEDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

- **Bit 3:2 – PDDRV1:0 - Driver Strength Port D**

Table 12-9 PDDRV Register Bits

Register Bits	Value	Description
PDDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

- **Bit 1:0 – PBDRV1:0 - Driver Strength Port B**

Table 12-10 PBDRV Register Bits

Register Bits	Value	Description
PBDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

12.6.14 DPDS1 – Port Driver Strength Register 1

Bit	7	6	5	4	3	2	1	0	
NA ($137)	Res5	Res4	Res3	Res2	Res1	Res0	PGDRV1	PGDRV0	DPDS1
Read/Write	R	R	R	R	R	R	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The output driver strength can be set individually for each digital I/O port. The following table shows output current levels for a typical supply voltage of DEVDD = 3.3V. Refer to section "Electrical Characteristics" for details.

- **Bit 7:2 – Res5:0 - Reserved**
- **Bit 1:0 – PGDRV1:0 - Driver Strength Port G**

Driver strength can be set for port G except the port pins PG3 and PG4. The leakage current of the ports PG3 and PG4 is reduced.

Table 12-11 PGDRV Register Bits

Register Bits	Value	Description
PGDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

13 System Control and Reset

13.1 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. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa. The circuit diagram in Figure 13-1 on page 177 shows the reset logic."System and Reset Characteristics" on page 502 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.

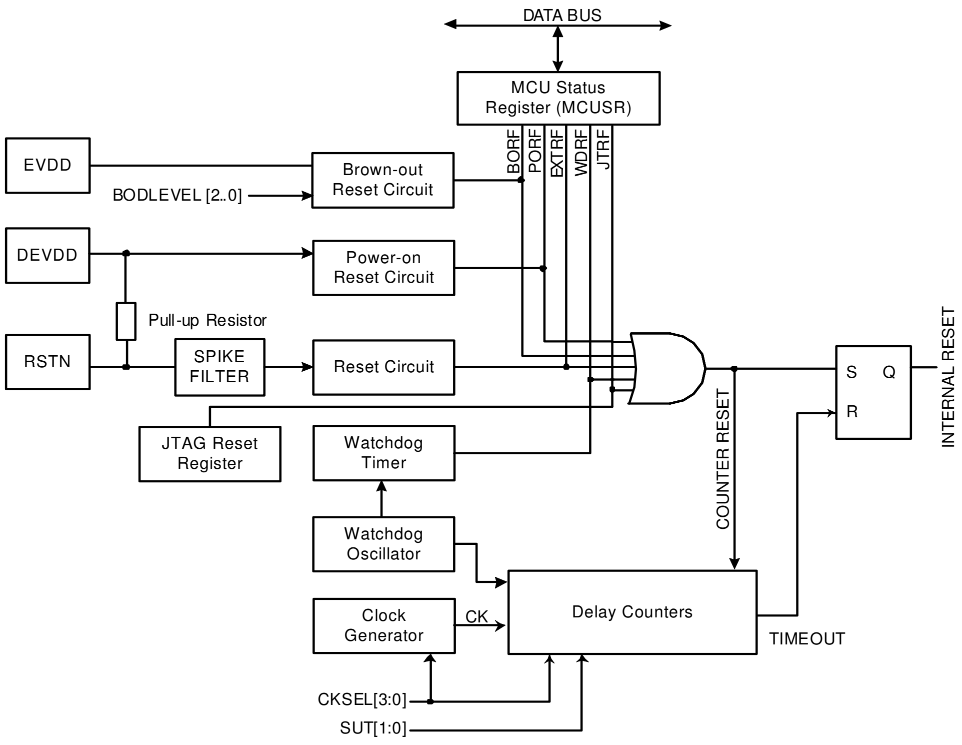
After all reset sources have gone inactive, a delay counter is invoked, stretching the internal reset. This allows the power to reach a stable level before normal operation starts. The time-out period of the delay counter is defined by the user through the SUT and CKSEL Fuses. The different selections for the delay period are presented in "Clock Sources" on page 148.

13.2 Reset Sources

The ATmega128RFA1 has five sources of reset:

- Power-on Reset. The MCU is reset when the supply voltage is below the Power-on Reset threshold (V_{POT}).
- External Reset. The MCU is reset when a low level is present on the RSTN 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.
- Brown-out Reset. The MCU is reset when the supply voltage EVDD is below the Brown-out Reset threshold (V_{BOT}) and the Brown-out Detector is enabled.
- JTAG AVR Reset. The MCU is reset as long as there is a logic one in the Reset Register, one of the scan chains of the JTAG system. Refer to the section "IEEE 1149.1 (JTAG) Boundary-scan" on page 441 for details.

Figure 13-1. Reset Logic



13.2.1 Power-on Reset

A Power-on Reset (POR) pulse is generated by a dynamic, on-chip detection circuit. The POR is active when DEVDD is rising. The electrical characteristics are defined in "System and Reset Characteristics" on page 502. The POR circuit can be used to trigger the start-up reset. To detect a failure in the supply voltage (e.g. a voltage drop) the brown-own detector should be used.

A Power-on Reset (POR) circuit ensures that the device is reset from Power-on. Reaching the Power-on Reset threshold voltage invokes the delay counter, which determines how long the device is kept in RESET after the DEVDD rise. The RESET signal is activated again without any delay, when DEVDD decreases below the detection level.

Figure 13-2. MCU Start-up, RSTN Tied to DEVDD

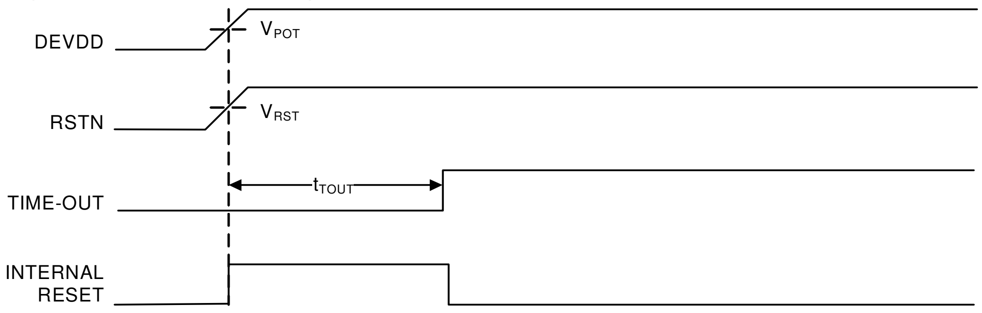
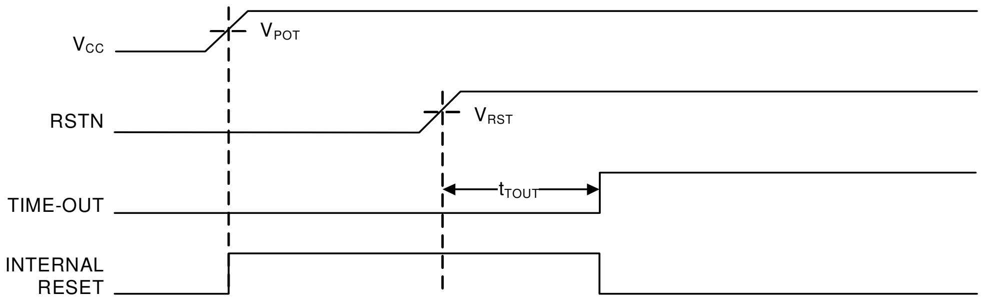


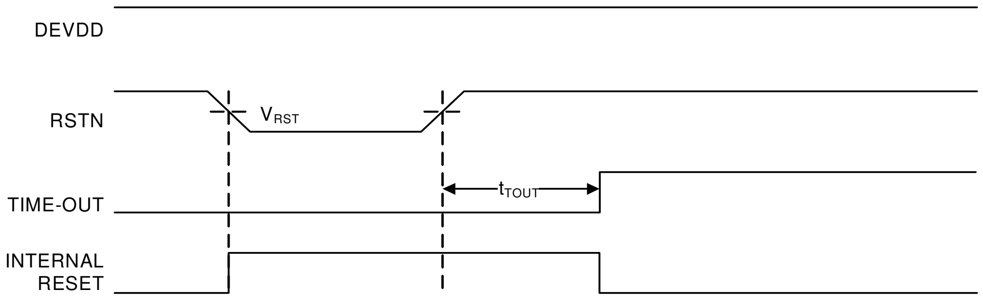
Figure 13-3. MCU Start-up, RSTN Extended Externally



13.2.2 External Reset

An External Reset is generated by a low level on the RSTN pin. Reset pulses longer than the minimum pulse width (see "System and Reset Characteristics" on page 502) will generate a reset, even if the clock is not running. Shorter pulses are not guaranteed to generate a reset. When the applied signal reaches the Reset Threshold Voltage – V_{RST} – on its positive edge, the delay counter starts the MCU after the Time-out period – t_{TOUT} – has expired.

Figure 13-4. Reset During Operation



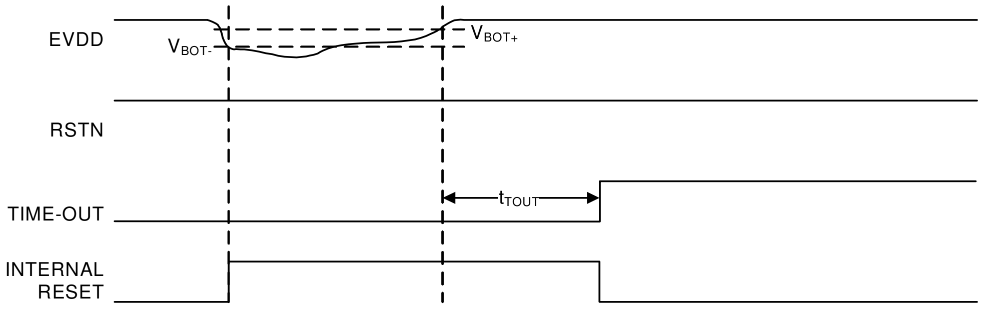
13.2.3 Brown-out Detection

ATmega128RFA1 has an On-chip Brown-out Detection (BOD) circuit for monitoring the EVDD level during operation by comparing it to a fixed trigger level. The trigger level for the BOD can be selected by the BODLEVEL Fuses. The trigger level has a hysteresis to ensure spike free Brown-out Detection. The hysteresis on the detection level should be interpreted as $V_{BOT+} = V_{BOT} + V_{HYST}/2$ and $V_{BOT-} = V_{BOT} - V_{HYST}/2$.

When the BOD is enabled, and EVDD decreases to a value below the trigger level (V_{BOT-} in Figure 13-5 on page 179), the Brown-out Reset is immediately activated. When EVDD increases above the trigger level (V_{BOT+} in Figure 13-5 on page 179), the delay counter starts the MCU after the Time-out period t_{TOUT} has expired.

The BOD circuit will only detect a drop in EVDD if the voltage stays below the trigger level for longer than t_{BOD} given in "System and Reset Characteristics" on page 502.

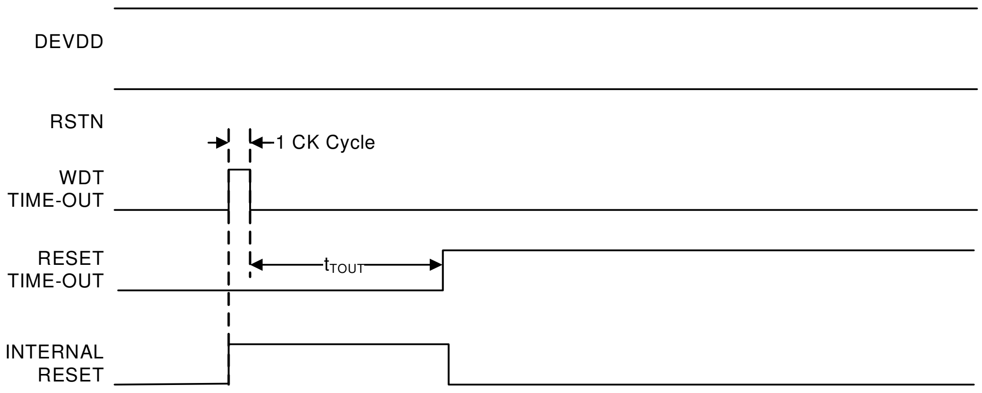
Figure 13-5. Brown-out Reset During Operation



13.2.4 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 Time-out period t_{TOUT}. See "Watchdog Timer" on page 180. for details on operation of the Watchdog Timer.

Figure 13-6. Watchdog Reset During Operation



13.3 Internal Voltage Reference

ATmega128RFA1 features an internal bandgap reference. This reference is used for Brown-out Detection, and it can be used as an input to the Analog Comparator or the ADC.

Voltage Reference Enable Signals and Start-up Time

The voltage reference has a start-up time that may influence the way it should be used. The start-up time is given in "System and Reset Characteristics" on page 502. To save power, the reference is not always turned on. The reference is on during the following situations:

1. When the BOD is enabled (by programming the BODLEVEL [2:0] Fuse).
2. When the bandgap reference is connected to the Analog Comparator (by setting the ACBG bit in ACSR).
3. When the ADC is enabled.

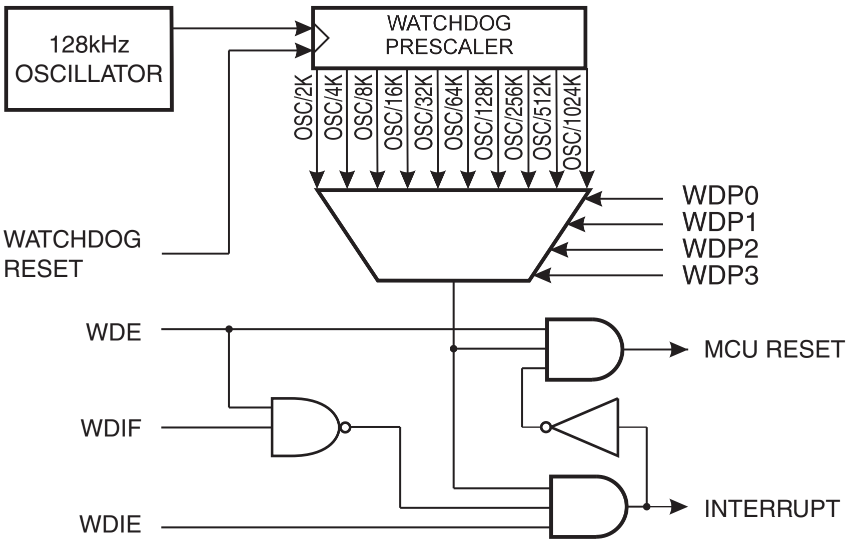
Thus, when the BOD is not enabled, after setting the ACBG bit or enabling the ADC, the user must always allow the reference to start up before the output from the Analog Comparator or ADC is used. To reduce power consumption in Power-down mode, the user can avoid the three conditions above to ensure that the reference is turned off before entering Power-down mode.

13.4 Watchdog Timer

13.4.1 Features

- **Clocked from separate On-chip Oscillator**
- **3 Operating modes**
  - **Interrupt**
  - **System Reset**
  - **Interrupt and System Reset**
- **Selectable Time-out period from 16ms to 8s**
- **Possible Hardware fuse Watchdog always on (WDTON) for fail-safe mode**

Figure 13-7. Watchdog Timer



13.4.2 Overview

ATmega128RFA1 has an Enhanced Watchdog Timer (WDT). The WDT is a timer counting cycles of a separate on-chip 128 kHz oscillator. The WDT gives an interrupt or a system reset when the counter reaches a given time-out value. In normal operation mode, it is required that the system uses the WDR -Watchdog Timer Reset - instruction to restart the counter before the time-out value is reached. If the system doesn't restart the counter, an interrupt or system reset will be issued.

In Interrupt mode, the WDT gives an interrupt when the timer expires. This interrupt can be used to wake the device from sleep-modes, and also as a general system timer. One example is to limit the maximum time allowed for certain operations, giving an interrupt when the operation has run longer than expected. In System Reset mode, the WDT gives a reset when the timer expires. This is typically used to prevent system hang-up in case of runaway code. The third mode, Interrupt and System Reset mode, combines the other two modes by first giving an interrupt and then switch to System Reset mode. This mode will for instance allow a safe shutdown by saving critical parameters before a system reset.

The Watchdog always on (WDTON) fuse, if programmed, will force the Watchdog Timer to System Reset mode. With the fuse programmed the System Reset mode bit (WDE) and Interrupt mode bit (WDIE) are locked to 1 and 0 respectively. To further ensure program security, alterations to the Watchdog set-up must follow timed sequences. The sequence for clearing WDE and changing time-out configuration is as follows:

1. In the same operation, write a logic one to the Watchdog change enable bit (WDCE) and WDE. A logic one must be written to WDE regardless of the previous value of the WDE bit.
2. Within the next four clock cycles, write the WDE and Watchdog prescaler bits (WDP) as desired, but with the WDCE bit cleared. This must be done in one operation.

The following code example shows one assembly and one C function for turning off the Watchdog Timer. The example assumes that interrupts are controlled (e.g. by disabling interrupts globally) so that no interrupts will occur during the execution of these functions.

Assembly Code Example(1)

```
WDT_off:
 ; Turn off global interrupt
 cli
 ; Reset Watchdog Timer
 wdr
 ; Clear WDRF in MCUSR
 in    r16, MCUSR
 andi  r16, (0xff & (0<<WDRF))
 out   MCUSR, r16
 ; Write logical one to WDCE and WDE
 ; Keep old prescaler setting to prevent unintentional time-out
 ldi r16, WDTCSR
 ori r16, (1<<WDCE) | (1<<WDE)
 sts WDTCSR, r16
 ; Turn off WDT
 ldi r16, (0<<WDE)
 sts WDTCSR, r16
 ; Turn on global interrupt
 sei
 ret
```

C Code Example(1)

```
void WDT_off(void)
{
  __disable_interrupt();
  __watchdog_reset();
  /* Clear WDRF in MCUSR*/
  MCUSR &= ~(1<<WDRF);
  /* Write logical one to WDCE and WDE */
  /* Keep old prescaler setting to prevent unintentional time-out */
  WDTCSR |= (1<<WDCE) | (1<<WDE);
  /* Turn off WDT */
  WDTCSR = 0x00;
  __enable_interrupt();
}
```

Note: 1. The example code assumes that the part specific header file is included.

If the Watchdog is accidentally enabled, for example by a runaway pointer or brown-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 time-out resets. To avoid this situation, the application software should always clear the Watchdog System Reset Flag (WDRF) and the WDE control bit in the initialization routine, even if the Watchdog is not in use.

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

Assembly Code Example(1,2)

```
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(time-out) 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(1,2)

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

Note: 1. The example code assumes that the part specific header file is included.
2. The Watchdog Timer should be reset before any change of the WDP bits, since a change in the WDP bits can result in a time-out when switching to a shorter time-out period.

13.5 Register Description

13.5.1 MCUSR – MCU Status Register

Bit	7	6	5	4	3	2	1	0	
$34 ($54)	Res2	Res1	Res0	JTRF	WDRF	BORF	EXTRF	PORF	MCUSR
Read/Write	R	R	R	RW	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The MCU Status Register provides information on which reset source caused an MCU reset. To make use of the Reset Flags to identify a reset condition, the user should read and then Reset the MCUSR as early as possible in the program. If the register is cleared before another reset occurs, the source of the reset can be found by examining the Reset Flags. Note, after power on the bit EXTRF has to be ignored.

- **Bit 7:5 – Res2:0 - Reserved**
- **Bit 4 – JTRF - JTAG Reset Flag**

This bit is set if a reset is being caused by a logic one in the JTAG Reset Register selected by the JTAG instruction AVR_RESET. This bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 3 – WDRF - Watchdog Reset Flag**

This bit is set if a Watchdog Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 2 – BORF - Brown-out Reset Flag**

This bit is set if a Brown-out Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 1 – EXTRF - External Reset Flag**

This bit is set if an External Reset occurs. The bit is reset by a Power-on Reset, or by writing a logic zero to the flag.

- **Bit 0 – PORF - Power-on Reset Flag**

This bit is set if a Power-on Reset occurs. The bit is reset only by writing a logic zero to the flag.

13.5.2 WDTCSR – Watchdog Timer Control Register

Bit	7	6	5	4	3	2	1	0	
NA ($60)	WDIF	WDIE	WDP3	WDCE	WDE	WDP2	WDP1	WDP0	WDTCSR
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – WDIF - Watchdog Timeout Interrupt Flag**

This bit is set when a time-out occurs in the Watchdog Timer and the Watchdog Timer is configured for interrupt. WDIF is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, WDIF is cleared by writing a logic one to the flag. When the I-bit in SREG and WDIE are set, the Watchdog Time-out Interrupt is executed.

- **Bit 6 – WDIE - Watchdog Timeout Interrupt Enable**

When this bit is written to one and the I-bit in the Status Register is set, the Watchdog Interrupt is enabled. If WDE is cleared in combination with this setting, the Watchdog Timer is in Interrupt Mode, and the corresponding interrupt is executed if time-out in the Watchdog Timer occurs. If WDE is set, the Watchdog Timer is in Interrupt and System Reset Mode. The first time-out in the Watchdog Timer will set WDIF. Executing the corresponding interrupt vector will clear WDIE and WDIF automatically by hardware (the Watchdog goes to System Reset Mode). This is useful for keeping the Watchdog Timer security while using the interrupt. To stay in Interrupt and System Reset Mode, WDIE must be set after each interrupt. This should however not be done within the interrupt service routine itself, as this might compromise the safety-function of the Watchdog System Reset mode. If the interrupt is not executed before the next time-out, a System Reset will be applied.

Table 13-1. Watchdog Timer Configuration

WDTON[1]	WDE	WDIE	Mode	Action on Time-out
1	0	0	Stopped	None
1	0	1	Interrupt Mode	Interrupt
1	1	0	System Reset Mode	Reset
1	1	1	Interrupt and System Reset Mode	Interrupt, then go to System Reset Mode
0	x	x	System Reset Mode	Reset

Note: 1. WDTON Fuse set to “0“ means programmed and “1” means un-programmed.

- **Bit 4 – WDCE - Watchdog Change Enable**

This bit is used in timed sequences for changing WDE and prescaler bits. To clear the WDE bit, and/or change the prescaler bits, WDCE must be set. Once written to one, hardware will clear WDCE after four clock cycles.

- **Bit 3 – WDE - Watch Dog Enable**

When the WDE is set (one) the Watchdog Timer is enabled, and if the WDE is cleared (zero) the Watchdog Timer function is disabled. WDE can only be cleared if the WDTOE bit is set (one). To disable an enabled watchdog timer, the following procedure must be followed: 1. In the same operation, write a logical one to WDTOE and WDE. A logical one must be written to WDE even though it is set to one before the disable operation starts. 2. Within the next four clock cycles, write a logical 0 to WDE. This disables the watchdog. WDE is overridden by WDRF in MCUSR. This means that WDE is always set when WDRF is set. To clear WDE, WDRF must be cleared first. This feature ensures multiple resets during conditions causing failure, and a safe start-up after the failure.

- **Bit 5, 2:0 – WDP3:0 – Watchdog Timer Prescaler 3, 2, 1 and 0**

The WDP3:0 bits determine the Watchdog Timer prescaling when the Watchdog Timer is running.

Table 13-2. WDP Register Bits

Register Bits	Value	Description
WDP3:0	0x00	Oscillator Cycles 2k, (16ms)
	0x01	Oscillator Cycles 4k, (32ms)
	0x02	Oscillator Cycles 8k, (64ms)
	0x03	Oscillator Cycles 16k, (0.125s)

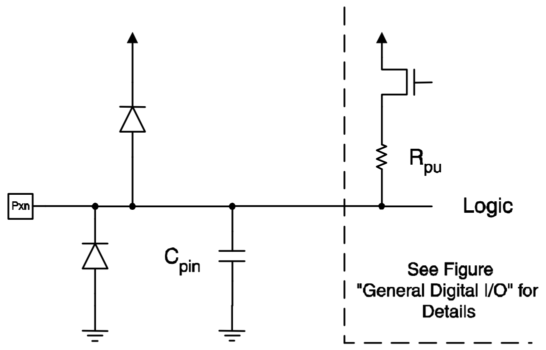
Register Bits	Value	Description
	0x04	Oscillator Cycles 32k, (0.25s)
	0x05	Oscillator Cycles 64k, (0.5s)
	0x06	Oscillator Cycles 128k, (1.0s)
	0x07	Oscillator Cycles 256k, (2.0s)
	0x08	Oscillator Cycles 512k, (4.0s)
	0x09	Oscillator Cycles 1024k, (8.0s)

14 I/O-Ports

14.1 Introduction

All ATmega128RFA1 ports have true Read-Modify-Write functionality when used as general digital I/O ports. This means that the direction of one port pin can be changed without unintentionally changing the direction of any other pin with the SBI and CBI instructions. The same applies when changing drive value (if configured as output) or enabling/disabling of pull-up resistors (if configured as input). Each output buffer has symmetrical drive characteristics with both configurable sink and source capability. Every port is individually configurable in four different drive strengths. The pin driver is strong enough to drive LED displays directly. All port pins have individually selectable pull-up resistors with a supply-voltage invariant resistance. All I/O pins have protection diodes to both DEVDD and DVSS as indicated in Figure 14-1 below. Refer to "Electrical Characteristics" on page 501 for a complete list of parameters.

Figure 14-1. I/O Pin Equivalent Schematic



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, PORTB3 for bit no. 3 in Port B, here documented generally as PORTxn.

Three I/O memory address locations are allocated for each port, one each for the Data Register – PORTx, Data Direction Register – DDRx, and the Port Input Pins – PINx. The Port Input Pins I/O location is read only, while the Data Register and the Data Direction Register are read/write. However, writing a logic one to a bit in the PINx Register, will result in a toggle in the corresponding bit in the Data Register. In addition, the Pull-up Disable – PUD bit in MCUCR disables the pull-up function for all pins in all ports when set.

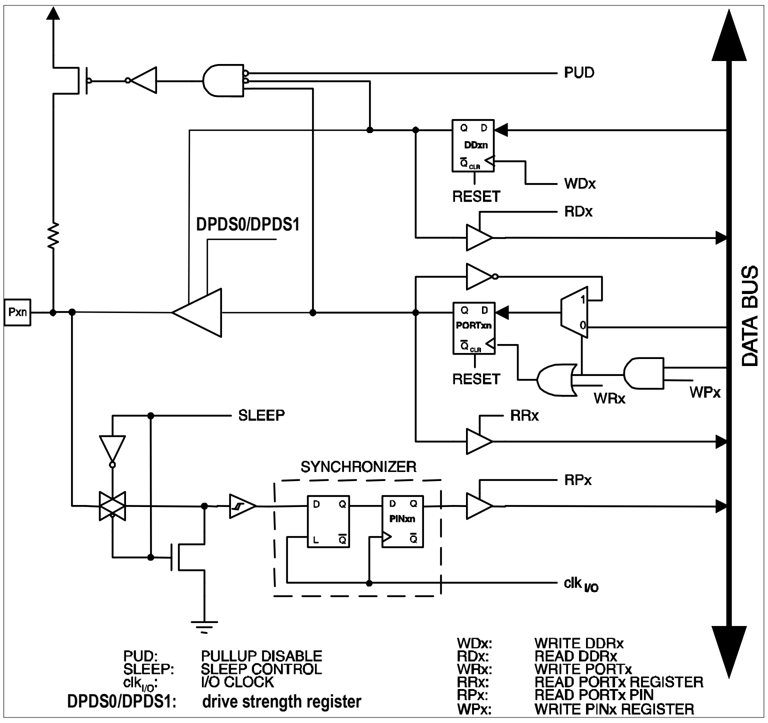
Using the I/O port as General Digital I/O is described in "Ports as General Digital I/O" on page 187. Most port pins are multiplexed with alternate functions for the peripheral features on the device. How each alternate function interferes with the port pin is described in "Alternate Port Functions" on page 191. Refer to the individual module sections for a full description of the alternate functions.

Note that enabling the alternate function of some of the port pins does not affect the use of the other pins in the port as general digital I/O.

14.2 Ports as General Digital I/O

The ports are bi-directional I/O ports with optional internal pull-ups. Figure 14-2 below shows a functional description of one I/O-port pin, here generically called Pxn.

Figure 14-2. General Digital I/O[(1)]



Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports.

14.2.1 Configuring the Port

Drive strength of output buffers is configurable port-wise. Source/sink capably of 2mA, 4mA, 6mA or 8mA is selectable through registers DPDS1 and DPDS0. Note that pins PG3 and PG4 of PORTG have fixed drive strength of 2mA to enable the operation of the low power crystal oscillator.

14.2.2 Configuring the Pin

Each port pin consists of three register bits: DDxn, PORTxn, and PINxn. The DDxn bits are accessed at the DDRx I/O address, the PORTxn bits at the PORTx I/O address, and the PINxn bits at the PINx I/O address.

The DDxn bit in the DDRx Register selects the direction of this pin. If DDxn is written logic one, Pxn is configured as an output pin. If DDxn is written logic zero, Pxn is configured as an input pin.

If PORTxn is written logic one when the pin is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTxn has to be written logic zero or the pin has to be configured as an output pin. The port pins are tri-stated when reset condition becomes active, even if no clocks are running.

If PORTxn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTxn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

14.2.3 Toggling the Pin

Writing a logic one to PINxn toggles the value of PORTxn, independent on the value of DDRxn. Note that the SBI instruction can be used to toggle one single bit in a port.

14.2.4 Switching Between Input and Output

When switching between tri-state ({DDxn, PORTxn} = 0b00) and output high ({DDxn, PORTxn} = 0b11), an intermediate state with either pull-up enabled {DDxn, PORTxn} = 0b01) or output low ({DDxn, PORTxn} = 0b10) must occur. Normally, the pull-up enabled state is fully acceptable, as a high-impedant environment will not notice the difference between a strong high driver and a pull-up. If this is not the case, the PUD bit in the MCUCR Register can be set to disable all pull-ups in all ports.

Switching between input with pull-up and output low generates the same problem. The user must use either the tri-state ({DDxn, PORTxn} = 0b00) or the output high state ({DDxn, PORTxn} = 0b11) as an intermediate step.

The following table summarizes the control signals for the pin value.

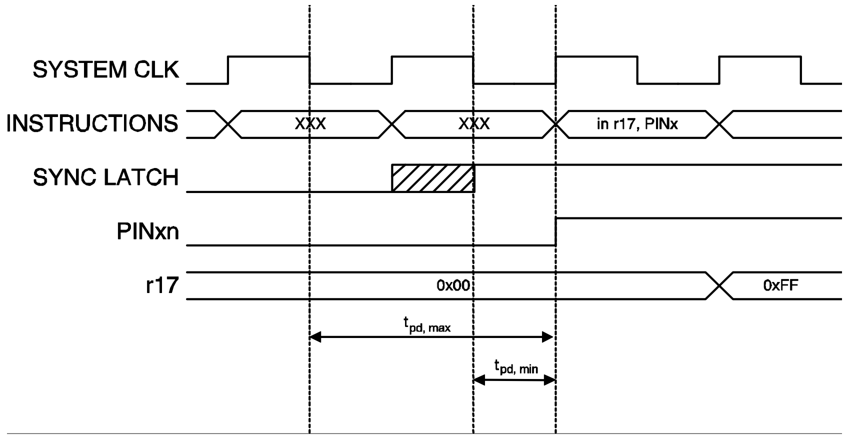
Table 14-1. Port Pin Configurations

DDxn	**PORTxn**	**PUD (in MCUCR)**	**I/O**	**Pull-up**	**Comment**
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

14.2.5 Reading the Pin Value

Independent of the setting of Data Direction bit DDxn, the port pin can be read through the PINxn Register bit. As shown in Figure 14-2 on page 187, the PINxn Register bit and the preceding latch constitute a synchronizer. This is needed to avoid meta-stability if the physical pin changes value near the edge of the internal clock, but it also introduces a delay. Figure 14-3 on page 189 shows a timing diagram of the synchronization when reading an externally applied pin value. The maximum and minimum propagation delays are denoted $t_{PD,MAX}$ and $t_{PD,MIN}$ respectively.

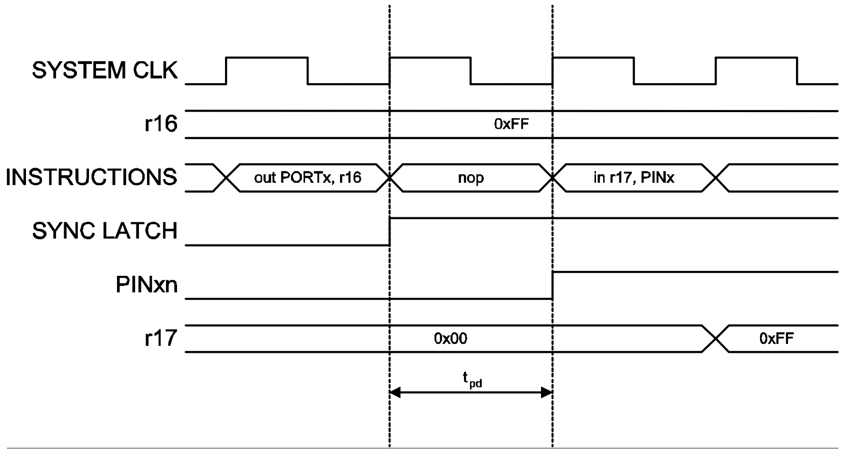
Figure 14-3. Synchronization when reading an external applied pin value



Consider the clock period starting shortly after the first falling edge of the system clock. The latch is closed when the clock is low, and goes transparent when the clock is high, as indicated by the shaded region of the "SYNC LATCH" signal. The signal value is latched when the system clock goes low. It is clocked into the PINxn Register at the succeeding positive clock edge. As indicated by the two arrows $t_{PD,MAX}$ and $t_{PD,MIN}$, a single signal transition on the pin will be delayed between ½ and 1½ system clock period depending upon the time of assertion.

When reading back a software assigned pin value, a NOP instruction must be inserted as indicated in Figure 14-4 below. The out instruction sets the "SYNC LATCH" signal at the positive edge of the clock. In this case, the delay t_{PD} through the synchronizer is 1 system clock period.

Figure 14-4. Synchronization when reading software assigned pin value



The following code example shows how to set port B pins 0 and 1 high, 2 and 3 low, and define the port pins from 4 to 7 as input with pull-ups assigned to port pins 6 and 7. The resulting pin values are read back again, but as previously discussed, a NOP instruction is included to be able to read back the value recently assigned to some of the pins.

Assembly Code Example[1]

```
…
; Define pull-ups and set outputs high
; Define directions for port pins
ldi r16,(1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0)
ldi r17,(1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0)
out PORTB,r16
out DDRB,r17
; Insert nop for synchronization
nop
; Read port pins
in r16,PINB
…
```

C Code Example

```
 unsigned char i;
…
/* Define pull-ups and set outputs high */
/* Define directions for port pins */
PORTB = (1<<PB7)|(1<<PB6)|(1<<PB1)|(1<<PB0);
DDRB = (1<<DDB3)|(1<<DDB2)|(1<<DDB1)|(1<<DDB0);
/* Insert nop for synchronization*/
__no_operation();
/* Read port pins */
i = PINB;
…
```

Note: 1. For the assembly program, two temporary registers are used to minimize the time from pull-ups are set on pins 0, 1, 6, and 7, until the direction bits are correctly set, defining bit 2 and 3 as low and redefining bits 0 and 1 as strong high drivers.

14.2.6 Digital Input Enable and Sleep Modes

As shown in Figure 14-2 on page 187, the digital input signal can be clamped to ground at the input of the Schmitt-Trigger. The signal denoted SLEEP in the figure, is set by the MCU Sleep Controller in Power-down mode, Power-save mode, and Standby mode to avoid high power consumption if some input signals are left floating, or have an analog signal level close to $V_{DEVDD}/2$.

SLEEP is overridden for port pins enabled as external interrupt pins. If the external interrupt request is not enabled, SLEEP is active also for these pins. SLEEP is also overridden by various other alternate functions as described in "Alternate Port Functions" on page 191.

If a logic high level ("one") is present on an asynchronous external interrupt pin configured as "Interrupt on Rising Edge, Falling Edge, or Any Logic Change on Pin" while the external interrupt is not enabled, the corresponding External Interrupt Flag will be set when resuming from the above mentioned Sleep mode, as the clamping in these sleep mode produces the requested logic change.

14.2.7 Unconnected Pins

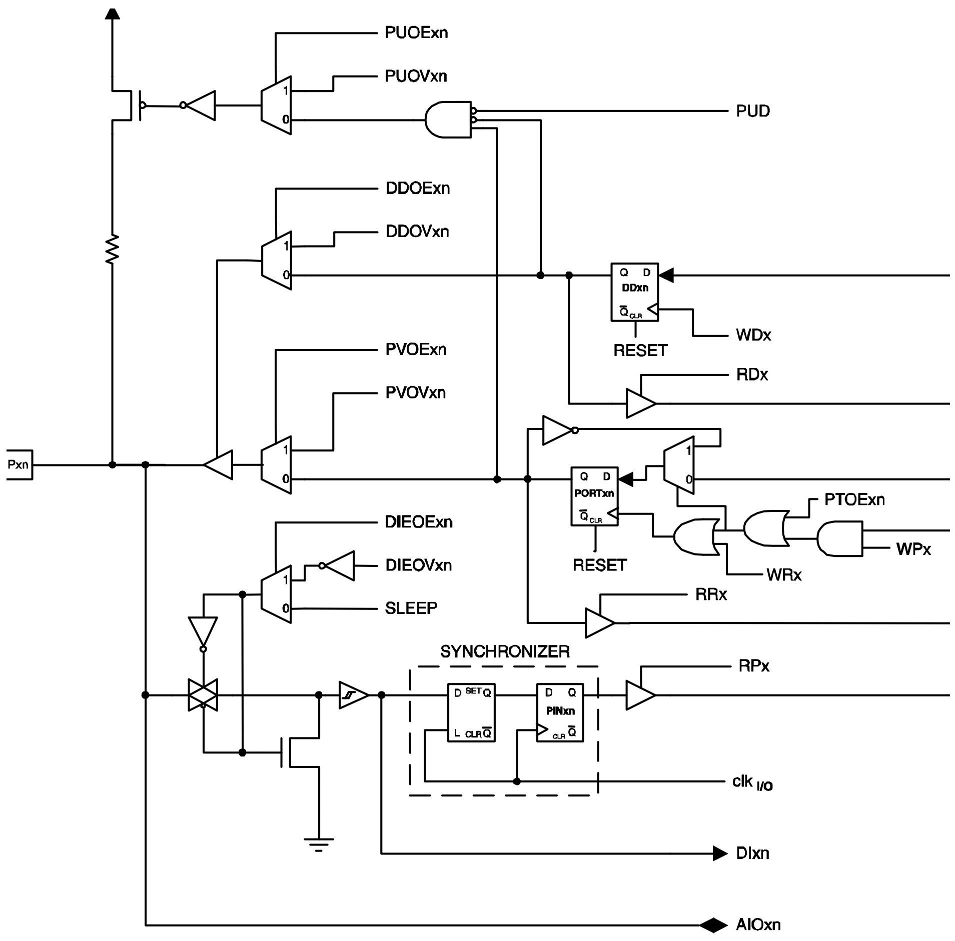
If some pins are unused, it is recommended to ensure that these pins have a defined level. Even though most of the digital inputs are disabled in the deep sleep modes as described above, floating inputs should be avoided to reduce current consumption in all other modes where the digital inputs are enabled (Reset-, Active- and Idle-mode).

The simplest method to ensure a defined level of an unused pin is to enable the internal pull-up. In this case, the pull-up will be disabled during reset. If low power consumption during reset is important, it is recommended to use an external pull-up or pull-down. Connecting unused pins directly to DEVDD or DVSS is not recommended, since this may cause excessive currents if the pin is accidentally configured as an output.

14.3 Alternate Port Functions

Most port pins have alternate functions in addition to being general digital I/O ports. Figure 14-5 below shows how the port pin control signals from the simplified Figure 14-2 on page 187 can be overridden by alternate functions. The overriding signals may not be present in all port pins, but the figure serves as a generic description applicable to all port pins in the AVR microcontroller family.

Figure 14-5. Alternate Port Functions [(1)]



ꟼUOExn:	Pxn PULL-UP OVERRIDE ENABLE	PUD:	PULLUP DISABLE
ꟼUOVxn:	Pxn PULL-UP OVERRIDE VALUE	WDx:	WRITE DDRx
ↃDOExn:	Pxn DATA DIRECTION OVERRIDE ENABLE	RDx:	READ DDRx
ↃDOVxn:	Pxn DATA DIRECTION OVERRIDE VALUE	RRx:	READ PORTx REGISTER
ꟼVOExn:	Pxn PORT VALUE OVERRIDE ENABLE	WRx:	WRITE PORTx
ꟼVOVxn:	Pxn PORT VALUE OVERRIDE VALUE	RPx:	READ PORTx PIN

Note: 1. WRx, WPx, WDx, RRx, RPx, and RDx are common to all pins within the same port. $clk_{I/O}$, SLEEP, and PUD are common to all ports. All other signals are unique for each pin.

The following table summarizes the function of the overriding signals. The pin and port indexes from Figure 14-5 on page 191 are not shown in the succeeding tables. The overriding signals are generated internally in the modules having the alternate function.

Table 14-2. Generic Description of Overriding Signals for Alternate Functions

Signal Name	Full Name	Description
PUOE	Pull-up Override Enable	If this signal is set, the pull-up enable is controlled by the PUOV signal. If this signal is cleared, the pull-up is enabled when {DDxn, PORTxn, PUD} = 0b010.
PUOV	Pull-up Override Value	If PUOE is set, the pull-up is enabled/disabled when PUOV is set/cleared, regardless of the setting of the DDxn, PORTxn, and PUD Register bits.
DDOE	Data Direction Override Enable	If this signal is set, the Output Driver Enable is controlled by the DDOV signal. If this signal is cleared, the Output driver is enabled by the DDxn Register bit.
DDOV	Data Direction Override Value	If DDOE is set, the Output Driver is enabled/disabled when DDOV is set/cleared, regardless of the setting of the DDxn Register bit.
PVOE	Port Value Override Enable	If this signal is set and the Output Driver is enabled, the port value is controlled by the PVOV signal. If PVOE is cleared, and the Output Driver is enabled, the port Value is controlled by the PORTxn Register bit.
PVOV	Port Value Override Value	If PVOE is set, the port value is set to PVOV, regardless of the setting of the PORTxn Register bit.
PTOE	Port Toggle Override Enable	If PTOE is set, the PORTxn Register bit is inverted.
DIEOE	Digital Input Enable Override Enable	If this bit is set, the Digital Input Enable is controlled by the DIEOV signal. If this signal is cleared, the Digital Input Enable is determined by MCU state (Normal mode, sleep mode).
DIEOV	Digital Input Enable Override Value	If DIEOE is set, the Digital Input is enabled/disabled when DIEOV is set/cleared, regardless of the MCU state (Normal mode, sleep mode).
DI	Digital Input	This is the Digital Input to alternate functions. In the figure, the signal is connected to the output of the Schmitt-Trigger but before the synchronizer. Unless the Digital Input is used as a clock source, the module with the alternate function will use its own synchronizer.
AIO	Analog Input/Output	This is the Analog Input/output to/from alternate functions. The signal is connected directly to the pad, and can be used bi-directionally.

The following subsections shortly describe the alternate functions for each port, and relate the overriding signals to the alternate function. Refer to the alternate function description for further details.

14.3.1 Alternate Functions of Port B

The Port B pins with alternate functions are shown in the following table.

Table 14-3. Port B Pins Alternate Functions

Port Pin	Alternate Functions
PB7	OC0A/OC1C/PCINT7 (Output Compare and PWM Output A for Timer/Counter0, Output Compare and PWM Output C for Timer/Counter1 or Pin Change Interrupt 7)
PB6	OC1B/PCINT6 (Output Compare and PWM Output B for Timer/Counter1 or Pin Change Interrupt 6)
PB5	OC1A/PCINT5 (Output Compare and PWM Output A for Timer/Counter1 or Pin Change Interrupt 5)
PB4	OC2A/PCINT4 (Output Compare and PWM Output A for Timer/Counter2 or Pin Change Interrupt 4)
PB3	MISO/PDO/PCINT3 (SPI Bus Master Input/Slave Output, Programming Data Output or Pin Change Interrupt 3)
PB2	MOSI/PDI/PCINT2 (SPI Bus Master Output/Slave Input , Programming Data Input or Pin Change Interrupt 2)
PB1	SCK/PCINT1 (SPI Bus Serial Clock or Pin Change Interrupt 1)
PB0	SS/PCINT0 (SPI Slave Select input or Pin Change Interrupt 0)

The alternate pin configuration is as follows:

- **OC0A/OC1C/PCINT7, Bit 7**

OC0A, Output Compare Match A output: The PB7 pin can serve as an external output for the Timer/Counter0 Output Compare. The pin has to be configured as an output (DDB7 set “one”) to serve this function. The OC0A pin is also the output pin for the PWM mode timer function.

OC1C, Output Compare Match C output: The PB7 pin can serve as an external output for the Timer/Counter1 Output Compare C. The pin has to be configured as an output (DDB7 set (one)) to serve this function. The OC1C pin is also the output pin for the PWM mode timer function.

PCINT7, Pin Change Interrupt source 7: The PB7 pin can serve as an external interrupt source.

- **OC1B/PCINT6, Bit 6**

OC1B, Output Compare Match B output: The PB6 pin can serve as an external output for the Timer/Counter1 Output Compare B. The pin has to be configured as an output (DDB6 set (one)) to serve this function. The OC1B pin is also the output pin for the PWM mode timer function.

PCINT6, Pin Change Interrupt source 6: The PB6 pin can serve as an external interrupt sourceOC1A, Output Compare Match A output: The PB5 pin can serve as an external output for the Timer/Counter1 Output Compare A. The pin has to be configured as an output (DDB5 set (one)) to serve this function. The OC1A pin is also the output pin for the PWM mode timer function.

PCINT5, Pin Change Interrupt source 5: The PB5 pin can serve as an external interrupt source.

- **OC2A/PCINT4, Bit 4**

OC2A, Output Compare Match output: The PB4 pin can serve as an external output for the Timer/Counter2 Output Compare. The pin has to be configured as an output (DDB4 set (one)) to serve this function. The OC2A pin is also the output pin for the PWM mode timer function.

PCINT4, Pin Change Interrupt source 4: The PB4 pin can serve as an external interrupt source.

- **MISO/PDO/PCINT3 – Port B, Bit 3**

MISO: Master Data input, Slave Data output pin for SPI channel. When the SPI is enabled as a master, this pin is configured as an input regardless of the setting of DDB3. When the SPI is enabled as a slave, the data direction of this pin is controlled by DDB3. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB3 bit.

PDO, SPI Serial Programming Data Output. During Serial Program Downloading, this pin is used as data output line (see section "Serial Downloading" on page 477 for details).

PCINT3, Pin Change Interrupt source 3: The PB3 pin can serve as an external interrupt source.

- **MOSI/PDI/PCINT2 – Port B, Bit 2**

MOSI: SPI Master Data output, Slave Data input for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB2. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB2. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB2 bit.

PDI, SPI Serial Programming Data Input. During Serial Program Downloading, this pin is used as data input line (see section "Serial Downloading" on page 477 for details).

PCINT2, Pin Change Interrupt source 2: The PB2 pin can serve as an external interrupt source.

- **SCK/PCINT1 – Port B, Bit 1**

SCK: Master Clock output, Slave Clock input pin for SPI channel. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB1. When the SPI0 is enabled as a master, the data direction of this pin is controlled by DDB1. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB1 bit.

PCINT1, Pin Change Interrupt source 1: The PB1 pin can serve as an external interrupt source.

- **SS/PCINT0 – Port B, Bit 0**

SS: Slave Port Select input. When the SPI is enabled as a slave, this pin is configured as an input regardless of the setting of DDB0. As a slave, the SPI is activated when this pin is driven low. When the SPI is enabled as a master, the data direction of this pin is controlled by DDB0. When the pin is forced to be an input, the pull-up can still be controlled by the PORTB0 bit.

Table 14-4 below and Table 14-5 on page 195 relate the alternate functions of Port B to the overriding signals shown in Figure 14-5 on page 191. SPI MSTR INPUT and SPI SLAVE OUTPUT constitute the MISO signal, while MOSI is divided into SPI MSTR OUTPUT and SPI SLAVE INPUT.

PCINT0, Pin Change Interrupt source 0: The PB0 pin can serve as an external interrupt source.

Table 14-4. Overriding Signals for Alternate Functions in PB7:PB4

Signal Name	PB7/OC0A/OC1C	PB6/OC1B	PB5/OC1A	PB4/OC2A
PUOE	0	0	0	0

Signal Name	PB7/OC0A/OC1C	PB6/OC1B	PB5/OC1A	PB4/OC2A
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	OC0/OC1C ENABLE	OC1B ENABLE	OC1A ENABLE	OC2A ENABLE
PVOV	OC0/OC1C	OC1B	OC1A	OC2A
DIEOE	PCINT7•PCIE0	PCINT6•PCIE0	PCINT5•PCIE0	PCINT4•PCIE0
DIEOV	1	1	1	1
DI	PCINT7 INPUT	PCINT6 INPUT	PCINT5 INPUT	PCINT4 INPUT
AIO	–	–	–	–

Table 14-5. Overriding Signals for Alternate Functions in PB3:PB0

Signal Name	PB3/MISO/PDO	PB2/MOSI/PDI	PB1/SCK	PB0/SS
PUOE	SPE•MSTR	SPE•(~MSTR)	SPE•(~MSTR)	SPE•(~MSTR)
PUOV	PORTB3•(~PUD)	PORTB2•(~PUD)	PORTB1•(~PUD)	PORTB0•(~PUD)
DDOE	SPE•MSTR	SPE•(~MSTR)	SPE•(~MSTR)	SPE•(~MSTR)
DDOV	0	0	0	0
PVOE	SPE•(~MSTR)	SPE•MSTR	SPE•MSTR	0
PVOV	SPI SLAVE OUTPUT	SPI MSTR OUTPUT	SCK OUTPUT	0
DIEOE	PCINT3•PCIE0	PCINT2•PCIE0	PCINT1•PCIE0	PCINT0•PCIE0
DIEOV	1	1	1	1
DI	SPI MSTR INPUT PCINT3 INPUT	SPI SLAVE INPUT PCINT2 INPUT	SCK INPUT PCINT1 INPUT	SPI SS PCINT0 INPUT
AIO	–	–	–	–

14.3.2 Alternate Functions of Port D

The Port D pins with alternate functions are shown in the following table.

Table 14-6. Port D Pins Alternate Functions

Port Pin	Alternate Function
PD7	T0 (Timer/Counter0 Clock Input)
PD6	T1 (Timer/Counter1 Clock Input)
PD5	XCK1 (USART1 External Clock Input/Output)
PD4	ICP1 (Timer/Counter1Input Capture Trigger)
PD3	INT3/TXD1 (External Interrupt3 Input or USART1 Transmit Pin)
PD2	INT2/RXD 1(External Interrupt2 Input or USART1 Receive Pin)
PD1	INT1/SDA (External Interrupt1 Input or TWI Serial Data)
PD0	INT0/SCL (External Interrupt0 Input or TWI Serial Clock)

The alternate pin configuration is as follows:

- **T0 – Port D, Bit 7**

T0, this is Timer/Counter0 counter source.

- **T1 – Port D, Bit 6**

T1, this is Timer/Counter1 counter source.

- **XCK1 – Port D, Bit 5**

XCK1, USART1 External clock: The Data Direction Register (DDD5) controls whether the clock is output (DDD5 set) or input (DDD5 cleared). The XCK1 pin is active only when the USART1 operates in Synchronous mode.

- **ICP1 – Port D, Bit 4**

ICP1 – Input Capture Pin 1: The PD4 pin can act as an input capture pin for Timer/Counter1.

- **INT3/TXD1 – Port D, Bit 3**

INT3, External Interrupt source 3: The PD3 pin can serve as an external interrupt source to the MCU.

TXD1, Transmit Data (Data output pin for the USART1). When the USART1 Transmitter is enabled, this pin is configured as an output regardless of the value of DDD3.

- **INT2/RXD1 – Port D, Bit 2**

INT2, External Interrupt source 2: The PD2 pin can serve as an External Interrupt source to the MCU.

RXD1, Receive Data (Data input pin for the USART1). When the USART1 receiver is enabled this pin is configured as an input regardless of the value of DDD2. When the USART forces this pin to be an input, the pull-up can still be controlled by the PORTD2 bit.

- **INT1/SDA – Port D, Bit 1**

INT1, External Interrupt source 1: The PD1 pin can serve as an external interrupt source to the MCU.

SDA, 2-wire Serial Interface Data: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PD1 is disconnected from the port and becomes the Serial Data I/O pin for the 2-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew rate limitation.

- **INT0/SCL – Port D, Bit 0**

INT0, External Interrupt source 0: The PD0 pin can serve as an external interrupt source to the MCU.

SCL, 2-wire Serial Interface Clock: When the TWEN bit in TWCR is set (one) to enable the 2-wire Serial Interface, pin PD0 is disconnected from the port and becomes the Serial Clock I/O pin for the 2-wire Serial Interface. In this mode, there is a spike filter on the pin to suppress spikes shorter than 50 ns on the input signal, and the pin is driven by an open drain driver with slew-rate limitation.

Table 14-7 below and Table 14-8 on page 197 relates the alternate functions of Port D to the overriding signals shown in Figure 14-5 on page 191.

Table 14-7. Overriding Signals for Alternate Functions PD7:PD4

Signal Name	PD7/T0	PD6/T1	PD5/XCK1	PD4/ICP1
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	XCK1 OUTPUT ENABLE	0

Signal Name	PD7/T0	PD6/T1	PD5/XCK1	PD4/ICP1
DDOV	0	0	1	0
PVOE	0	0	XCK1 OUTPUT ENABLE	0
PVOV	0	0	XCK1 OUTPUT	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	T0 INPUT	T1 INPUT	XCK1 INPUT	ICP1 INPUT
AIO	–	–	–	–

Table 14-8. Overriding Signals for Alternate Functions PD3:PD0

Signal Name	PD3/INT3/TXD1	PD2/INT2/RXD1	PD1/INT1/SDA	PD0/INT0/SCL
PUOE	TXEN1	RXEN1	TWEN	TWEN
PUOV	0	PORTD2&(~PUD)	PORTD1&(~PUD)	PORTD0&(~PUD)
DDOE	TXEN1	RXEN1	TWEN	TWEN
DDOV	1	0	SDA_OUT	SCL_OUT
PVOE	TXEN1	0	TWEN	TWEN
PVOV	TXD1	0	0	0
DIEOE	INT3 ENABLE	INT2 ENABLE	INT1 ENABLE	INT0 ENABLE
DIEOV	1	1	1	1
DI	INT3 INPUT	INT2 INPUT/RXD1	INT1 INPUT	INT0 INPUT
AIO	-	-	SDA INPUT	SCL INPUT

Note: 1. When enabled, the 2-wire Serial Interface enables Slew-Rate controls on the output pins PD0 and PD1. This is not shown in this table. In addition, spike filters are connected between the AIO outputs shown in the port figure and the digital logic of the TWI module.

14.3.3 Alternate Functions of Port E

The Port E pins with alternate functions are shown in the following table.

Table 14-9. Port E Pins Alternate Functions

Port Pin	Alternate Function
PE7	INT7/ICP3/CLK0 (External Interrupt7 Input, Timer/Counter3 Input Capture Trigger or Divided System Clock)
PE6	INT6/T3 (External Interrupt6 Input or Timer/Counter3 Clock Input)
PE5	INT5/OC3C (External Interrupt5 Input or Output Compare and PWM Output C for Timer/Counter3)
PE4	INT4/OC3B (External Interrupt4 Input or Output Compare and PWM Output B for Timer/Counter3)
PE3	AIN1/OC3A (Analog Comparator Negative Input or Output Compare and PWM Output A for Timer/Counter3)
PE2	AIN0/XCK0 (Analog Comparator or Positive Input or USART0 external clock input/output)
PE1	TXD0 (USART0 Transmit Pin)

Port Pin	Alternate Function
PE0	RXD0/PCINT8 (USART0 Receive Pin or Pin Change Interrupt8)

- **INT7/ICP3/CLKO – Port E, Bit 7**

INT7, External Interrupt source 7: The PE7 pin can serve as an external interrupt source.

ICP3, Input Capture Pin 3: The PE7 pin can act as an input capture pin for Timer/Counter3.

CLKO - Divided System Clock: The divided system clock can be output on the PE7 pin. The divided system clock will be output if the CKOUT Fuse is programmed, regardless of the PORTE7 and DDE7 settings. It will also be output during reset.

- **INT6/T3 – Port E, Bit 6**

INT6, External Interrupt source 6: The PE6 pin can serve as an external interrupt source.

T3, this is the Timer/Counter3 counter source.

- **INT5/OC3C – Port E, Bit 5**

INT5, External Interrupt source 5: The PE5 pin can serve as an External Interrupt source.

OC3C, Output Compare Match C output: The PE5 pin can serve as an External output for the Timer/Counter3 Output Compare C. The pin has to be configured as an output (DDE5 set "one") to serve this function. The OC3C pin is also the output pin for the PWM mode timer function.

- **INT4/OC3B – Port E, Bit 4**

INT4, External Interrupt source 4: The PE4 pin can serve as an External Interrupt source.

OC3B, Output Compare Match B output: The PE4 pin can serve as an External output for the Timer/Counter3 Output Compare B. The pin has to be configured as an output (DDE4 set (one)) to serve this function. The OC3B pin is also the output pin for the PWM mode timer function.

- **AIN1/OC3A – Port E, Bit 3**

AIN1 – Analog Comparator Negative input. This pin is directly connected to the negative input of the Analog Comparator.

OC3A, Output Compare Match A output: The PE3 pin can serve as an External output for the Timer/Counter3 Output Compare A. The pin has to be configured as an output (DDE3 set "one") to serve this function. The OC3A pin is also the output pin for the PWM mode timer function.

- **AIN0/XCK0 – Port E, Bit 2**

AIN0 – Analog Comparator Positive input. This pin is directly connected to the positive input of the Analog Comparator.

XCK0, this is the USART0 External clock. The Data Direction Register (DDE2) controls whether the clock is output (DDE2 set) or input (DDE2 cleared). The XCK0 pin is active only when the USART0 operates in Synchronous mode.

- **TXD0 – Port E, Bit 1**

TXD0, this is the USART0 Transmit pin.

- **RXD0/PCINT8 – Port E, Bit 0**

RXD0, USART0 Receive Pin. Receive Data (Data input pin for the USART0). When the

USART0 receiver is enabled this pin is configured as an input regardless of the value of DDRE0. When the USART0 forces this pin to be an input, a logical one in PORTE0 will turn on the internal pull-up.

PCINT8, Pin Change Interrupt source 8: The PE0 pin can serve as an external interrupt source.

Table 14-10 below and Table 14-11 below relates the alternate functions of Port E to the overriding signals shown in Figure 14-5 on page 191.

Table 14-10. Overriding Signals for Alternate Functions PE7:PE4

Signal Name	**PE7/INT7/ICP3**	**PE6/INT6/T3**	**PE5/INT5/OC3C**	**PE4/INT4/OC3B**
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	0	0	0	0
DDOV	0	0	0	0
PVOE	0	0	OC3C ENABLE	OC3B ENABLE
PVOV	0	0	OC3C	OC3B
DIEOE	INT7 ENABLE	INT6 ENABLE	INT5 ENABLE	INT4 ENABLE
DIEOV	1	1	1	1
DI	INT7 INPUT / ICP3 INPUT	INT7 INPUT / T3 INPUT	INT5 INPUT	INT4 INPUT
AIO	–	–	–	–

Table 14-11. Overriding Signals for Alternate Functions PE3:PE0

Signal Name	**PE3/AIN1/OC3A**	**PE2/AIN0/XCK0**	**PE1/TXD0**	**PE0 / RXD0/PCINT8**
PUOE	0	0	TXEN0	RXEN0
PUOV	0	0	0	PORTE0 & (~PUD)
DDOE	0	XCK0 OUTPUT ENABLE	TXEN0	RXEN0
DDOV	0	1	1	0
PVOE	OC3BENABLE	XCK0 OUTPUT ENABLE	TXEN0	0
PVOV	OC3B	XCK0 OUTPUT	TXD0	0
DIEOE	0	0	0	PCINT8 & PCIE1
DIEOV	0	0	0	1
DI	0	XCK0 INPUT	–	RXD0
PE0	0	0	0	PCINT8 INPUT
AIO	AIN1 INPUT	AIN0 INPUT	-	-

14.3.4 Alternate Functions of Port F

The Port F has an alternate function as analog input for the ADC as shown in Table 14-12 on page 200. If some Port F pins are configured as outputs, it is essential that these do not switch when a conversion is in progress. This might corrupt the result of the conversion. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a Reset occurs.

Table 14-12. Port F Pins Alternate Functions

Port Pin	Alternate Function
PF7	ADC7/TDI (ADC input channel 7 or JTAG Test Data Input)
PF6	ADC6/TDO (ADC input channel 6 or JTAG Test Data Output)
PF5	ADC5/TMS (ADC input channel 5 or JTAG Test Mode Select)
PF4	ADC4/TCK (ADC input channel 4 or JTAG Test Clock)
PF3	ADC3/DIG4 (ADC input channel 3 or Radio Transceiver RX/TX Indicator Output)
PF2	ADC2/DIG2 (ADC input channel 2 or Radio Transceiver Antenna Diversity Control Output)
PF1	ADC1 (ADC input channel 1)
PF0	ADC0 (ADC input channel 0)

- **TDI, ADC7 – Port F, Bit 7**

ADC7, Analog to Digital Converter, Channel 7.

TDI, JTAG Test Data In: Serial input data to be shifted in to the Instruction Register or Data Register (scan chains). When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **TDO, ADC6 – Port F, Bit 6**

ADC6, Analog to Digital Converter, Channel 6.

TDO, JTAG Test Data Out: Serial output data from Instruction Register or Data Register. When the JTAG interface is enabled, this pin can not be used as an I/O pin. The TDO pin is tri-stated unless TAP states that shift out data are entered.

- **TMS, ADC5 – Port F, Bit 5**

ADC5, Analog to Digital Converter, Channel 5.

TMS, JTAG Test Mode Select: This pin is used for navigating through the TAP-controller state machine. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **TCK, ADC4 – Port F, Bit 4**

ADC4, Analog to Digital Converter, Channel 4.

TCK, JTAG Test Clock: JTAG operation is synchronous to TCK. When the JTAG interface is enabled, this pin can not be used as an I/O pin.

- **DIG4, ADC3 – Port F, Bit 3**

ADC3, Analog to Digital Converter, Channel 3.

DIG4, Radio Transceiver RX/TX Indicator Output: If the bit PA_EXT_EN in TRX_CTRL_1 is set to one then the PF3 pin serves as the Radio Transceiver receive/transmit indicator output to control an external RF front-end.

- **DIG2, ADC2 – Port F, Bit 2**

ADC2, Analog to Digital Converter, Channel 2.

DIG2, Radio Transceiver Antenna Diversity Control Output: If the bit ANT_EXT_SW_EN in ANT_DIV is set to one then the PF2 pin serves as a Radio Transceiver output to control External Antenna Diversity.

- **ADC1 – ADC0 – Port F, Bit 1:0**

Analog to Digital Converter, Channel 1:0.

Table 14-13. Overriding Signals for Alternate Functions PF7:PF4

Signal Name	PF7/ADC7/TDI	PF6/ADC6/TDO	PF5/ADC5/TMS	PF4/ADC4/TCK
PUOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
PUOV	1	0	1	1
DDOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DDOV	0	SHIFT_IR+SHIFT_DR	0	0
PVOE	0	JTAGEN	0	0
PVOV	0	TDO	0	0
DIEOE	JTAGEN	JTAGEN	JTAGEN	JTAGEN
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	TDI/ADC7 INPUT	ADC6 INPUT	TMS/ADC5 INPUT	TCK/ADC4 INPUT

Table 14-14. Overriding Signals for Alternate Functions PF3:PF0

Signal Name	PF3/ADC3/DIG4	PF2/ADC2/DIG2	PF1/ADC1	PF0/ADC0
PUOE	0	0	0	0
PUOV	0	0	0	0
DDOE	PA_EXT_EN	ANT_EXT_SW_EN	0	0
DDOV	PA_EXT_EN	ANT_EXT_SW_EN	0	0
PVOE	PA_EXT_EN	ANT_EXT_SW_EN	0	0
PVOV	DIG4	DIG2	0	0
DIEOE	0	0	0	0
DIEOV	0	0	0	0
DI	–	–	–	–
AIO	ADC3 INPUT	ADC2 INPUT	ADC1 INPUT	ADC0 INPUT

14.3.5 Alternate Functions of Port G

The Port G alternate pin configuration is as follows:

Table 14-15. Port G Pins Alternate Functions

Port Pin	Alternate Function
PG5	OC0B (Output Compare and PWM Output B for Timer/Counter0)
PG4	TOSC1 (RTC Oscillator Timer/Counter2)
PG3	TOSC2 (RTC Oscillator Timer/Counter2)
PG2	AMR (Automated Meter Reading - Counter Input for Timer/Counter2)
PG1	DIG1 (Radio Transceiver Antenna Diversity Control Output)
PG0	DIG3 (Radio Transceiver RX/TX Indicator Output)

- **OC0B – Port G, Bit 5**

OC0B, Output Compare match B output: The PG5 pin can serve as an external output for the TImer/Counter0 Output Compare. The pin has to be configured as an output (DDG5 set) to serve this function. The OC0B pin is also the output pin for the PWM mode timer function.

- **TOSC1 – Port G, Bit 4**

TOSC2, Timer Oscillator pin 1: Setting the AS2 bit to one and the EXCLKAMR bit to zero in ASSR, enables asynchronous clocking of Timer/Counter2 by a Crystal Oscillator. The pin PG4 is disconnected from the port, and becomes the input of the inverting Oscillator amplifier. In this mode, a Crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

TOSC2 – Port G, Bit 3

TOSC2, Timer Oscillator pin 2: Setting the AS2 bit to one and the EXCLKAMR bit to zero in ASSR, enables asynchronous clocking of Timer/Counter2 by a Crystal Oscillator. The pin PG3 is disconnected from the port, and becomes the inverting output of the Oscillator amplifier. In this mode, a Crystal Oscillator is connected to this pin, and the pin can not be used as an I/O pin.

- **AMR – Port G, Bit 2**

AMR, Automated Meter Reading Input: Setting the AS2 and the EXCLKAMR bits in ASSR to one, enables asynchronous clocking of Timer/Counter2 by the AMR pin

- **DIG1 – Port G, Bit 1**

DIG1, Radio Transceiver Antenna Diversity Control Output: If the bit ANT_EXT_SW_EN in ANT_DIV is set to one then the PG1 pin serves as a Radio Transceiver output to control External Antenna Diversity.

- **DIG3 – Port G, Bit 0**

DIG3, Radio Transceiver RX/TX Indicator Output: If the bit PA_EXT_EN in TRX_CTRL_1 is set to one then the PG0 pin serves as the Radio Transceiver receive/transmit indicator output to control an external RF front-end.

Table 14-16 below relates the alternate functions of Port G to the overriding signals shown in Figure 14-5 on page 191.

Table 14-16. Overriding Signals for Alternate Functions PG5:PG2

Signal Name	PG5/OC0B	PG4/TOSC1	PG3/TOSC2	PG2/AMR
PUOE	–	AS2 & (~EXCLKAMR)	AS2 & (~EXCLKAMR) & (~EXCLK)	AS2 & EXCLKAMR
PUOV	–	0	0	0
DDOE	–	AS2 & (~EXCLKAMR)	AS2 & (~EXCLKAMR) & (~EXCLK)	AS2 & EXCLKAMR
DDOV	–	0	0	0
PVOE	OC0B Enable	0	0	0
PVOV	OC0B	0	0	0
DIEOE	–	AS2 & (~EXCLKAMR)	AS2 & (~EXCLKAMR) & (~EXCLK)	AS2 & EXCLKAMR
DIEOV	–	EXCLK	0	1
DI	–	–	–	AMR
AIO	–	T/C2 OSC INPUT	T/C2 OSC OUTPUT	–

Table 14-17. Overriding Signals for Alternate Functions PG1:PG0

Signal Name	PG1/DIG1	PG0/DIG3
PUOE	0	0
PUOV	0	0
DDOE	ANT_EXT_SW_EN	PA_EXT_EN
DDOV	ANT_EXT_SW_EN	PA_EXT_EN
PVOE	ANT_EXT_SW_EN	PA_EXT_EN
PVOV	DIG1	DIG3
DIEOE	0	0
DIEOV	0	0
DI	–	–
AIO	–	–

14.4 Register Description

14.4.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
$35 ($55)				PUD					MCUCR
Read/Write				RW					
Initial Value				0					

The MCU Control Register contains control bits of the general Microcontroller Unit functions.

- **Bit 4 – PUD - Pull-up Disable**

When this bit is written to one, the I/O ports pull-up resistors are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-up resistor ({DDxn, PORTxn} = 2'b01). See section "Ports as General Digital I/O" for more details about this feature.

14.4.2 DPDS0 – Port Driver Strength Register 0

Bit	7	6	5	4	3	2	1	0	
NA ($136)	PFDRV1	PFDRV0	PEDRV1	PEDRV0	PDDRV1	PDDRV0	PBDRV1	PBDRV0	DPDS0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The output driver strength can be set individually for each digital I/O port. The following tables show output current levels for a typical supply voltage of DEVDD = 3.3V. Refer to section "Electrical Characteristics" for details.

- **Bit 7:6 – PFDRV1:0 - Driver Strength Port F**

Table 14-18 PFDRV Register Bits

Register Bits	Value	Description
PFDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

- **Bit 5:4 – PEDRV1:0 - Driver Strength Port E**

Table 14-19 PEDRV Register Bits

Register Bits	Value	Description
PEDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

- **Bit 3:2 – PDDRV1:0 - Driver Strength Port D**

Table 14-20 PDDRV Register Bits

Register Bits	Value	Description
PDDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

- **Bit 1:0 – PBDRV1:0 - Driver Strength Port B**

Table 14-21 PBDRV Register Bits

Register Bits	Value	Description
PBDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

14.4.3 DPDS1 – Port Driver Strength Register 1

Bit	7	6	5	4	3	2	1	0	
NA ($137)	Res5	Res4	Res3	Res2	Res1	Res0	PGDRV1	PGDRV0	DPDS1
Read/Write	R	R	R	R	R	R	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The output driver strength can be set individually for each digital I/O port. The following table shows output current levels for a typical supply voltage of DEVDD = 3.3V. Refer to section "Electrical Characteristics" for details.

- **Bit 7:2 – Res5:0 - Reserved**
- **Bit 1:0 – PGDRV1:0 - Driver Strength Port G**

Driver strength can be set for port G except the port pins PG3 and PG4. The leakage current of the ports PG3 and PG4 is reduced.

Table 14-22 PGDRV Register Bits

Register Bits	Value	Description
PGDRV1:0	0	2 mA
	1	4 mA
	2	6 mA
	3	8 mA

14.4.4 PORTB – Port B Data Register

Bit	7	6	5	4	3	2	1	0	
$05 ($25)	PORTB7:0								PORTB
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

If PORTBn is written logic one when the PORTB pin n is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTBn has to be written logic zero or the pin has to be configured as an output pin. If PORTBn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTBn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

- **Bit 7:0 – PORTB7:0 - Port B Data Register Value**

14.4.5 DDRB – Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
$04 ($24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DDBn bit in the DDRB Register selects the direction of the PORTB pin n. If DDBn is written logic one, PBn is configured as an output pin. If DDBn is written logic zero, PBn is configured as an input pin.

- **Bit 7:0 – DDB7:0 - Port B Data Direction Register Value**

14.4.6 PINB – Port B Input Pins Address

Bit	7	6	5	4	3	2	1	0	
$03 ($23)	PINB7:0								PINB
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register allows access to the PORTB pins independent of the setting of the Data Direction bit DDBn. The port pin can be read through the PINBn Register bit, and writing a logic one to PINBn toggles the value of PORTBn.

- **Bit 7:0 – PINB7:0 - Port B Input Pins Value**

14.4.7 PORTD – Port D Data Register

Bit	7	6	5	4	3	2	1	0	
$0B ($2B)	PORTD7:0								PORTD
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

If PORTDn is written logic one when the PORTD pin n is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTDn has to be written logic zero or the pin has to be configured as an output pin. If PORTDn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTDn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

- **Bit 7:0 – PORTD7:0 - Port D Data Register Value**

14.4.8 DDRD – Port D Data Direction Register

Bit	7	6	5	4	3	2	1	0	
$0A ($2A)	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0	DDRD
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DDDn bit in the DDRD Register selects the direction of the PORTD pin n. If DDDn is written logic one, PDn is configured as an output pin. If DDDn is written logic zero, PDn is configured as an input pin.

- **Bit 7:0 – DDD7:0 - Port D Data Direction Register Value**

14.4.9 PIND – Port D Input Pins Address

Bit	7	6	5	4	3	2	1	0	
$09 ($29)	PIND7:0								PIND
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register allows access to the PORTD pins independent of the setting of the Data Direction bit DDDn. The port pin can be read through the PINDn Register bit, and writing a logic one to PINDn toggles the value of PORTDn.

- **Bit 7:0 – PIND7:0 - Port D Input Pins Value**

14.4.10 PORTE – Port E Data Register

Bit	7	6	5	4	3	2	1	0	
$0E ($2E)	PORTE7:0								PORTE
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

If PORTEn is written logic one when the PORTE pin n is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTEn has to be written logic zero or the pin has to be configured as an output pin. If PORTEn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTEn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

- **Bit 7:0 – PORTE7:0 - Port E Data Register Value**

14.4.11 DDRE – Port E Data Direction Register

Bit	7	6	5	4	3	2	1	0	
$0D ($2D)	DDE7	DDE6	DDE5	DDE4	DDE3	DDE2	DDE1	DDE0	DDRE
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DDEn bit in the DDRE Register selects the direction of the PORTE pin n. If DDEn is written logic one, PEn is configured as an output pin. If DDEn is written logic zero, PEn is configured as an input pin.

- **Bit 7:0 – DDE7:0 - Port E Data Direction Register Value**

14.4.12 PINE – Port E Input Pins Address

Bit	7	6	5	4	3	2	1	0	
$0C ($2C)	PINE7:0								PINE
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register allows access to the PORTE pins independent of the setting of the Data Direction bit DDEn. The port pin can be read through the PINEn Register bit, and writing a logic one to PINEn toggles the value of PORTEn.

- **Bit 7:0 – PINE7:0 - Port E Input Pins Value**

14.4.13 PORTF – Port F Data Register

Bit	7	6	5	4	3	2	1	0	
$11 ($31)	PORTF7:0								PORTF
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

If PORTFn is written logic one when the PORTF pin n is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTFn has to be written logic zero or the pin has to be configured as an output pin. If PORTFn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTFn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

- **Bit 7:0 – PORTF7:0 - Port F Data Register Value**

14.4.14 DDRF – Port F Data Direction Register

Bit	7	6	5	4	3	2	1	0	
$10 ($30)	DDF7	DDF6	DDF5	DDF4	DDF3	DDF2	DDF1	DDF0	DDRF
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DDFn bit in the DDRF Register selects the direction of the PORTF pin n. If DDFn is written logic one, PFn is configured as an output pin. If DDFn is written logic zero, PFn is configured as an input pin.

- **Bit 7:0 – DDF7:0 - Port F Data Direction Register Value**

14.4.15 PINF – Port F Input Pins Address

Bit	7	6	5	4	3	2	1	0	
$0F ($2F)	PINF7:0								PINF
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register allows access to the PORTF pins independent of the setting of the Data Direction bit DDFn. The port pin can be read through the PINFn Register bit, and writing a logic one to PINFn toggles the value of PORTFn.

- **Bit 7:0 – PINF7:0 - Port F Input Pins Value**

14.4.16 PORTG – Port G Data Register

Bit	7	6	5	4	3	2	1	0	
$14 ($34)	Res1	Res0	PORTG5	PORTG4	PORTG3	PORTG2	PORTG1	PORTG0	PORTG
Read/Write	R	R	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

If PORTGn is written logic one when the PORTG pin n is configured as an input pin, the pull-up resistor is activated. To switch the pull-up resistor off, PORTGn has to be written logic zero or the pin has to be configured as an output pin. If PORTGn is written logic one when the pin is configured as an output pin, the port pin is driven high (one). If PORTGn is written logic zero when the pin is configured as an output pin, the port pin is driven low (zero).

- **Bit 7:6 – Res1:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 5:0 – PORTG5:0 - Port G Data Register Value**

14.4.17 DDRG – Port G Data Direction Register

Bit	7	6	5	4	3	2	1	0	
$13 ($33)	Res1	Res0	DDG5	DDG4	DDG3	DDG2	DDG1	DDG0	DDRG
Read/Write	R	R	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The DDGn bit in the DDRG Register selects the direction of the PORTG pin n. If DDGn is written logic one, PGn is configured as an output pin. If DDGn is written logic zero, PGn is configured as an input pin.

- **Bit 7:6 – Res1:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 5:0 – DDG5:0 - Port G Data Direction Register Value**

14.4.18 PING – Port G Input Pins Address

Bit	7	6	5	4	3	2	1	0	
$12 ($32)	Res1	Res0	PING5	PING4	PING3	PING2	PING1	PING0	PING
Read/Write	R	R	R	R	R	R	R	R	
Initial Value	0	0	0	0	0	0	0	0	

This register allows access to the PORTG pins independent of the setting of the Data Direction bit DDGn. The port pin can be read through the PINGn Register bit, and writing a logic one to PINGn toggles the value of PORTGn.

- **Bit 7:6 – Res1:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 5:0 – PING5:0 - Port G Input Pins Value**

15 Interrupts

This section describes the specifics of the interrupt handling as performed in ATmega128RFA1. For a general explanation of the AVR interrupt handling, refer to "Reset and Interrupt Handling" on page 15.

15.1 Interrupt Vectors in ATmega128RFA1

Table 15-1. Reset and Interrupt Vectors

Vector No.	Program Address[2]	Source	Interrupt Definition
0	$0000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset, Watchdog Reset, and JTAG AVR Reset
1	$0002	INT0	External Interrupt Request 0
2	$0004	INT1	External Interrupt Request 1
3	$0006	INT2	External Interrupt Request 2
4	$0008	INT3	External Interrupt Request 3
5	$000A	INT4	External Interrupt Request 4
6	$000C	INT5	External Interrupt Request 5
7	$000E	INT6	External Interrupt Request 6
8	$0010	INT7	External Interrupt Request 7
9	$0012	PCINT0	Pin Change Interrupt Request 0
10	$0014	PCINT1	Pin Change Interrupt Request 1
11	$0016[3]	PCINT2	Pin Change Interrupt Request 2
12	$0018	WDT	Watchdog Time-out Interrupt
13	$001A	TIMER2_COMPA	Timer/Counter2 Compare Match A
14	$001C	TIMER2_COMPB	Timer/Counter2 Compare Match B
15	$001E	TIMER2_OVF	Timer/Counter2 Overflow
16	$0020	TIMER1_CAPT	Timer/Counter1 Capture Event
17	$0022	TIMER1_COMPA	Timer/Counter1 Compare Match A
18	$0024	TIMER1_COMPB	Timer/Counter1 Compare Match B
19	$0026	TIMER1_COMPC	Timer/Counter1 Compare Match C
20	$0028	TIMER1_OVF	Timer/Counter1 Overflow
21	$002A	TIMER0_COMPA	Timer/Counter0 Compare Match A
22	$002C	TIMER0_COMPB	Timer/Counter0 Compare match B
23	$002E	TIMER0_OVF	Timer/Counter0 Overflow
24	$0030	SPI_STC	SPI Serial Transfer Complete
25	$0032	USART0_RX	USART0 Rx Complete
26	$0034	USART0_UDRE	USART0 Data Register Empty
27	$0036	USART0_TX	USART0 Tx Complete
28	$0038	ANALOG_COMP	Analog Comparator

Vector No.	Program Address[2]	Source	Interrupt Definition
29	$003A	ADC	ADC Conversion Complete
30	$003C	EE_READY	EEPROM Ready
31	$003E	TIMER3_CAPT	Timer/Counter3 Capture Event
32	$0040	TIMER3_COMPA	Timer/Counter3 Compare Match A
33	$0042	TIMER3_COMPB	Timer/Counter3 Compare Match B
34	$0044	TIMER3_COMPC	Timer/Counter3 Compare Match C
35	$0046	TIMER3_OVF	Timer/Counter3 Overflow
36	$0048	USART1_RX	USART1 Rx Complete
37	$004A	USART1_UDRE	USART1 Data Register Empty
38	$004C	USART1_TX	USART1 Tx Complete
39	$004E	TWI	2-wire Serial Interface
40	$0050	SPM_READY	Store Program Memory Ready
41	$0052(3)	TIMER4_CAPT	Timer/Counter4 Capture Event
42	$0054	TIMER4_COMPA	Timer/Counter4 Compare Match A
43	$0056	TIMER4_COMPB	Timer/Counter4 Compare Match B
44	$0058	TIMER4_COMPC	Timer/Counter4 Compare Match C
45	$005A	TIMER4_OVF	Timer/Counter4 Overflow
46	$005C(3)	TIMER5_CAPT	Timer/Counter5 Capture Event
47	$005E	TIMER5_COMPA	Timer/Counter5 Compare Match A
48	$0060	TIMER5_COMPB	Timer/Counter5 Compare Match B
49	$0062	TIMER5_COMPC	Timer/Counter5 Compare Match C
50	$0064	TIMER5_OVF	Timer/Counter5 Overflow
51	$0066(3)	Reserved	
52	$0068(3)	Reserved	
53	$006A(3)	Reserved	
54	$006C(3)	Reserved	
55	$006E(3))	Reserved	
56	$0070(3)	Reserved	
57	$0072	TRX24_PLL_LOCK	Transceiver PLL Lock
58	$0074	TRX24_PLL_UNLOCK	Transceiver PLL Unlock
59	$0076	TRX24_RX_START	Transceiver Receive Start
60	$0078	TRX24_RX_END	Transceiver Receive End
61	$007A	TRX24_CCA_ED_DONE	Transceiver CCAED Meassurement finished
62	$007C	TRX24_XAH_AMI	Transceiver Frame Address Match
63	$007E	TRX24_TX_END	Transceiver Transmit End
64	$0080	TRX24_AWAKE	Transceiver Wakeup finished

Vector No.	Program Address[2]	Source	Interrupt Definition
65	$0082	SCNT_CMP1	Symbol Counter Compare Match 1
66	$0084	SCNT_CMP 2	Symbol Counter Compare Match 2
67	$0086	SCNT_CMP 3	Symbol Counter Compare Match 3
68	$0088	SCNT_OVFL	Symbol Counter Overflow
69	$008A	SCNT_BACKOFF	Symbol Counter Backoff Slot Counter
70	$008C	AES_READY	AES Encryption Ready
71	$008E	BAT_LOW	Batterie Monitor Allert

Note: 1. When the BOOTRST Fuse is programmed, the device will jump to the Boot Loader address at reset, see "Memory Programming" on page 464.

2. When the IVSEL bit in MCUCR is set, Interrupt Vectors will be moved to the start of the Boot Flash Section. The address of each Interrupt Vector will then be the address in this table added to the start address of the Boot Flash Section.

3. Not usefull in ATmega128RFA1 due to limited pin count.

15.2 Reset and Interrupt Vector Placement

Table 15-2 below shows Reset and Interrupt Vectors placement for the various combinations of BOOTRST and IVSEL settings. If the program never enables an interrupt source, the Interrupt Vectors are not used, and regular program code can be placed at these locations. This is also the case if the Reset Vector is in the Application section while the Interrupt Vectors are in the Boot section or vice versa.

Table 15-2. Reset and Interrupt Vectors Placement [(1)]

BOOTRST	IVSEL	Reset Address	Interrupt Vectors Start Address
1	0	0x0000	0x0002
1	1	0x0000	Boot Reset Address + 0x0002
0	0	Boot Reset Address	0x0002
0	1	Boot Reset Address	Boot Reset Address + 0x0002

Note: 1. The Boot Reset Address is shown in Table 30-7 on page 461 through Table 30-6 on page 461. For the BOOTRST Fuse “1” means unprogrammed while “0” means programmed.

The most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega128RFA1 is:

```
Address   Labels   Code                        Comments
 0x0000             jmp     RESET              ;Reset Handler
 0x0002             jmp     INT0               ;IRQ0 Handler
 0x0004             jmp     INT1               ;IRQ1 Handler
 0x0006             jmp     INT2               ;IRQ2 Handler
 0x0008             jmp     INT3               ;IRQ3 Handler
 0x000A             jmp     INT4               ;IRQ4 Handler
 0x000C             jmp     INT5               ;IRQ5 Handler
 0x000E             jmp     INT6               ;IRQ6 Handler
 0x0010             jmp     INT7               ;IRQ7 Handler
 0x0012             jmp     PCINT0             ;PCINT0 Handler
 0x0014             jmp     PCINT1             ;PCINT1 Handler
 0x0016             jmp     PCINT2             ;PCINT2 Handler
```

```
0X0018          jmp    WDT                  ;Watchdog Timeout Handler
0x001A          jmp    TIM2_COMPA           ;Timer2 CompareA Handler
0x001C          jmp    TIM2_COMPB           ;Timer2 CompareB Handler
0x001E          jmp    TIM2_OVF             ;Timer2 Overflow Handler
0x0020          jmp    TIM1_CAPT            ;Timer1 Capture Handler
0x0022          jmp    TIM1_COMPA           ;Timer1 CompareA Handler
0x0024          jmp    TIM1_COMPB           ;Timer1 CompareB Handler
0x0026          jmp    TIM1_COMPC           ;Timer1 CompareC Handler
0x0028          jmp    TIM1_OVF             ;Timer1 Overflow Handler
0x002A          jmp    TIM0_COMPA           ;Timer0 CompareA Handler
0x002C          jmp    TIM0_COMPB           ;Timer0 CompareB Handler
0x002E          jmp    TIM0_OVF             ;Timer0 Overflow Handler
0x0030          jmp    SPI_STC              ;SPI Transfer Complete Handler
0x0032          jmp    USART0_RX            ;USART0 RX Complete Handler
0x0034          jmp    USART0_UDRE          ;USART0,UDR Empty Handler
0x0036          jmp    USART0_TX            ;USART0 TX Complete Handler
0x0038          jmp    ANA_COMP             ;Analog Comparator Handler
0x003A          jmp    ADC                  ;ADC Conversion Complete Handler
0x003C          jmp    EE_RDY               ;EEPROM Ready Handler
0x003E          jmp    TIM3_CAPT            ;Timer3 Capture Handler
0x0040          jmp    TIM3_COMPA           ;Timer3 CompareA Handler
0x0042          jmp    TIM3_COMPB           ;Timer3 CompareB Handler
0x0044          jmp    TIM3_COMPC           ;Timer3 CompareC Handler
0x0046          jmp    TIM3_OVF             ;Timer3 Overflow Handler
0x0048          jmp    USART1_RX            ;USART1 RX Complete Handler
0x004A          jmp    USART1_UDRE          ;USART1,UDR Empty Handler
0x004C          jmp    USART1_TX            ;USART1 TX Complete Handler
0x004E          jmp    TWI                  ;2-wire Serial Handler
0x0050          jmp    SPM_RDY              ;SPM Ready Handler
0x0052          jmp    TIM4_CAPT            ;Timer4 Capture Handler
0x0054          jmp    TIM4_COMPA           ;Timer4 CompareA Handler
0x0056          jmp    TIM4_COMPB           ;Timer4 CompareB Handler
0x0058          jmp    TIM4_COMPC           ;Timer4 CompareC Handler
0x005A          jmp    TIM4_OVF             ;Timer4 Overflow Handler
0x005C          jmp    TIM5_CAPT            ;Timer5 Capture Handler
0x005E          jmp    TIM5_COMPA           ;Timer5 CompareA Handler
0x0060          jmp    TIM5_COMPB           ;Timer5 CompareB Handler
0x0062          jmp    TIM5_COMPC           ;Timer5 CompareC Handler
0x0064          jmp    TIM5_OVF             ;Timer5 Overflow Handler
0x0066          jmp    0x15e                ;0x15e <__bad_interrupt>
0x0068          jmp    0x15e                ;0x15e <__bad_interrupt>
0x006A          jmp    0x15e                ;0x15e <__bad_interrupt>
0x006C          jmp    0x15e                ;0x15e <__bad_interrupt>
0x006E          jmp    0x15e                ;0x15e <__bad_interrupt>
0x0070          jmp    0x15e                ;0x15e <__bad_interrupt>
0x0072          jmp    TRX24_PLL_LOCK       ;Transceiver PLL Lock Handler
0x0074          jmp    TRX24_PLL_UNLOCK     ;Transceiver PLL Unlock Handler
0x0076          jmp    TRX24_RX_START       ;Transceiver RX Start Handler
0x0078          jmp    TRX24_RX_END         ;Transceiver RX End Handler
0x007A          jmp    TRX24_CCA_ED_DONE    ;Transceiver CCAED DONE Handler
0x007C          jmp    TRX24_XAH_AMI        ;Transceiver Addr. Match Handler
0x007E          jmp    TRX24_TX_END         ;Transceiver Transmit End Handler
0x0080          jmp    TRX24_AWAKE          ;Transceiver Wake Up Handler
0x0082          jmp    SCNT_CMP1            ;Symbol Counter Compare Match 1
0x0084          jmp    SCNT_CMP2            ;Symbol Counter Compare Match 2
0x0086          jmp    SCNT_CMP3            ;Symbol Counter Compare Match 3
0x0088          jmp    SCNT_OVFL            ;Symbol Counter Overflow Handler
0x008A          jmp    SCNT_BACKOFF         ;Symbol Backoff Slot Counter H.
```

```
0x008C            jmp    AES_READY         ;Encryption/Decryption Ready H.
0x008E            jmp    BAT_LOW           ;Batterie Monitor Alert Handler
  ;
0x0090   RESET:   ldi    r16, high(RAMEND) ;Main program start
0x0091            out    SPH,r16           ;Set Stack Pointer to top of RAM
0x0092            ldi    r16, low(RAMEND)
0x0093            out    SPL,r16
0x0094            sei                      ;Enable interrupts
0x0095          <instr>  xxx
 ...     ...      ...    ...
```

When the BOOTRST Fuse is unprogrammed, the Boot section size set to 8KBytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address Labels       Code                   Comments
 0x0000  RESET:       ldi r16,high(RAMEND)  ;Main program start
 0x0001               out SPH,r16           ;Set Stack Pointer to top of RAM
 0x0002               ldi r16,low(RAMEND)
 0x0003               out SPL,r16
 0x0004               sei                   ;Enable interrupts
 0x0005               <instr> xxx
.org 0xF002
 0xF002               jmp EXT_INT0          ;IRQ0 Handler
 0xF004               jmp EXT_INT1          ;IRQ1 Handler
 ... ... ... ;
 0xFO70               jmp USART3_TXC        ;USART3 TX Complete Handler
```

When the BOOTRST Fuse is programmed and the Boot section size set to 8KBytes, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address Labels       Code                   Comments
.org 0x0002
 0x0002               jmp EXT_INT0          ;IRQ0 Handler
 0x0004               jmp EXT_INT1          ;IRQ1 Handler
 ... ... ... ;
.org 0xF000
 0xF000  RESET:       ldi r16,high(RAMEND)  ;Main program start
 0xF001               out SPH,r16           ;Set Stack Pointer to top of RAM
 0xF002               ldi r16,low(RAMEND)
 0xF003               out SPL,r16
 0xF004               sei                   ;Enable interrupts
 0xF005               <instr> xxx
```

When the BOOTRST Fuse is programmed, the Boot section size set to 8KBytes and the IVSEL bit in the MCUCR Register is set before any interrupts are enabled, the most typical and general program setup for the Reset and Interrupt Vector Addresses is:

```
Address Labels       Code                   Comments
.org 0xF000
 0xF000               jmp RESET             ;Reset handler
 0xF002               jmp EXT_INT0          ;IRQ0 Handler
 0xF004               jmp EXT_INT1          ;IRQ1 Handler
 ... ... ... ;
 0xF072  RESET:       ldi r16,high(RAMEND)  ; Main program start
```

```
0xF073                 out SPH,r16             ;Set Stack Pointer to top of RAM
0xF074                 ldi r16,low(RAMEND)
0xF075                 out SPL,r16
0xF076                 sei                     ;Enable interrupts
0xFO77                 <instr> xxx
```

15.3 Moving Interrupts Between Application and Boot Section

The MCU Control Register controls the placement of the Interrupt Vector table, see Code Example below. For more details, see "Reset and Interrupt Handling" on page 15.

Assembly Code Example

```
Move_interrupts:
  ; Get MCUCR
  in r16, MCUCR
  mov r17, r16
  ; Enable change of Interrupt Vectors
  ori r16, (1<<IVCE)
  out MCUCR, r16
  ; Move interrupts to Boot Flash section
  ori r16, (1<<IVSEL)
  out MCUCR, r17
  ret
```

C Code Example

```
void Move_interrupts(void)
{
uchar temp;
  /* Get MCUCR */
  temp = MCUCR;
  /* Enable change of Interrupt Vectors */
  MCUCR = temp|(1<<IVCE);
  /* Move interrupts to Boot Flash section */
  MCUCR = temp|(1<<IVSEL);
}
```

15.4 Register Description

15.4.1 MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
$35 ($55)	JTD	Res1	Res0	PUD	Res1	Res0	IVSEL	IVCE	MCUCR
Read/Write	RW	R	R	RW	R	R	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The MCU Control Register contains control bits for general Microcontroller Unit functions.

- **Bit 7 – JTD - JTAG Interface Disable**

When this bit is zero, the JTAG interface is enabled if the JTAGEN Fuse is programmed. If this bit is one, the JTAG interface is disabled. In order to avoid unintentional disabling or enabling of the JTAG interface, a timed sequence must be followed when changing this bit: The application software must write this bit to the desired value twice within four cycles to change its value. Note that this bit must not be altered when using the On-chip Debug system.

- **Bit 6:5 – Res1:0 - Reserved**
- **Bit 4 – PUD - Pull-up Disable**

When this bit is written to one, the I/O ports pull-up resistors are disabled even if the DDxn and PORTxn Registers are configured to enable the pull-up resistor ({DDxn, PORTxn} = 2'b01). See section "Ports as General Digital I/O" for more details about this feature.

- **Bit 3:2 – Res1:0 - Reserved**
- **Bit 1 – IVSEL - Interrupt Vector Select**

When the IVSEL bit is cleared (zero), the Interrupt Vectors are placed at the start of the Flash memory. When this bit is set (one), the Interrupt Vectors are moved to the beginning of the Boot Loader section of the Flash. The actual address of the start of the Boot Flash Section is determined by the BOOTSZ Fuses. Refer to the section "Memory Programming" for details. To avoid unintentional changes of Interrupt Vector tables, a special write procedure must be followed to change the IVSEL bit (see section "Moving Interrupts Between Application and Boot Section" for details): 1. Write the Interrupt Vector Change Enable (IVCE) bit to one; 2. Within four cycles, write the desired value to IVSEL while writing a zero to IVCE. Interrupts will be automatically disabled while this sequence is executed. Interrupts are disabled in the same cycle IVCE is set, and they remain disabled until after the instruction following the write to IVSEL. If IVSEL is not written, interrupts remain disabled for four cycles. The I-bit in the Status Register is unaffected by the automatic disabling. Note that if Interrupt Vectors are placed in the Boot Loader section and Boot Lock bit BLB02 is programmed, interrupts are disabled while executing from the Application section. If Interrupt Vectors are placed in the Application section and Boot Lock bit BLB12 is programed, interrupts are disabled while executing from the Boot Loader section.

- **Bit 0 – IVCE - Interrupt Vector Change Enable**

The IVCE bit must be written to logic one to enable change of the IVSEL bit. IVCE is cleared by hardware four cycles after it is written or when IVSEL is written. Setting the IVCE bit will disable interrupts as explained in the IVSEL description.

16 External Interrupts

The External Interrupts are triggered by the INT7:0 pin or any of the PCINT8:0 pins. Observe that if enabled, the interrupts will trigger even if the INT7:0 or PCINT8:0 pins are configured as outputs. This feature provides a way of generating a software interrupt.

The Pin Change Interrupt PCI0 will trigger if any enabled PCINT7:0 pin toggles, Pin change interrupt PCI1 if the enabled PCINT8 toggles. PCINT23:9 have no function inside the ATmega128RFA1. Their corresponding I/O port are not implemented. PCMSK1 and PCMSK0 Registers control which pins contribute to the pin change interrupts. PCI2 and PCMSK2 associated to PCINT23:16 have no task in this design. Pin change interrupts on PCINT8:0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode.

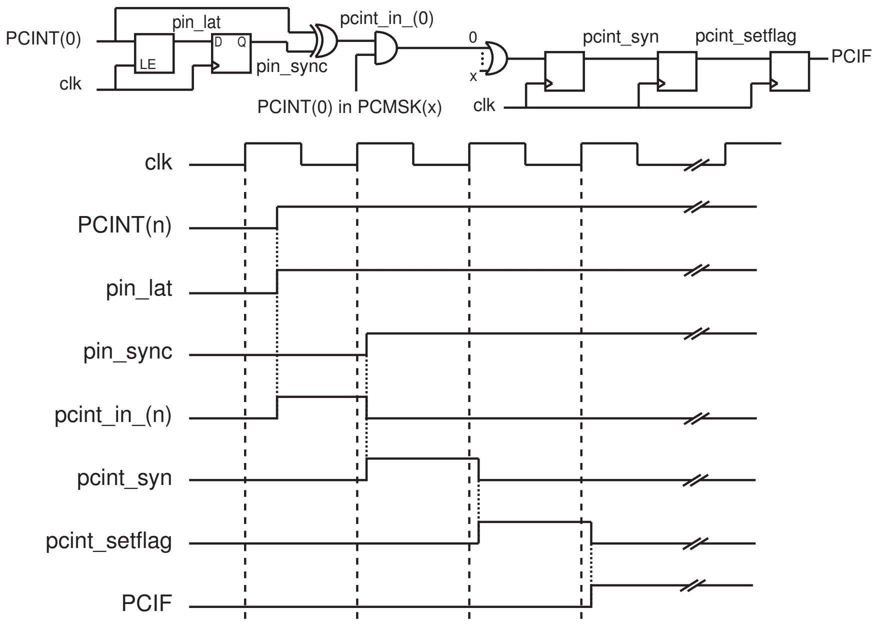
The External Interrupts can be triggered by a falling or rising edge or a low level. This is set up as indicated in the specification for the External Interrupt Control Registers – EICRA (INT3:0) and EICRB (INT7:4). When the external interrupt is enabled and is configured as level triggered, the interrupt will trigger as long as the pin is held low. Note that recognition of falling or rising edge interrupts on INT7:4 requires the presence of an I/O clock, described in "Overview" on page 3. Low level interrupts and the edge interrupt on INT3:0 are detected asynchronously. This implies that these interrupts can be used for waking the part also from sleep modes other than Idle mode. The I/O clock is halted in all sleep modes except Idle mode.

Note that if a level triggered interrupt is used for wake-up from Power-down, the required level must be held long enough for the MCU to complete the wake-up to trigger the level interrupt. If the level disappears before the end of the Start-up Time, the MCU will still wake up, but no interrupt will be generated. The start-up time is defined by the SUT and CKSEL Fuses as described in "Clock Sources" on page 148.

16.1 Pin Change Interrupt Timing

An example of timing of a pin change interrupt is shown in Figure 16-1 below.

Figure 16-1. Normal Pin Change Interrupt



16.2 Register Description

16.2.1 EICRA – External Interrupt Control Register A

Bit	7	6	5	4	3	2	1	0	
NA ($69)	ISC31	ISC30	ISC21	ISC20	ISC11	ISC10	ISC01	ISC00	EICRA
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The External Interrupts 3 - 0 are activated by the external pins INT3:0 if the SREG I-flag and the corresponding interrupt mask in the EIMSK is set. The level and edges on the external pins that activate the interrupts are defined in the following tables. Edges on INT3:0 are registered asynchronously. Pulses on INT3:0 pins wider than the minimum pulse width of typical 50 ns will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt. If enabled, a level triggered interrupt will generate an interrupt request as long as the pin is held low. When changing the ISCn bit, an interrupt can occur. Therefore, it is recommended to first disable INTn by clearing its Interrupt Enable bit in the EIMSK Register. Then, the ISCn bit can be changed. Finally, the INTn interrupt flag should be cleared by writing a logical one to its Interrupt Flag bit (INTFn) in the EIFR Register before the interrupt is re-enabled. When changing the ISCn1/ISCn0 bits, the interrupt must be disabled by clearing its Interrupt Enable bit in the EIMSK Register. Otherwise an interrupt can occur when the bits are changed.

- **Bit 7:6 – ISC31:30 - External Interrupt 3 Sense Control Bit**

Table 16-126 ISC3 Register Bits

Register Bits	Value	Description
ISC31:30	0x00	The low level of INTn generates an interrupt request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

- **Bit 5:4 – ISC21:20 - External Interrupt 2 Sense Control Bit**

Table 16-127 ISC2 Register Bits

Register Bits	Value	Description
ISC21:20	0x00	The low level of INTn generates an interrupt request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

- **Bit 3:2 – ISC11:10 - External Interrupt 1 Sense Control Bit**

Table 16-128 ISC1 Register Bits

Register Bits	Value	Description
ISC11:10	0x00	The low level of INTn generates an interrupt request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

- **Bit 1:0 – ISC01:00 - External Interrupt 0 Sense Control Bit**

Table 16-129 ISC0 Register Bits

Register Bits	Value	Description
ISC01:00	0x00	The low level of INTn generates an interrupt request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

16.2.2 EICRB – External Interrupt Control Register B

Bit	7	6	5	4	3	2	1	0	
NA ($6A)	ISC71	ISC70	ISC61	ISC60	ISC51	ISC50	ISC41	ISC40	EICRB
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

The External Interrupts 7 - 4 are activated by the external pins INT7:4 if the SREG I-flag and the corresponding interrupt mask in the EIMSK is set. The level and edges on the external pins that activate the interrupts are defined in the following tables. Edges on INT7:4 are registered asynchronously. Pulses on INT7:4 pins wider than the minimum pulse width of typical 50 ns will generate an interrupt. Shorter pulses are not guaranteed to generate an interrupt. If low level interrupt is selected, the low level must be held until the completion of the currently executing instruction to generate an interrupt. If enabled, a level triggered interrupt will generate an interrupt request as long as the pin is held low. When changing the ISCn bit, an interrupt can occur. Therefore, it is recommended to first disable INTn by clearing its Interrupt Enable bit in the EIMSK Register. Then, the ISCn bit can be changed. Finally, the INTn interrupt flag should be cleared by writing a logical one to its Interrupt Flag bit (INTFn) in the EIFR Register before the interrupt is re-enabled. When changing the ISCn1/ISCn0 bits, the interrupt must be disabled by clearing its Interrupt Enable bit in the EIMSK Register. Otherwise an interrupt can occur when the bits are changed.

- **Bit 7:6 – ISC71:70 - External Interrupt 7 Sense Control Bit**

Table 16-130 ISC7 Register Bits

Register Bits	Value	Description
ISC71:70	0x00	The low level of INTn generates an interrupt

Register Bits	Value	Description
		request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

- **Bit 5:4 – ISC61:60 - External Interrupt 6 Sense Control Bit**

Table 16-131 ISC6 Register Bits

Register Bits	Value	Description
ISC61:60	0x00	The low level of INTn generates an interrupt request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

- **Bit 3:2 – ISC51:50 - External Interrupt 5 Sense Control Bit**

Table 16-132 ISC5 Register Bits

Register Bits	Value	Description
ISC51:50	0x00	The low level of INTn generates an interrupt request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

- **Bit 1:0 – ISC41:40 - External Interrupt 4 Sense Control Bit**

Table 16-133 ISC4 Register Bits

Register Bits	Value	Description
ISC41:40	0x00	The low level of INTn generates an interrupt request.
	0x01	Any edge of INTn generates asynchronously an interrupt request.
	0x02	The falling edge of INTn generates asynchronously an interrupt request.
	0x03	The rising edge of INTn generates asynchronously an interrupt request.

16.2.3 EIMSK – External Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
$1D ($3D)	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0	EIMSK
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

When an INT7:0 bit 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 External Interrupt Control Registers EICRA and EICRB define 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.

- **Bit 7:0 – INT7:0 - External Interrupt Request Enable**

Table 16-134 INT Register Bits

Register Bits	Value	Description
INT7:0	0x00	All external pin interrupts are disabled.
	0xff	All external pin interrupts are enabled.

16.2.4 EIFR – External Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
$1C ($3C)	INTF7	INTF6	INTF5	INTF4	INTF3	INTF2	INTF1	INTF0	EIFR
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

When an edge or logic change on the INT7:0 pin triggers an interrupt request, INTF7:0 becomes set (one). If the I-bit in SREG and the corresponding interrupt enable bit INT7:0 in EIMSK are set (one), the MCU will jump to the interrupt vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. These flags are always cleared when INT7:0 are configured as level interrupt. Note that when entering sleep mode with the INT3:0 interrupts disabled, the input buffers on these pins will be disabled. This may cause a logic change in internal signals which will set the INTF3:0 flags. See "Digital Input Enable and Sleep Modes" for more information.

- **Bit 7:0 – INTF7:0 - External Interrupt Flag**

Table 16-135 INTF Register Bits

Register Bits	Value	Description
INTF7:0	0x00	No edge or logic change on INT7:0 occurred.
	0x01	A edge or logic change on INT0 occurred and triggered an interrupt request.
	0x02	...
	0x80	A edge or logic change on INT7 occurred and triggered an interrupt request.

16.2.5 PCICR – Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
NA ($68)	Res4	Res3	Res2	Res1	Res0	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:3 – Res4:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 2 – PCIE2 - Pin Change Interrupt Enable 2**

When the PCIE2 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 2 is enabled. Any change on any enabled PCINT23:16 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI2 Interrupt Vector. PCINT23:16 pins are enabled individually by the PCMSK2 Register. Note that the I/O ports corresponding to PCINT23:16 are not implemented. Therefore PCIE2 has no function in this device.

- **Bit 1 – PCIE1 - Pin Change Interrupt Enable 1**

When the PCIE1 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 1 is enabled. Any change on any enabled PCINT15:8 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI1 Interrupt Vector. PCINT15:8 pins are enabled individually by the PCMSK1 Register. Note that the I/O ports corresponding to PCINT15:9 are not implemented.

- **Bit 0 – PCIE0 - Pin Change Interrupt Enable 0**

When the PCIE0 bit is set (one) and the I-bit in the Status Register (SREG) is set (one), pin change interrupt 0 is enabled. Any change on any enabled PCINT7:0 pin will cause an interrupt. The corresponding interrupt of Pin Change Interrupt Request is executed from the PCI0 Interrupt Vector. PCINT7:0 pins are enabled individually by the PCMSK0 Register.

16.2.6 PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
$1B ($3B)	Res4	Res3	Res2	Res1	Res0	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:3 – Res4:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 2 – PCIF2 - Pin Change Interrupt Flag 2**

When a logic change on any PCINT23:16 pin triggers an interrupt request, PCIF2 becomes set (one). If the I-bit in SREG and the PCIE2 bit in PCICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. Note that the I/O ports corresponding to PCINT23:16 are not implemented. Therefore PCIF2 has no function in this device.

- **Bit 1 – PCIF1 - Pin Change Interrupt Flag 1**

When a logic change on any PCINT15:8 pin triggers an interrupt request, PCIF1 becomes set (one). If the I-bit in SREG and the PCIE1 bit in PCICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it. Note that the I/O ports corresponding to PCINT15:9 are not implemented.

- **Bit 0 – PCIF0 - Pin Change Interrupt Flag 0**

When a logic change on any PCINT7:0 pin triggers an interrupt request, PCIF0 becomes set (one). If the I-bit in SREG and the PCIE0 bit in PCICR are set (one), the MCU will jump to the corresponding Interrupt Vector. The flag is cleared when the interrupt routine is executed. Alternatively, the flag can be cleared by writing a logical one to it.

16.2.7 PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	
NA ($6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCMSK2
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($6D)	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

Note that the PCMSK2 register has no function in this device. The I/O ports associated to PCINT23:16 are not implemented. Normally each bit PCINT23:16 selects whether the pin change interrupt is enabled on the corresponding I/O pin. If PCINT23:16 is set and the PCIE2 bit in PCICR is set, the pin change interrupt is enabled on the corresponding I/O pin. If PCINT23:16 is cleared, the pin change interrupt on the corresponding I/O pin is disabled.

- **Bit 7:0 – PCINT23:16 - Pin Change Enable Mask**

16.2.8 PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	
NA ($6C)	PCINT15	PCINT14	PCINT13	PCINT12	PCMSK1
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
NA ($6C)	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

Bit PCINT8 selects whether the pin change interrupt is enabled on the corresponding I/O pin. If PCINT8 is set and the PCIE1 bit in PCICR is set, the pin change interrupt is enabled on the corresponding I/O pin. If PCINT8 is cleared, the pin change interrupt on the corresponding I/O pin is disabled.

- **Bit 7:1 – PCINT15:9 - Pin Change Enable Mask**

Bits 15:9 of the PCMSK1 register have no function in this device. The I/O ports associated to PCINT15:9 are not implemented.

- **Bit 0 – PCINT8 - Pin Change Enable Mask 8**

If this bit is set to one the pin change interrupt on the corresponding I/O pin is enabled. If this bit is set to zero the pin change interrupt is disabled.

16.2.9 PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
NA ($6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
Read/Write	RW	RW	RW	RW	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

Each bit PCINT7:0 selects whether the pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is set and the PCIE0 bit in PCICR is set, the pin change interrupt is enabled on the corresponding I/O pin. If PCINT7:0 is cleared, the pin change interrupt on the corresponding I/O pin is disabled.

- **Bit 7:0 – PCINT7:0 - Pin Change Enable Mask**

17 8-bit Timer/Counter0 with PWM

17.1 Features

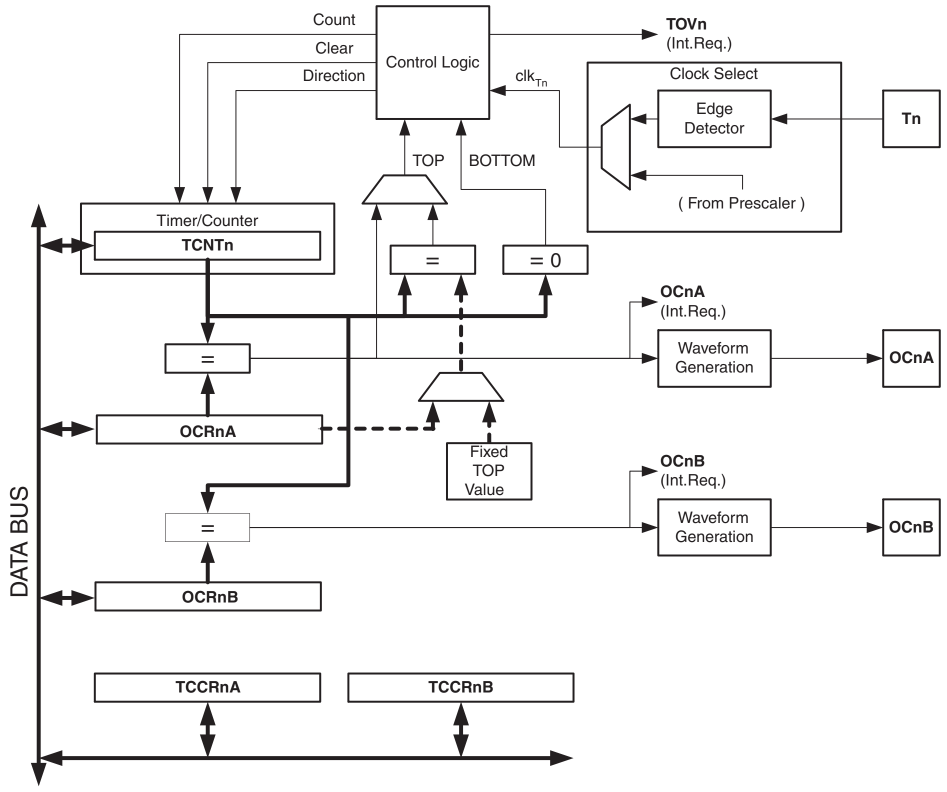
- **Two Independent Output Compare Units**
- **Double Buffered Output Compare Registers**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch Free, Phase Correct Pulse Width Modulator (PWM)**
- **Variable PWM Period**
- **Frequency Generator**
- **Three Independent Interrupt Sources (TOV0, OCF0A, and OCF0B)**

17.2 Overview

Timer/Counter0 is a general purpose 8-bit Timer/Counter module with two independent Output Compare Units and with PWM support. It allows accurate program execution timing (event management) and wave generation.

A simplified block diagram of the 8-bit Timer/Counter is shown in Figure 17-1. For the actual placement of I/O pins refer to section "Pin Configurations" on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins, are shown in bold. The device-specific I/O Register and bit locations are listed in the "Register Description" on page 238.

Figure 17-1. 8-bit Timer/Counter Block Diagram



17.2.1 Registers

The Timer/Counter (TCNT0) and Output Compare Registers (OCR0A and OCR0B) are 8-bit registers. Interrupt request signals (abbreviated to *Int.Req.* in the figure) are all visible in the Timer Interrupt Flag Register (TIFR0). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK0). TIFR0 and TIMSK0 are not shown in the figure.

The Timer/Counter can be clocked internally, via the prescaler or by an external clock source on the T0 pin. The Clock Select logic block controls which clock source and edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the Clock Select logic is referred to as the timer clock (clk_{T0}).

The double buffered Output Compare Registers (OCR0A and OCR0B) are compared with the Timer/Counter value at all times. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pins (OC0A and OC0B); see "Output Compare Unit" on page 228 for details. The Compare Match event will also set the Compare Flag (OCF0A or OCF0B) which can be used to generate an Output Compare interrupt request.

17.2.2 Definitions

Many register and bit references in this section are written in general form. A lower case "*n*" replaces the Timer/Counter number (in this case 0). A lower case "*x*" replaces the Output Compare Unit (in this case Compare Unit A or Compare Unit B). However when using the register or bit defines in a program, the precise form must be used i.e., TCNT0 for accessing Timer/Counter0 counter value and so on.

The definitions in Table 17-1 are also used extensively throughout the document.

Table 17-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0x00.
MAX	The counter reaches its MAXimum when it becomes 0xFF (decimal 255).
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 (MAX) or the value stored in the OCR0A Register. The assignment is dependent on the mode of operation.

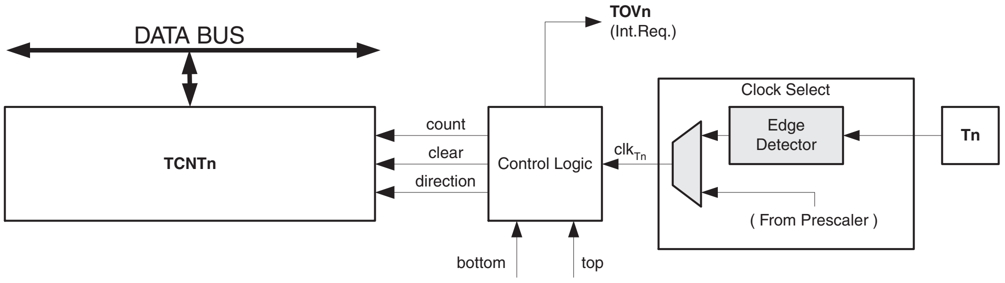
17.3 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the Clock Select (CS02:0) bits located in the Timer/Counter Control Register (TCCR0B). For details on clock sources and prescaler see Timer/Counter 0, 1, 3, 4, and 5 Prescaler on page 304

17.4 Counter Unit

The main part of the 8-bit Timer/Counter is the programmable bi-directional counter unit. Figure 17-2 shows a block diagram of the counter and its surroundings.

Figure 17-2. Counter Unit Block Diagram



Signal description (internal signals):

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

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

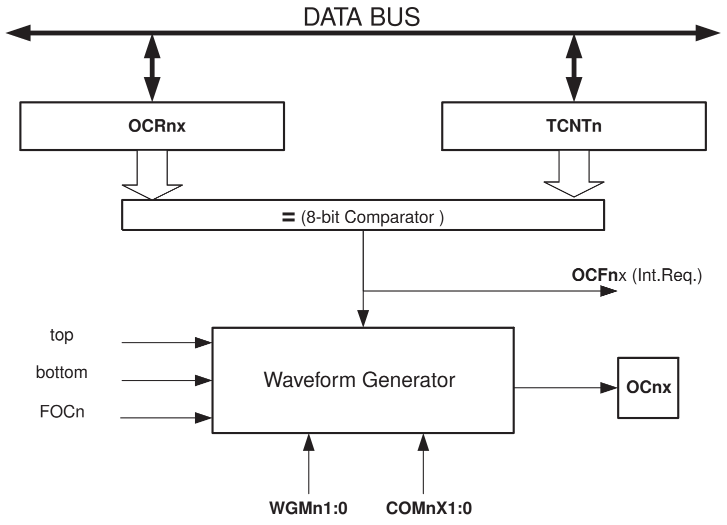
The counting sequence is determined by the setting of the WGM01 and WGM00 bits located in the Timer/Counter Control Register (TCCR0A) and the WGM02 bit located in the Timer/Counter Control Register B (TCCR0B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC0A and OC0B. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 232.

The Timer/Counter Overflow Flag (TOV0) is set according to the mode of operation selected by the WGM02:0 bits. TOV0 can be used for generating a CPU interrupt.

17.5 Output Compare Unit

The 8-bit comparator continuously compares TCNT0 with the Output Compare Registers (OCR0A and OCR0B). The comparator signals a match whenever TCNT0 equals OCR0A or OCR0B. A match will set the Output Compare Flag (OCF0A or OCF0B) at the next clock cycle of the timer. 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. The flag can alternatively be software-cleared by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to the operating mode set by the WGM02:0 bits and Compare Output mode (COM0*x*1:0) bits. The MAX and BOTTOM signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (refer to "Modes of Operation" on page 232).

Figure 17-3. Output Compare Unit, Block Diagram



The OCR0*x* Registers are double buffered when using any of the Pulse Width Modulation (PWM) modes. For the normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR0*x* Compare Registers to either TOP or BOTTOM of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses and thereby making the output glitch-free.

The OCR0*x* Register access may seem complex, but this is not the case. When the double buffering is enabled, the CPU has access to the OCR0*x* Buffer Register. If double buffering is disabled the CPU will access the OCR0*x* directly.

17.5.1 Force Output Compare

In non-PWM waveform generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC0*x*) bit. Forcing Compare Match will not set the OCF0*x* Flag or reload/clear the timer, but the OC0*x* pin will be updated as if a real Compare Match had occurred (the COM0*x*1:0 bits settings define whether the OC0*x* pin is set, cleared or toggled).

17.5.2 Compare Match Blocking by TCNT0 Write

All CPU write operations to the TCNT0 Register will block any Compare Match that occur in the next timer clock cycle, even when the timer is stopped. This feature allows OCR0*x* to be initialized to the same value as TCNT0 without triggering an interrupt when the Timer/Counter clock is enabled.

17.5.3 Using the Output Compare Unit

Since writing TCNT0 in any mode of operation will block all Compare Matches for one timer clock cycle, there are risks involved when changing TCNT0 while using the Output Compare Unit, independently of whether the Timer/Counter is running or not. If the value written to TCNT0 equals the OCR0*x* value, the Compare Match will be missed resulting in an incorrect waveform generation. Similarly, do not write the TCNT0 value equal to BOTTOM when the counter is down-counting.

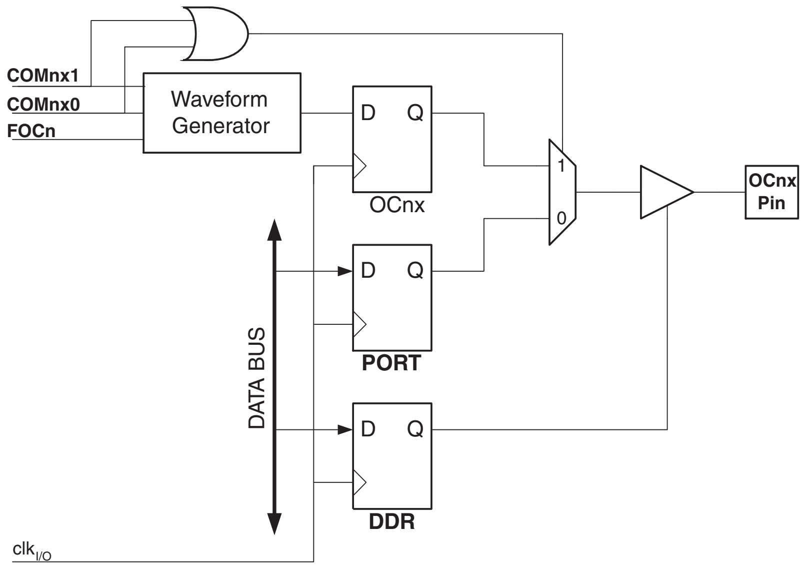
The setup of the OC0*x* should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC0*x* value is to use the Force Output Compare (FOC0*x*) strobe bits in Normal mode. The OC0*x* Registers keep their values even when changing between Waveform Generation modes.

Be aware that the COM0*x*1:0 bits are not double buffered together with the compare value. A Change of the COM0*x*1:0 bits will take effect immediately.

17.6 Compare Match Output Unit

The Compare Output mode (COM0*x*1:0) bits have two functions. The Waveform Generator uses the COM0*x*1:0 bits for defining the Output Compare (OC0*x*) state at the next Compare Match. The COM0x1:0 bits control also the OC0x pin output source. Figure 17-4 shows a simplified schematic of the logic affected by the COM0*x*1:0 bit setting. The I/O Registers, I/O bits and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) affected by the COM0*x*1:0 bits are shown. When referring to the OC0*x* state, the reference is to the internal OC0*x* Register and not to the OC0*x* pin. The OC0*x* Register is reset to “0” if a system reset occurs.

Figure 17-4. Compare Match Output Unit Schematic



The general I/O port function is overridden by the Output Compare (OC0*x*) from the Waveform Generator if either of the COM0*x*1:0 bits are set. However the OC0*x* pin direction (input or output) is still controlled by the Data Direction Register (DDR) of the port pin. The Data Direction Register bit of the OC0*x* pin (DDR_OC0*x*) must be set as output before the OC0*x* value is visible at the pin. The port override function is independent of the Waveform Generation mode.

The design of the Output Compare pin logic allows initializing the OC0*x* state before the output is enabled. Note that some COM0*x*1:0 bit settings are reserved for certain modes of operation (see "Register Description" on page 238).

17.6.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM0*x*1:0 bits differently in Normal, CTC and PWM modes. A setting of COM0*x*1:0 = 0 tells the Waveform Generator in all modes that no action on the OC0*x* Register is to be performed on the next Compare Match. For compare output actions in the non-PWM modes refer to Table 17-2. For fast PWM mode refer to Table 17-3 and for phase correct PWM refer to Table 17-4.

A state change of the COM0*x*1:0 bits will have effect at the first Compare Match after the bits are written. For non-PWM modes the action can be forced to have immediate effect by using the FOC0*x* strobe bits.

The following table shows the COM0*x*1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM).

Table 17-2. Compare Output Mode, non-PWM Mode

COM0A1 COM0B1	COM0A0 COM0B0	Description
0	0	Normal port operation, OC0*x* disconnected;
0	1	Toggle OC0*x* on Compare Match;
1	0	Clear OC0*x* on Compare Match;
1	1	Set OC0*x* on Compare Match;

Table 17-3 shows the COM0*x*1:0 bit functionality when the WGM01:0 bits are set to fast PWM mode.

Table 17-3. Compare Output Mode, Fast PWM Mode

COM0A1 COM0B1	COM0A0 COM0B0	Description
0	0	Normal port operation, OC0*x* disconnected.

0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected.
		WGM02 = 1: Toggle OC0A on Compare Match.
		OC0B: not applicable, reserved function;

1	0	Clear OC0*x* on Compare Match, set OC0*x* at BOTTOM, (non-inverting mode).
1	1	Set OC0*x* on Compare Match, clear OC0*x* at BOTTOM, (inverting mode).

Note: A special case occurs when OCR0*x* equals TOP and COM0*x*1 is set. In this case, the Compare Match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 233.

Table 17-4 shows the COM0*x*1:0 bit functionality when the WGM02:0 bits are set to phase correct PWM mode.

Table 17-4. Compare Output Mode, Phase Correct PWM Mode

COM0A1 COM0B1	COM0A0 COM0B0	Description
0	0	Normal port operation, OC0*x* disconnected.

0	1	WGM02 = 0: Normal Port Operation, OC0A Disconnected.
		WGM02 = 1: Toggle OC0A on Compare Match.
		OC0B: not applicable, reserved function;

1	0	Clear OC0*x* on Compare Match when up-counting. Set OC0*x* on Compare Match when down-counting.
1	1	Set OC0*x* on Compare Match when up-counting. Clear OC0*x* on Compare Match when down-counting.

Note: A special case occurs when OCR0*x* equals TOP and COM0*x*1 is set. In this case, the Compare Match is ignored, but the set or clear is done at TOP. See "Fast PWM Mode" on page 233 for more details.

17.7 Modes of Operation

The mode of operation i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM02:0) and Compare Output mode (COM0*x*1:0) bits. The Compare Output mode bits do not affect the counting sequence while the Waveform Generation mode bits do. The COM0*x*1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM0*x*1:0 bits control whether the output should be set, cleared, or toggled at a Compare Match (see "Output Compare Unit" on page 228).

For detailed timing information see "Timer/Counter Timing Diagrams" on page 236.

Table 17-5 shows the function of the WGM2:0 bits of registers TCCR0A and TCCR0B. These bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used.

Table 17-5. Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRX at	TOV Flag Set on(0,0)
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	TOP	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

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

17.7.1 Normal Mode

The simplest mode of operation is the Normal mode (WGM02:0 = 0). In this mode the counting direction is always up (incrementing) and no counter clear is performed. The counter simply overruns when it passes its maximum 8-bit value (TOP = 0xFF) and then restarts from the BOTTOM (0x00). In normal operation the Timer/Counter Overflow Flag (TOV0) will be set at the same timer clock cycle when the TCNT0 becomes zero. The TOV0 Flag in this case behaves like a 9th bit, except that it is only set and not cleared. However, the timer resolution can be increased by software utilizing the timer overflow interrupt that automatically clears the TOV0 Flag. There are no special cases to consider in the Normal mode. A new counter value can be written at anytime.

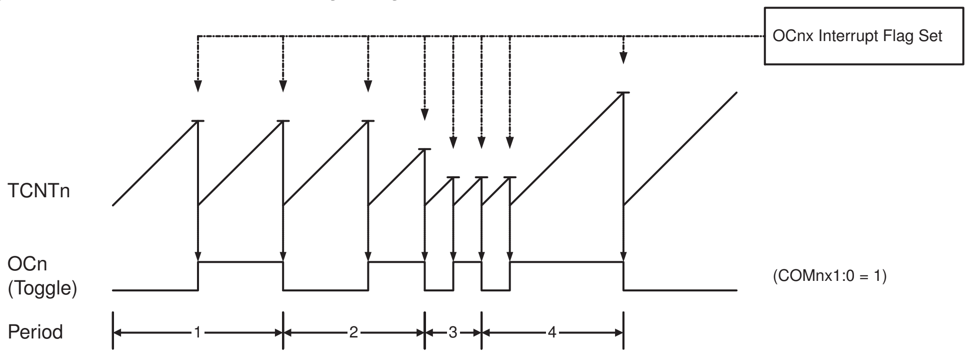
The Output Compare Unit can be used to generate interrupts at some given time. It is not recommended to use the Output Compare for waveform generation in Normal mode, since this will occupy too much CPU time.

17.7.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare (CTC) mode (WGM02:0 = 2), the OCR0A Register is used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT0) matches OCR0A. The OCR0A value defines the TOP value for the counter, hence also its resolution. This mode allows greater control of the Compare Match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in Figure 17-5. The counter value (TCNT0) increases until a Compare Match occurs between TCNT0 and OCR0A. The counter (TCNT0) is then cleared.

Figure 17-5. CTC Mode Timing Diagram



An interrupt can be generated each time the counter value reaches the TOP value by using the OCF0A Flag. If the interrupt is enabled, the interrupt handler routine can update the TOP value. However, changing TOP to a value close to BOTTOM when the counter is running with no or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. If the new value written to OCR0A is lower than the current value of TCNT0, the counter will miss the Compare Match. The counter will then have to count to its maximum value (0xFF) and wrap around starting at 0x00 before the Compare Match can occur.

For generating a waveform output in CTC mode, the OC0A output can be set to toggle its logical level on each Compare Match by setting the Compare Output mode bits to toggle mode (COM0A1:0 = 1). The OC0A value will not be visible on the port pin unless the data direction of the pin is set to output. The generated waveform will have a maximum frequency of $f_{OC0} = f_{clk\_I/O}/2$ when OCR0A is set to zero (0x00). The waveform frequency is defined by the following equation:

$$f_{OC0x} = \frac{f_{CLK\_I/O}}{2 \cdot N \cdot (1 + OCR0x)}$$

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

As for the Normal mode of operation, the TOV0 Flag is set in the same timer clock cycle that the counter changes from MAX to 0x00.

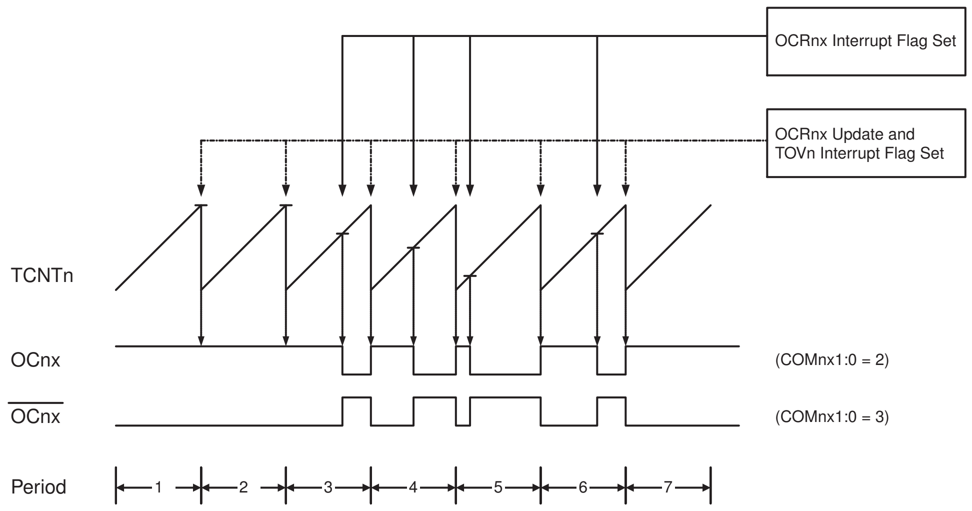
17.7.3 Fast PWM Mode

The fast Pulse Width Modulation (PWM) mode (WGM02:0 = 3 or 7) provides a high frequency PWM waveform generation option. The fast PWM mode differs from the other PWM modes by its single-slope operation. The counter counts from BOTTOM to TOP and then restarts from BOTTOM. TOP is defined as 0xFF when WGM2:0 = 3, and OCR0A when WGM2:0 = 7. In non-inverting Compare Output mode the Output Compare (OC0*x*) is cleared on the Compare Match between TCNT0 and OCR0*x* and set at BOTTOM. In inverting Compare Output mode the output is set on Compare Match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as in the phase correct PWM mode that uses dual-slope operation. This high frequency operation makes the fast

PWM mode well suited for power regulation, rectification and DAC applications. The high frequency allows physically small sized external components (coils, capacitors), and therefore reduces total system cost.

In fast PWM mode, the counter is incremented until the counter value matches the TOP value. The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 17-6. The TCNT0 value is shown in the timing diagram as a histogram illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent Compare Matches between OCR0*x* and TCNT0.

Figure 17-6. Fast PWM Mode Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches TOP. The interrupt handler routine can be used for updating the compare value if the interrupt is enabled.

In fast PWM mode the compare unit allows generating PWM waveforms on the OC0*x* pins. Setting the COM0*x*1:0 bits to 2 will produce a non-inverted PWM. An inverted PWM output can be generated by setting the COM0*x*1:0 to 3. Setting the COM0A1:0 bits to 1 allows the OC0A pin to toggle on Compare Matches if the WGM02 bit is set. This option is not available for the OC0B pin (see Table 17-3 on page 231). The actual OC0*x* value will only be visible at the port pin if the data direction of the port pin is set to output. The PWM waveform is generated by setting (or clearing) the OC0*x* Register at the Compare Match between OCR0*x* and TCNT0, and by clearing (or setting) the OC0*x* Register at the timer clock cycle when the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency for the output $f_{OC0xPWM}$ can be calculated with the following equation:

$$f_{OC0xPWM} = \frac{f_{clk\_I/O}}{N \cdot 256}$$

The N variable represents the pre-scale factor (1, 8, 64, 256 or 1024).

The extreme values for the OCR0A Register represent special cases when generating a PWM waveform output in the fast PWM mode. If the OCR0A is set equal to BOTTOM, the output will be a narrow spike for each MAX+1 timer clock cycle. Setting the OCR0A equal to MAX will result in a constantly high or low output (depending on the polarity of the output set by the COM0A1:0 bits.)

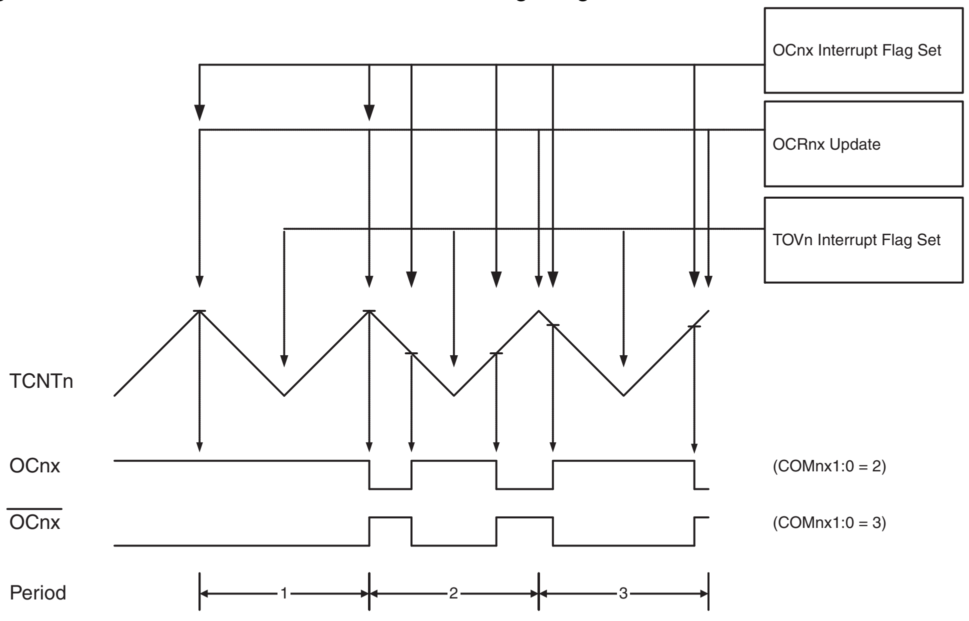
A frequency with 50% duty cycle waveform output in fast PWM mode can be achieved by setting OC0*x* to toggle its logical level on each Compare Match (COM0*x*1:0 = 1). The generated waveform will have a maximum frequency of $f_{OC0xPWM} = f_{clk\_I/O}/2$ when OCR0A is set to zero. This feature is similar to the OC0A toggle in CTC mode, except that in the fast PWM mode the double buffer feature of the Output Compare unit is enabled.

17.7.4 Phase Correct PWM Mode

The phase correct pulse-width modulation (PWM) mode (WGM02:0 = 1 or 5) provides a phase-correct, high-resolution PWM waveform generation option. The phase correct PWM mode is based on a dual-slope operation. The counter counts repeatedly from BOTTOM to TOP and then from TOP to BOTTOM. TOP is defined as 0xFF when WGM2:0 = 1 and TOP = OCR0A when WGM2:0 = 5. In non-inverting Compare Output mode, the Output Compare (OC0x) is cleared on the Compare Match between TCNT0 and OCR0*x* while up-counting, and OC0*x* is set on the Compare Match while down-counting. The operation is inverted in inverting Output Compare mode. The dual-slope operation has a lower maximum operation frequency than single-slope operation. However, due to the symmetric feature of the dual-slope PWM modes, these modes are preferred for motor control applications.

In phase correct PWM mode the counter is incremented until the counter value matches TOP. The counter changes the direction when reaching TOP. The TCNT0 value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown in Figure 17-7 below. The TCNT0 value is shown in the timing diagram as a histogram illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT0 slopes represent Compare Matches between OCR0*x* and TCNT0.

Figure 17-7. Phase Correct PWM Mode Timing Diagram



The Timer/Counter Overflow Flag (TOV0) is set each time the counter reaches BOTTOM. The Interrupt Flag can be used to generate an interrupt each time the counter reaches the BOTTOM value.

In phase correct PWM mode, the compare unit allows generating PWM waveforms on the OC0*x* pins. Setting the COM0*x*1:0 bits to 2 will produce a non-inverted PWM. An inverted PWM output can be generated by setting the COM0*x*1:0 to 3. Setting the COM0A0 bits to 1 allows the OC0A pin to toggle on Compare Matches if the WGM02 bit is set. This option is not available for the OC0B pin (see Table 17-4 on page 231). The actual OC0*x* value will only be visible at the port pin if the data direction for the port pin is set to output. The PWM waveform is generated by clearing (or setting) the OC0*x* Register at the Compare Match between OCR0*x* and TCNT0 when the counter increments, and by setting (or clearing) the OC0*x* Register at Compare Match between OCR0*x* and TCNT0 when the counter decrements. The PWM frequency for the output $f_{OC0xPCPWM}$ when using phase-correct PWM can be calculated with the following equation:

$$f_{OC0xPCPWM} = \frac{f_{clk\_I/O}}{N \cdot 510}$$

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

The extreme values for the OCR0A Register represent special cases when generating a PWM waveform output in the phase-correct PWM mode. If the OCR0A is set equal to BOTTOM, the output will be continuously low and if set equal to MAX the output will be continuously high for non-inverted PWM mode. For inverted PWM the output will have the opposite logic values.

At the very start of period 2 in Figure 17-7 OC*nx* has a transition from high to low even though there is no Compare Match. The reason of this transition is to guarantee symmetry around BOTTOM. There are two cases that give a transition without Compare Match:

- OCR0*x* changes its value from MAX like in Figure 17-7 on page 235. When the OCR0*x* value is MAX the OC0*x* pin value is the same as the result of a down-counting Compare Match. To ensure symmetry around BOTTOM the OC0*x* value at MAX must correspond to the result of an up-counting Compare Match.
- The timer starts counting from a value higher than the one in OCR0*x*. For that reason it misses the Compare Match and hence the OC0*x* change that would have happened on the way up.

17.8 Timer/Counter Timing Diagrams

The Timer/Counter is a synchronous design and the timer clock (clk_{T0}) is therefore shown as a clock enable signal in the following figures. The figures include information on when Interrupt Flags are set. Figure 17-8 contains timing data for basic Timer/Counter operation. The figure shows the count sequence close to the MAX value in all modes other than phase correct PWM mode.

Figure 17-8. Timer/Counter Timing Diagram, no Prescaling

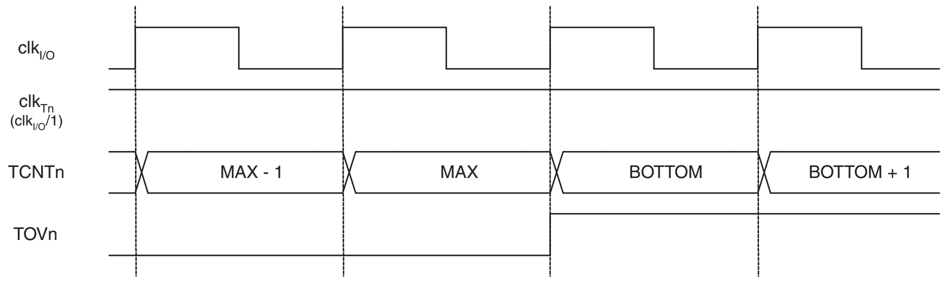


Figure 17-9 shows the same timing data, but with the prescaler enabled.

Figure 17-9. Timer/Counter Timing Diagram with Prescaler ($f_{clk\_I/O}/8$)

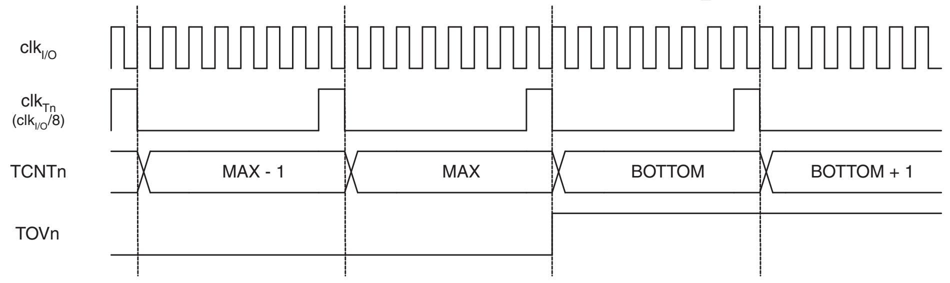


Figure 17-10 shows the setting of OCF0B and OCF0A in all modes except CTC and PWM mode, where OCR0A is TOP.

Figure 17-10. Timer/Counter Timing Diagram, setting of OCF0*x* with Prescaler ($f_{clk\_I/O}/8$)

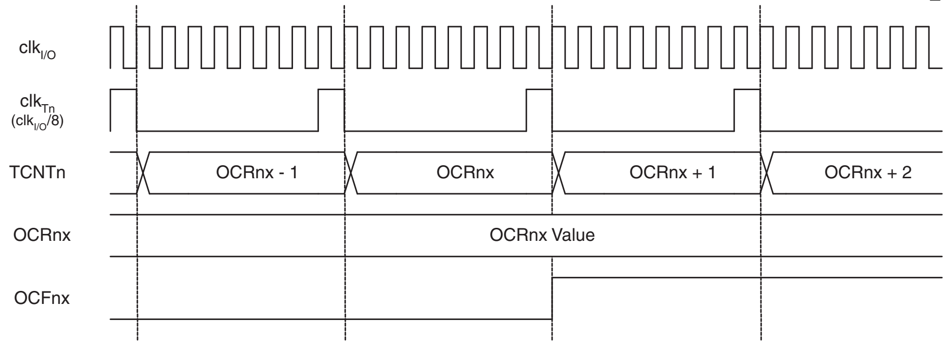
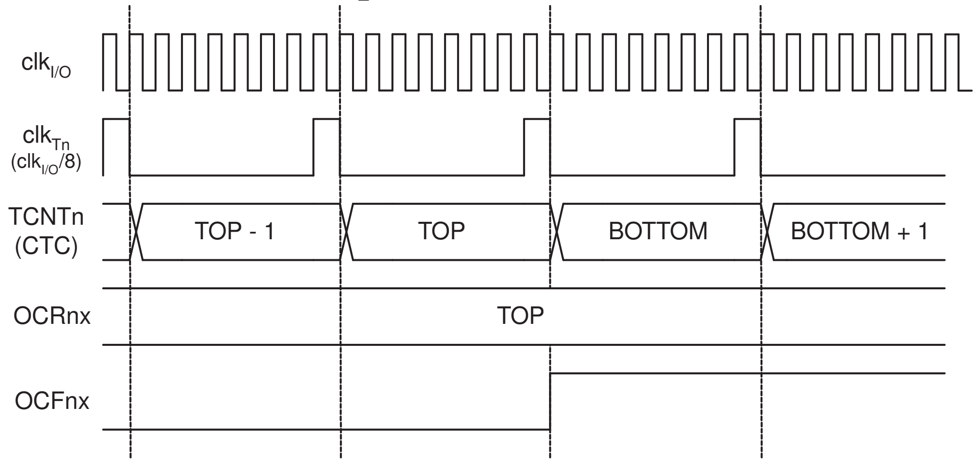


Figure 17-11 shows the setting of OCF0A and the clearing of TCNT0 in CTC mode and fast PWM mode where OCR0A is TOP.

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



17.9 Register Description

17.9.1 GTCCR – General Timer/Counter Control Register

Bit	7	6	5	4	3	2	1	0	
$23 ($43)	TSM	Res4	Res3	Res2	Res1	Res0	PSRASY	PSRSYNC	GTCCR
Read/Write	RW	R	R	R	R	R	R	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – TSM - Timer/Counter Synchronization Mode**

Writing the TSM bit to one activates the Timer/Counter Synchronization mode. In this mode the value that is written to the PSRASY and PSRSYNC bits is kept, hence keeping the corresponding prescaler reset signals asserted. This ensures that the corresponding Timer/Counters are halted and can be configured to the same value without the risk of one of them advancing during the configuration. When the TSM bit is written to zero, the PSRASY and PSRSYNC bits are cleared by hardware and the Timer/Counters simultaneously start counting.

- **Bit 6:2 – Res4:0 - Reserved**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 1 – PSRASY - Prescaler Reset Timer/Counter2**

When this bit is one, the Timer/Counter2 prescaler will be reset. This bit is normally cleared immediately by hardware. If the bit is written when Timer/Counter2 is operating in asynchronous mode, the bit will remain one until the prescaler has been reset. The bit will not be cleared by hardware if the TSM bit is set.

- **Bit 0 – PSRSYNC - Prescaler Reset for Synchronous Timer/Counters**

When this bit is one, the Timer/Counter0, Timer/Counter1, Timer/Counter3, Timer/Counter4 and Timer/Counter5 prescaler will be reset. This bit is normally cleared immediately by hardware, except if the TSM bit is set. Note that Timer/Counter0, Timer/Counter1, Timer/Counter3, Timer/Counter4 and Timer/Counter5 share the same prescaler and a reset of this prescaler will affect all timers.

17.9.2 TCCR0A – Timer/Counter0 Control Register A

Bit	7	6	5	4	3	2	1	0	
$24 ($44)	COM0A1	COM0A0	COM0B1	COM0B0	Res1	Res0	WGM01	WGM00	TCCR0A
Read/Write	RW	RW	RW	RW	R	R	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:6 – COM0A1:0 - Compare Match Output A Mode**

These bits control the Output Compare pin (OC0A) behavior. If one or both of the COM0A1:0 bits are set, the OC0A output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0A pin must be set in order to enable the output driver. When OC0A is connected to the pin, the function of the COM0A1:0 bits depends on the WGM02:0 bit setting. The following shows the COM0A1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM). For the functionality in other modes refer to section "Operating Modes".

Table 17-6 COM0A Register Bits

Register Bits	Value	Description
COM0A1:0	0	Normal port operation, OC0A disconnected
	1	Toggle OC0A on Compare Match
	2	Clear OC0A on Compare Match
	3	Set OC0A on Compare Match

- **Bit 5:4 – COM0B1:0 - Compare Match Output B Mode**

These bits control the Output Compare pin (OC0B) behavior. If one or both of the COM0B1:0 bits are set, the OC0B output overrides the normal port functionality of the I/O pin it is connected to. However, note that the Data Direction Register (DDR) bit corresponding to the OC0B pin must be set in order to enable the output driver. When OC0B is connected to the pin, the function of the COM0B1:0 bits depends on the WGM02:0 bit setting. The following shows the COM0B1:0 bit functionality when the WGM02:0 bits are set to a normal or CTC mode (non-PWM). For the functionality in other modes refer to section "Operating Modes".

Table 17-7 COM0B Register Bits

Register Bits	Value	Description
COM0B1:0	0	Normal port operation, OC0B disconnected
	1	Toggle OC0B on Compare Match
	2	Clear OC0B on Compare Match
	3	Set OC0B on Compare Match

- **Bit 3:2 – Res1:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 1:0 – WGM01:00 - Waveform Generation Mode**

Combined with the WGM02 bit found in the TCCR0B Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used according to the following table. Modes of operation supported by the Timer/Counter0 unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see section "Modes of Operation" for details).

Table 17-8 WGM0 Register Bits

Register Bits	Value	Description
WGM01:00	0x0	Normal mode of operation
	0x1	PWM, phase correct, TOP=0xFF
	0x2	CTC, TOP = OCRA
	0x3	Fast PWM, TOP=0xFF
	0x4	Reserved
	0x5	PWM, Phase correct, TOP = OCRA
	0x6	Reserved
	0x7	Fast PWM, TOP=OCRA

17.9.3 TCCR0B – Timer/Counter0 Control Register B

Bit	7	6	5	4	3	2	1	0	
$25 ($45)	FOC0A	FOC0B	Res1	Res0	WGM02	CS02	CS01	CS00	TCCR0B
Read/Write	W	W	R	R	RW	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7 – FOC0A - Force Output Compare A**

The FOC0A bit is only active when the WGM02:0 bits specify a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written in a PWM operation mode. When writing a logical one to the FOC0A bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0A output is changed according to its COM0A1:0 bits setting. Note that the FOC0A bit is implemented as a strobe. Therefore it is the value present in the COM0A1:0 bits that determines the effect of the forced compare. A FOC0A strobe will not generate any interrupt nor will it clear the timer in CTC mode using OCR0A as TOP. The FOC0A bit is always read as zero.

- **Bit 6 – FOC0B - Force Output Compare B**

The FOC0B bit is only active when the WGM02:0 bits specify a non-PWM mode. However, for ensuring compatibility with future devices, this bit must be set to zero when TCCR0B is written in a PWM operation mode. When writing a logical one to the FOC0B bit, an immediate Compare Match is forced on the Waveform Generation unit. The OC0B output is changed according to its COM0B1:0 bits setting. Note that the FOC0B bit is implemented as a strobe. Therefore it is the value present in the COM0B1:0 bits that determines the effect of the forced compare. A FOC0B strobe will not generate any interrupt nor will it clear the timer in CTC mode using OCR0B as TOP. The FOC0B bit is always read as zero.

- **Bit 5:4 – Res1:0 - Reserved Bit**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 3 – WGM02 -**

Combined with the WGM01:0 bit found in the TCCR0A Register, these bits control the counting sequence of the counter, the source for maximum (TOP) counter value, and what type of waveform generation to be used. Modes of operation supported by the Timer/Counter unit are: Normal mode (counter), Clear Timer on Compare Match (CTC) mode, and two types of Pulse Width Modulation (PWM) modes (see section "Modes of Operation").

- **Bit 2:0 – CS02:00 - Clock Select**

The three Clock Select bits select the clock source to be used by the Timer/Counter0 according to the following table.If external pin modes are used for Timer/Counter0, transitions on the T0 pin will clock the counter even if the pin is configured as an output. This feature allows software control of the counting.

Table 17-9 CS0 Register Bits

Register Bits	Value	Description
CS02:00	0x00	No clock source (Timer/Counter0 stopped)
	0x01	clk_IO/1 (no prescaling)
	0x02	clk_IO/8 (from prescaler)
	0x03	clk_IO/64 (from prescaler)

Register Bits	Value	Description
	0x04	clk_IO/256 (from prescaler)
	0x05	clk_IO/1024 (from prescaler)
	0x06	External clock source on T0 pin, clock on falling edge
	0x07	External clock source on T0 pin, clock on rising edge

17.9.4 TCNT0 – Timer/Counter0 Register

Bit	7	6	5	4	
$26 ($46)	TCNT0_7	TCNT0_6	TCNT0_5	TCNT0_4	TCNT0
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
$26 ($46)	TCNT0_3	TCNT0_2	TCNT0_1	TCNT0_0	TCNT0
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

The Timer/Counter Register gives direct access, both for read and write operations, to the Timer/Counter0 unit 8-bit counter. Writing to the TCNT0 Register blocks (removes) the Compare Match on the following timer clock. Modifying the counter (TCNT0) while the counter is running, introduces a risk of missing a Compare Match between TCNT0 and the OCR0x Registers.

- **Bit 7:0 – TCNT0_7:0 - Timer/Counter0 Byte**

17.9.5 OCR0A – Timer/Counter0 Output Compare Register

Bit	7	6	5	4	
$27 ($47)	OCR0A_7	OCR0A_6	OCR0A_5	OCR0A_4	OCR0A
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
$27 ($47)	OCR0A_3	OCR0A_2	OCR0A_1	OCR0A_0	OCR0A
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

The Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0A pin.

- **Bit 7:0 – OCR0A_7:0 - Output Compare Register**

17.9.6 OCR0B – Timer/Counter0 Output Compare Register B

Bit	7	6	5	4	
$28 ($48)	OCR0B_7	OCR0B_6	OCR0B_5	OCR0B_4	OCR0B
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	
Bit	3	2	1	0	
$28 ($48)	OCR0B_3	OCR0B_2	OCR0B_1	OCR0B_0	OCR0B
Read/Write	RW	RW	RW	RW	
Initial Value	0	0	0	0	

The Output Compare Register B contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0B pin.

- **Bit 7:0 – OCR0B_7:0 - Output Compare Register**

17.9.7 TIMSK0 – Timer/Counter0 Interrupt Mask Register

Bit	7	6	5	4	3	2	1	0	
NA ($6E)	Res4	Res3	Res2	Res1	Res0	OCIE0B	OCIE0A	TOIE0	TIMSK0
Read/Write	R	R	R	R	R	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:3 – Res4:0 - Reserved**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 2 – OCIE0B - Timer/Counter0 Output Compare Match B Interrupt Enable**

When the OCIE0B bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Compare Match B interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0B bit is set in the Timer/Counter0 Interrupt Flag Register TIFR0.

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

When the OCIE0A bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Compare Match A interrupt is enabled. The corresponding interrupt is executed if a Compare Match in Timer/Counter0 occurs, i.e., when the OCF0A bit is set in the Timer/Counter0 Interrupt Flag Register TIFR0.

- **Bit 0 – TOIE0 - Timer/Counter0 Overflow Interrupt Enable**

When the TOIE0 bit is written to one, and the I-bit in the Status Register is set, the Timer/Counter0 Overflow interrupt is enabled. The corresponding interrupt is executed if an overflow in Timer/Counter0 occurs i.e., when the TOV0 bit is set in the Timer/Counter0 Interrupt Flag Register TIFR0.

17.9.8 TIFR0 – Timer/Counter0 Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
$15 ($35)	Res4	Res3	Res2	Res1	Res0	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	RW	RW	RW	
Initial Value	0	0	0	0	0	0	0	0	

- **Bit 7:3 – Res4:0 - Reserved**

This bit is reserved for future use. A read access always will return zero. A write access does not modify the content.

- **Bit 2 – OCF0B - Timer/Counter0 Output Compare B Match Flag**

The OCF0B bit is set when a Compare Match occurs between the Timer/Counter0 and the data in OCR0B Output Compare Register. OCF0B is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0B is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0B (Timer/Counter Compare B Match Interrupt Enable) and OCF0B are set, the Timer/Counter Compare Match Interrupt is executed.

- **Bit 1 – OCF0A - Timer/Counter0 Output Compare A Match Flag**

The OCF0A bit is set when a Compare Match occurs between the Timer/Counter0 and the data in OCR0A Output Compare Register. OCF0A is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, OCF0A is cleared by writing a logic one to the flag. When the I-bit in SREG, OCIE0A (Timer/Counter Compare A Match Interrupt Enable) and OCF0A are set, the Timer/Counter Compare Match Interrupt is executed.

- **Bit 0 – TOV0 - Timer/Counter0 Overflow Flag**

The bit TOV0 is set when an overflow occurs in Timer/Counter0. TOV0 is cleared by hardware when executing the corresponding interrupt handling vector. Alternatively, TOV0 is cleared by writing a logic one to the flag. When the SREG I-bit, TOIE0 (Timer/Counter0 Overflow Interrupt Enable) and TOV0 are set, the Timer/Counter0 Overflow interrupt is executed. The setting of this flag is dependent of the WGM02:0 bit setting.

18 16-bit Timer/Counter (Timer/Counter 1, 3, 4, and 5)

18.1 Features

- **True 16-bit Design (i.e., allows 16-bit PWM)**
- **Three independent Output Compare Units**
- **Double Buffered Output Compare Registers**
- **One Input Capture Unit**
- **Input Capture Noise Canceller**
- **Clear Timer on Compare Match (Auto Reload)**
- **Glitch-free, Phase Correct Pulse Width Modulator (PWM)**
- **Variable PWM Period**
- **Frequency Generator**
- **External Event Counter**
- **Numerous independent interrupt sources**
    - **TOV1, OCF1A, OCF1B, OCF1C, ICF1**
    - **TOV3, OCF3A, OCF3B, OCF3C, ICF3**
    - **TOV4, OCF4A, OCF4B, OCF4C**
    - **TOV5, OCF5A, OCF5B, OCF5C**

18.2 Overview

The 16-bit Timer/Counter unit allows accurate program execution timing (event management), wave generation and signal timing measurement.

Most register and bit references in this section are written in general form. A lower case "*n*" replaces the Timer/Counter number, and a lower case "*x*" replaces the Output Compare unit channel. However when using the register or bit defines in a program, the precise form must be used i.e., TCNT1 for accessing Timer/Counter1 counter value and so on.

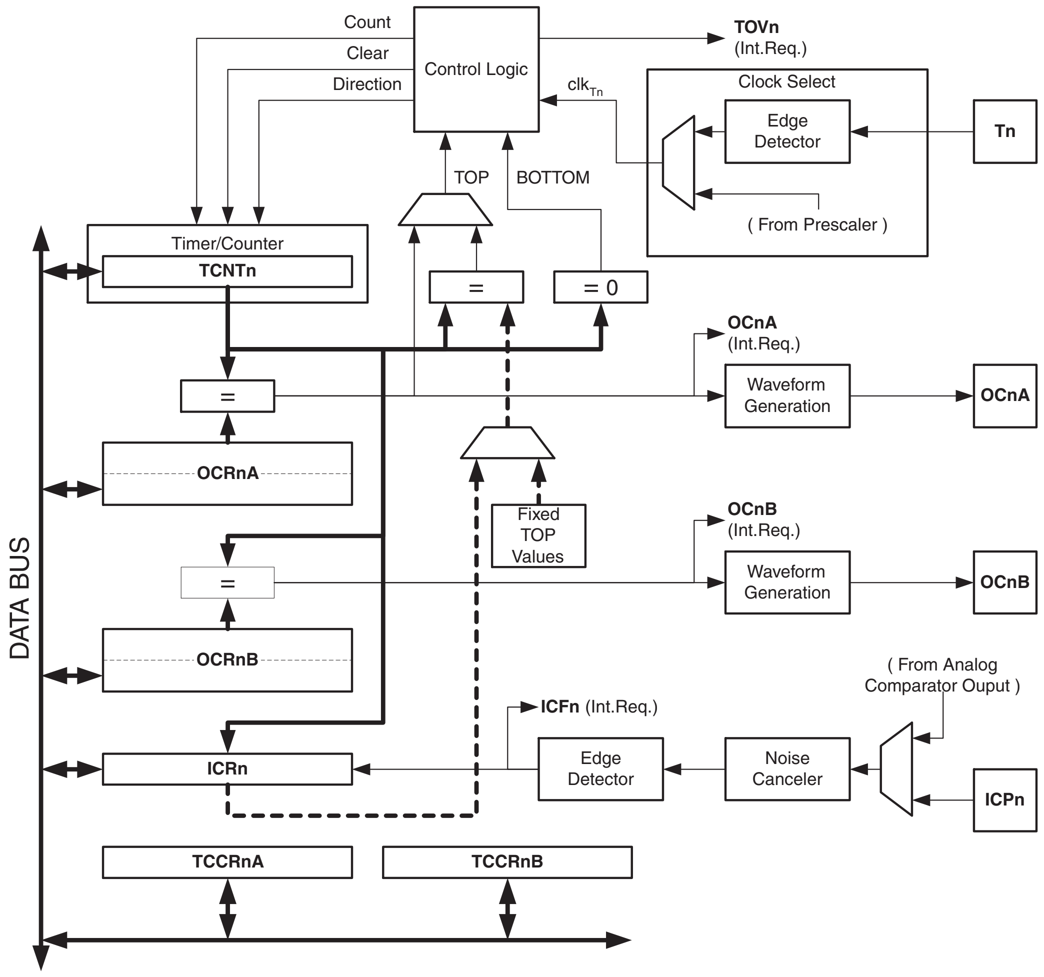
A simplified block diagram of the 16-bit Timer/Counter is shown in Figure 18-1. For the actual placement of I/O pins, see section "Pin Configurations" on page 2. CPU accessible I/O Registers, including I/O bits and I/O pins are shown in bold. The device-specific I/O Register and bit locations are listed in the section "Register Description" on page 266.

The Power Reduction Timer/Counter1 bit, PRTIM1, in "PRR0 – Power Reduction Register0" on page 167 must be written to zero to enable Timer/Counter1 module.

The Power Reduction bits of Timer/Counter3 (PRTIM3), Timer/Counter4 bit (PRTIM4) and Timer/Counter5 (PRTIM5) in "PRR1 – Power Reduction Register 1" on page 168 must be written to zero to enable the respective Timer/Counter module.

Note, note the complete Timer/Counter I/O functionality is provided for each Timer/Counter module depending on the available I/O pins.

Figure 18-1. 16-bit Timer/Counter Block Diagram[1]



Notes: 1. Refer to Figure 1-1 on page 2, Table 14-3 on page 193 and Table 14-9 on page 197 for Timer/Counter1, 2 and 3 pin placements and description.

18.2.1 Registers

The Timer/Counter (TCNT*n*) Output Compare Registers (OCR*n*A/B/C) and Input Capture Register (ICR*n*) are all 16-bit registers. Special procedures must be followed when accessing the 16-bit registers. These procedures are described in the section "Accessing 16-bit Registers" on page 246. The Timer/Counter Control Registers (TCCR*n*A/B/C) are 8-bit registers and have no CPU access restrictions. Interrupt requests (shorten as Int.Req.) signals are all visible in the Timer Interrupt Flag Register (TIFR*n*). All interrupts are individually masked with the Timer Interrupt Mask Register (TIMSK*n*). TIFR*n* and TIMSK*n* are not shown in the figure since these registers are shared by other timer units.

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 which clock edge the Timer/Counter uses to increment (or decrement) its value. The Timer/Counter is inactive when no clock source is selected. The output from the clock select logic is referred to as the timer clock (clk$_{Tn}$).

The double buffered Output Compare Registers (OCR*n*A/B/C) are compared with the

Timer/Counter value at all time. The result of the compare can be used by the Waveform Generator to generate a PWM or variable frequency output on the Output Compare pin (OC*n*A/B/C). See section "Output Compare Units" on page 252 for details. The compare match event will also set the Compare Match Flag (OCF*n*A/B/C) which can be used to generate an Output Compare interrupt request.

The Input Capture Register can capture the Timer/Counter value at a given external (edge triggered) event on either the Input Capture pin (ICP*n*) or on the Analog Comparator pins (see "AC – Analog Comparator" on page 407). The Input Capture unit includes a digital filtering unit (Noise Canceller) for reducing the chance of capturing noise spikes.

The TOP value, or maximum Timer/Counter value, can in some modes of operation be defined by either the OCR*n*A Register, the ICR*n* Register or by a set of fixed values. When using OCR*n*A as TOP value in a PWM mode, the OCR*n*A Register can not be used for generating a PWM output. However the TOP value will in this case be double buffered allowing the TOP value to be changed at run time. If a fixed TOP value is required, the ICR*n* Register can be used as an alternative, freeing the OCR*n*A to be used as PWM output.

18.2.2 Definitions

The following definitions are used extensively throughout the document:

Table 18-1. Definitions

BOTTOM	The counter reaches the BOTTOM when it becomes 0x0000.
MAX	The counter reaches its MAXimum when it becomes 0xFFFF (decimal 65535).
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 one of the fixed values: 0x00FF, 0x01FF, 0x03FF or to the value stored in the OCR*n*A or ICR*n* Register. The assignment is dependent of the mode of operation.

18.3 Accessing 16-bit Registers

The TCNT*n*, OCR*n*A/B/C and ICR*n* are 16-bit registers that can be accessed by the AVR CPU via the 8-bit data bus. The 16-bit register must be byte accessed using two read or write operations. Each 16-bit timer has a single 8-bit register for temporary storing of the high byte of the 16-bit access. The same Temporary Register is shared between all 16-bit registers within each 16-bit timer. Accessing the low byte triggers the 16-bit read or write operation. When the low byte of a 16-bit register is written by the CPU, the written low byte and the high byte stored in the Temporary Register are both copied into the 16-bit register in the same clock cycle. When the low byte of a 16-bit register is read by the CPU, the high byte of the 16-bit register is copied into the Temporary Register in the same clock cycle as the low byte is read.

Not all 16-bit accesses use the Temporary Register for the high byte. Reading the OCR*n*A/B/C 16-bit registers does not involve using the Temporary Register.

To do a 16-bit write, the high byte must be written before the low byte. For a 16-bit read, the low byte must be read before the high byte.

The following code examples show how to access the 16-bit timer registers assuming that no interrupt updates the temporary register. The same principle can be used directly for accessing the OCR*n*A/B/C and ICR*n* Registers. Note that when using the C-programming language, the compiler handles the 16-bit access.

Assembly Code Examples[1]

```
...
; Set TCNTn to 0x01FF
ldi r17,0x01
ldi r16,0xFF
out TCNTnH,r17
out TCNTnL,r16
; Read TCNTn into r17:r16
in r16,TCNTnL
in r17,TCNTnH
...
```

C Code Examples[1]

```
unsigned int i;
...
/* Set TCNTn to 0x01FF */
TCNTn = 0x1FF;
/* Read TCNTn into i */
i = TCNTn;
...
```

Notes: 1. See "About Code Examples" on page 7.

The assembly code example returns the TCNT*n* value in the r17:r16 register pair.

It is important to notice that accessing 16-bit registers are atomic operations. If an interrupt occurs between the two instructions accessing the 16-bit register and the interrupt code updates the temporary register by accessing the same or any other of the 16-bit Timer Registers, then the result of the access outside the interrupt will be corrupted. Therefore the main code must disable the interrupts during the 16-bit access when both the main code and the interrupt code update the temporary register.

The following code examples show how to do an atomic read of the TCNT*n* Register contents. Reading any of the OCR*n*A/B/C or ICR*n* Registers can be done by using the same principle.

The assembly code example returns the TCNT*n* value in the r17:r16 register pair.

Assembly Code Examples[1]

```
TIM16_ReadTCNTn:
  ; Save global interrupt flag
  in r18,SREG
  ; Disable interrupts
  cli
  ; Read TCNTn into r17:r16
  in r16,TCNTnL
  in r17,TCNTnH
  ; Restore global interrupt flag
  out SREG,r18
  ret
```

C Code Examples[1]

```
unsigned int TIM16_ReadTCNTn( void )
{
  unsigned char sreg;
  unsigned int i;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  __disable_interrupt();
  /* Read TCNTn into i */
  i = TCNTn;
  /* Restore global interrupt flag */
  SREG = sreg;
  return i;
}
```

Notes: 1. See "About Code Examples" on page 7 .

The following code examples show how to do an atomic write of the TCNT*n* Register contents. Writing any of the OCR*n*A/B/C or ICR*n* Registers can be done by using the same principle.

The assembly code example requires that the r17:r16 register pair contains the value to be written to TCNT*n*.

Assembly Code Examples[1]

```
TIM16_WriteTCNTn:
  ; Save global interrupt flag
  in r18,SREG
  ; Disable interrupts
  cli
  ; Set TCNTn to r17:r16
  out TCNTnH,r17
  out TCNTnL,r16
  ; Restore global interrupt flag
  out SREG,r18
  ret
```

C Code Examples[1]

```
void TIM16_WriteTCNTn( unsigned int i )
{
  unsigned char sreg;
  unsigned int i;
  /* Save global interrupt flag */
  sreg = SREG;
  /* Disable interrupts */
  __disable_interrupt();
  /* Set TCNTn to i */
  TCNTn = i;
  /* Restore global interrupt flag */
  SREG = sreg;
}
```

Notes: 1. See "About Code Examples" on page 7 .

18.3.1 Reusing the Temporary High Byte Register

If writing to more than one 16-bit register where the high byte is the same for all registers written, then the high byte only needs to be written once. However note that the same rule of atomic operation described previously also applies in this case.

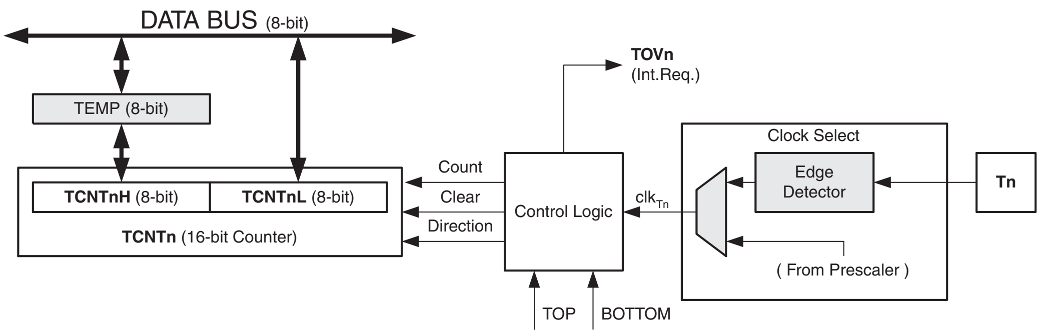
18.4 Timer/Counter Clock Sources

The Timer/Counter can be clocked by an internal or an external clock source. The clock source is selected by the Clock Select logic which is controlled by the Clock Select (CS*n*2:0) bits located in the Timer/Counter control Register B (TCCR*n*B). For details on clock sources and prescaler, see "Timer/Counter 0, 1, 3, 4, and 5 Prescaler" on page 304.

18.5 Counter Unit

The main part of the 16-bit Timer/Counter is the programmable 16-bit bi-directional counter unit. The following figure shows a block diagram of the counter and its surroundings.

Figure 18-2. Counter Unit Block Diagram



Signal description (internal signals):

Count	Increment or decrement TCNT*n* by 1;
Direction	Select between increment and decrement;

Clear	Clear TCNT*n* (set all bits to zero);
clk$_{Tn}$	Timer/Counter clock;
TOP	Signalize that TCNT*n* has reached maximum value;
BOTTOM	Signalize that TCNT*n* has reached minimum value (zero);

The 16-bit counter is mapped into two 8-bit I/O memory locations: Counter High (TCNT*n*H) contains the upper eight bits of the counter and Counter Low (TCNT*n*L) contains the lower eight bits. The TCNT*n*H Register can only be indirectly accessed by the CPU. When the CPU does an access to the TCNT*n*H I/O location, the CPU accesses the high byte temporary register (TEMP). The temporary register is updated with the TCNT*n*H value when the TCNT*n*L is read and TCNT*n*H is updated with the temporary register value when TCNT*n*L is written. This allows the CPU to read or write the entire 16-bit counter value within one clock cycle via the 8-bit data bus. It is important to notice that there are special cases of writing to the TCNT*n* Register giving unpredictable results when the counter is running. These special cases are described in the sections of their importance.

Depending on the mode of operation, the counter is cleared, incremented or decremented at each timer clock (clk$_{Tn}$). The clk$_{Tn}$ can be generated from an external or internal clock source selected by the Clock Select bits (CS*n*2:0). The timer is stopped when no clock source is selected (CS*n*2:0 = 0). However, the TCNT*n* value can be accessed by the CPU independent of whether clk$_{Tn}$ is present or not. A CPU write overrides (has priority over) all counter clear or count operations.

The counting sequence is determined by the settings of the Waveform Generation mode bits (WGM*n*3:0) located in the Timer/Counter Control Registers A and B (TCCR*n*A and TCCR*n*B). There are close connections between how the counter behaves (counts) and how waveforms are generated on the Output Compare outputs OC*n*x. For more details about advanced counting sequences and waveform generation, see "Modes of Operation" on page 256.

The Timer/Counter Overflow Flag (TOV*n*) is set according to the mode of operation selected by the WGM*n*3:0 bits. TOV*n* can be used for generating a CPU interrupt.

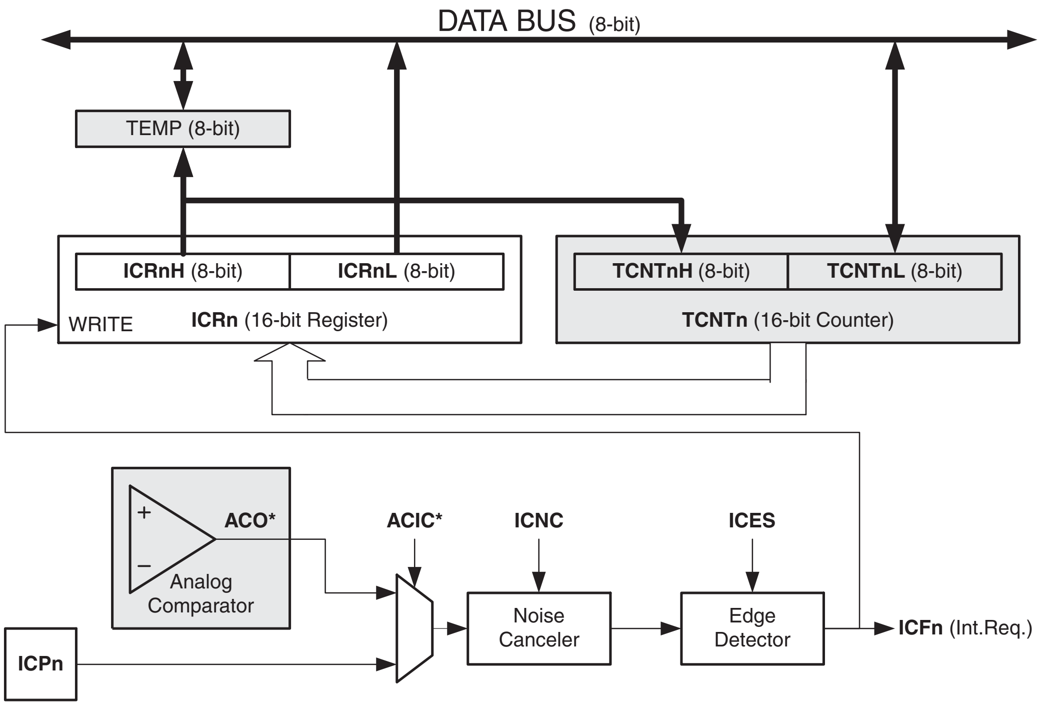
18.6 Input Capture Unit

The Timer/Counter incorporates an input capture unit that can capture external events and give them a time-stamp indicating time of occurrence. The external signal indicating an event, or multiple events, can be applied via the ICP*n* pin or alternatively, for the Timer/Counter1 only, via the Analog Comparator unit. The time-stamps can then be used to calculate frequency, duty-cycle and other features of the signal applied. Alternatively the time-stamps can be used for creating a log of the events.

The Input Capture unit is illustrated by the block diagram shown in Figure 18-3. The elements of the block diagram not direct parts of the input capture unit are gray shaded. The small “n” in register and bit names indicates the Timer/Counter number.

A capture will be triggered when a change of the logic level (an event) occurs on the Input Capture Pin (ICP*n*), or alternatively on the analog Comparator output (ACO), and this change matches the setting of the edge detector. When a capture is triggered, the 16-bit value of the counter (TCNT*n*) is written to the Input Capture Register (ICR*n*). The Input Capture Flag (ICF*n*) is set at the same system clock as the TCNT*n* value is copied into ICR*n* Register. If enabled (TICIE*n* = 1), the input capture flag generates an input capture interrupt. The ICF*n* flag is automatically cleared when the interrupt is executed. Alternatively the ICF*n* flag can be software-cleared by writing a logical one to its I/O bit location.

Figure 18-3. Input Capture Unit Block Diagram



Note: 1. The Analog Comparator Output (ACO) can only trigger the Timer/Counter1 ICP – not Timer/Counter3, 4 or 5.

Reading the 16-bit value in the Input Capture Register (ICR*n*) is done by first reading the low byte (ICR*n*L) and then the high byte (ICR*n*H). When the low byte is read the high byte is copied into the high byte Temporary Register (TEMP). The CPU will access the TEMP Register when reading the ICR*n*H I/O location.

The ICR*n* Register can only be written when using a Waveform Generation mode that utilizes the ICR*n* Register for defining the counter's TOP value. In these cases the Waveform Generation mode (WGMn3:0) bits must be set before the TOP value can be written to the ICR*n* Register. When writing the ICR*n* Register the high byte must be written to the ICR*n*H I/O location before the low byte is written to ICR*n*L.

For more information on how to access the 16-bit registers refer to "Accessing 16-bit Registers" on page 246.

18.6.1 Input Capture Trigger Source

The main trigger source for the input capture unit is the Input Capture Pin (ICP*n*). Timer/Counter1 can alternatively use the analog comparator output as trigger source for the input capture unit. The Analog Comparator is selected as trigger source by setting the analog Comparator Input Capture (ACIC) bit in the Analog Comparator Control and Status Register (ACSR). Be aware that changing trigger source can trigger a capture. The input capture flag must therefore be cleared after the change.

Both the Input Capture Pin (ICP*n*) and the Analog Comparator output (ACO) inputs are sampled using the same technique as for the T*n* pin (Figure 19-1 on page 304). The edge detector is also identical. However, when the noise canceller is enabled, additional logic is inserted before the edge detector increasing the delay by four system clock cycles. Note that the input of the noise canceller and edge detector is always enabled unless the Timer/Counter is set in a Waveform Generation mode that uses ICR*n* to define TOP.

An input capture can be software-triggered by controlling the port of the ICP*n* pin.

18.6.2 Noise Canceller

The noise canceller improves noise immunity by using a simple digital filtering scheme. The noise canceller 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 canceller is enabled by setting the Input Capture Noise Canceller (ICNC*n*) bit in Timer/Counter Control Register B (TCCR*n*B). When enabled the noise canceller introduces additional four system clock cycles of delay from a change applied to the input to the update of the ICR*n* Register. The noise canceller uses the system clock and is therefore not affected by the prescaler.

18.6.3 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. The ICR*n* will be overwritten with a new value if the processor has not read the captured value in the ICR*n* Register before the next event occurs. In this case the result of the capture will be incorrect.

When using the Input Capture interrupt, the ICR*n* Register should be read as early in the interrupt handler routine as possible. Even though the Input Capture interrupt has relatively high priority, the maximum interrupt response time is dependent on the maximum number of clock cycles it takes to handle any of the other interrupt requests.

It is not recommended to use the Input Capture unit in any mode of operation where the TOP value (resolution) is actively changed while counting.

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

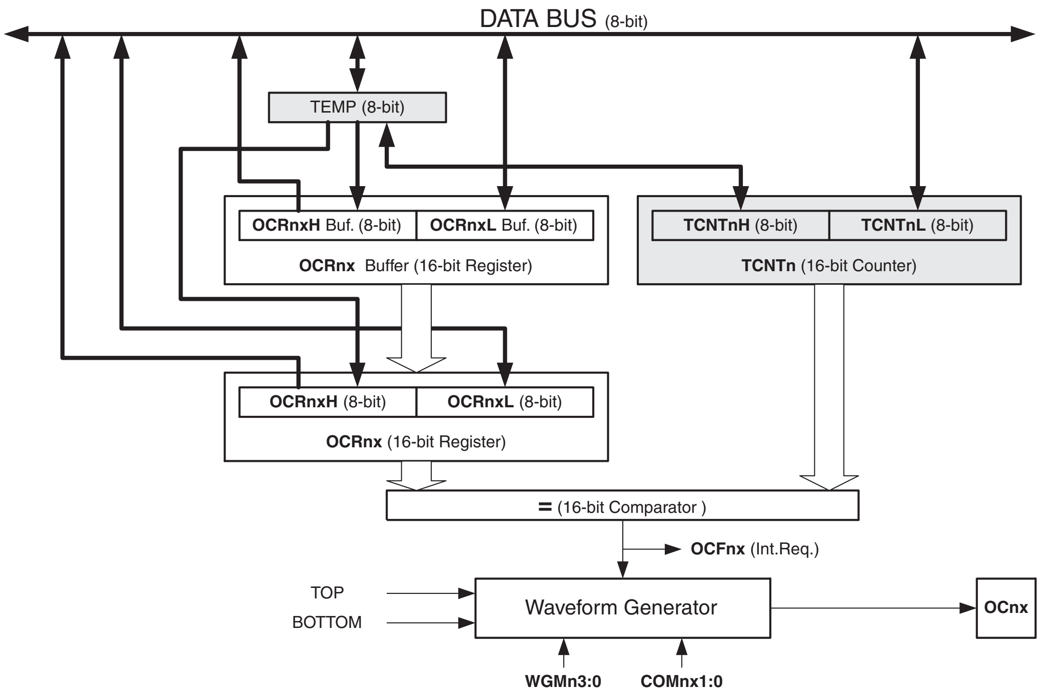
18.7 Output Compare Units

The 16-bit comparator continuously compares TCNT*n* with the Output Compare Register (OCR*nx*). If TCNTn equals OCR*nx* the comparator signals a match. A match will set the Output Compare Flag (OCF*nx*) at the next clock cycle of the timer. If enabled (OCIE*nx* = 1), the Output Compare Flag generates an Output Compare interrupt. The OCF*nx* Flag is automatically cleared when the interrupt is executed. Alternatively the OCF*nx* Flag can be software-cleared by writing a logical one to its I/O bit location. The Waveform Generator uses the match signal to generate an output according to the Waveform Generation mode bits (WGM*n*3:0) and Compare Output mode bits (COM*nx*1:0). The TOP and BOTTOM signals are used by the Waveform Generator for handling the special cases of the extreme values in some modes of operation (see "Modes of Operation" on page 256).

A special feature of Output Compare unit A allows it to define the Timer/Counter TOP value i.e., the counter resolution. In addition to the counter resolution, the TOP value defines the period time for waveforms generated by the Waveform Generator.

Figure 18-4 shows a block diagram of the Output Compare unit. The small "*n*" in the register and bit names indicates the device number (*n* = Timer/Counter *n*), and the "*x*" indicates Output Compare unit A, B or C. The elements of the block diagram not direct parts of the Output Compare unit are gray shaded.

Figure 18-4. Output Compare Unit Block Diagram



The OCR*nx* Register is double buffered when using any of the twelve Pulse Width Modulation (PWM) modes. For the Normal and Clear Timer on Compare (CTC) modes of operation, the double buffering is disabled. The double buffering synchronizes the update of the OCR*nx* Compare Register to either TOP or BOTTOM of the counting sequence. The synchronization prevents the occurrence of odd-length, non-symmetrical PWM pulses, thereby making the output glitch-free.

The OCR*nx* Register access may seem complex, but this is not the case. When the double buffering is enabled, the CPU has access to the OCR*nx* Buffer Register. If double buffering is disabled the CPU will access the OCR*nx* directly. The content of the OCR1*x* (Buffer or Compare) Register is only changed by a write operation (the Timer/Counter does not update this register automatically as the TCNT1 and ICR1 Register). Therefore OCR1*x* is not read via the high byte temporary register (TEMP). However, it is a good practice to read the low byte first similar to accessing other 16-bit registers. Writing the OCR*nx* Registers must be done via the TEMP Register since the compare of all 16 bits is done continuously. The high byte (OCR*nx*H) has to be written first. The TEMP Register will be updated with the value written by the CPU to the high byte I/O location. Then when the low byte (OCR*nx*L) is written to the lower eight bits, the high byte will be copied into the upper 8-bits of either the OCR*nx* buffer or OCR*nx* Compare Register in the same system clock cycle.

For more information of how to access the 16-bit registers refer to "Accessing 16-bit Registers" on page 246.

18.7.1 Force Output Compare

In non-PWM Waveform Generation modes, the match output of the comparator can be forced by writing a one to the Force Output Compare (FOC*nx*) bit. Forcing compare match will not set the OCF*nx* Flag or reload/clear the timer, but the OC*nx* pin will be updated as if a real compare match had occurred (the COM*n*1:0 bits settings define whether the OC*nx* pin is set, cleared or toggled).

18.7.2 Compare Match Blocking by TCNTn Write

All CPU writes to the TCNT*n* Register will block any compare match that occurs in the next clock cycle of the timer even when the timer is stopped. This feature allows OCR*nx* to be initialized to the same value as TCNT*n* without triggering an interrupt when the Timer/Counter clock is enabled.

18.7.3 Using the Output Compare Unit

Since writing TCNT*n* in any mode of operation will block all compare matches for one timer clock cycle, there are risks involved when changing TCNT*n* using any of the Output Compare channels, independent of whether the Timer/Counter is running or not. If the value written to TCNT*n* equals the OCR*nx* value, the compare match will be missed resulting in incorrect waveform generation. Do not write the TCNT*n* equal to TOP in PWM modes with variable TOP values. The compare match for the TOP will be ignored and the counter will continue to 0xFFFF. Similarly, do not write the TCNT*n* value equal to BOTTOM when the counter is down-counting.

The setup of the OC*nx* should be performed before setting the Data Direction Register for the port pin to output. The easiest way of setting the OC*nx* value is to use the Force Output Compare (FOC*nx*) strobe bits in Normal mode. The OC*nx* Register keeps its value even when changing between Waveform Generation modes.

Be aware that the COM*nx*1:0 bits are not double buffered together with the compare value. A change of the COM*nx*1:0 bits will immediately take effect.

18.8 Compare Match Output Unit

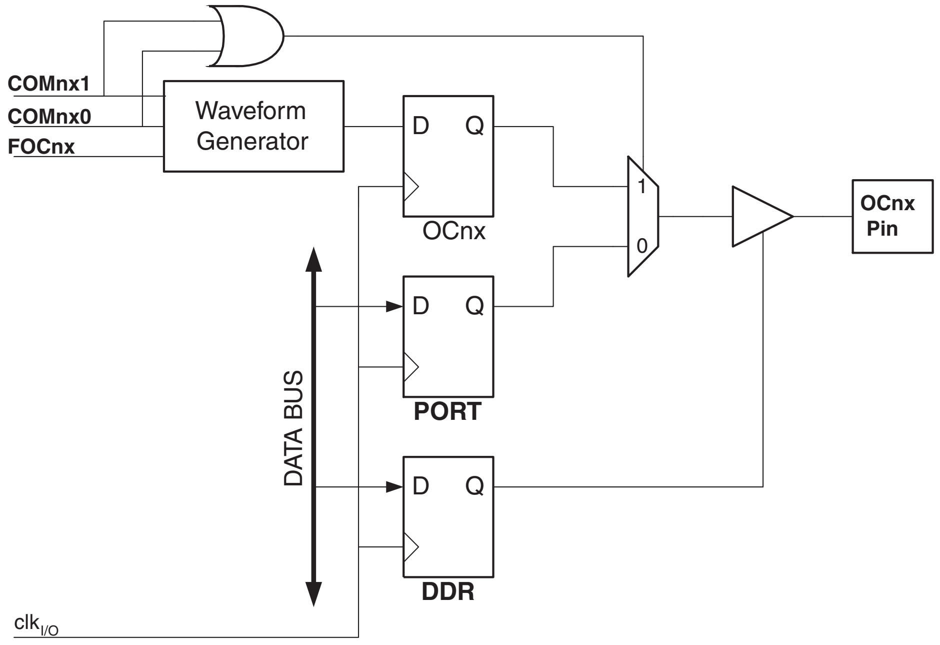
The Compare Output mode (COM*nx*1:0) bits have two functions. The Waveform Generator uses the COM*nx*1:0 bits for defining the Output Compare (OC*nx*) state at the next compare match. Secondly the COM*nx*1:0 bits control the OC*nx* pin output source. Figure 18-5 shows a simplified schematic of the logic affected by the COM*nx*1:0 bit setting. The I/O Registers, I/O bits and I/O pins in the figure are shown in bold. Only the parts of the general I/O Port Control Registers (DDR and PORT) that are affected by the COM*nx*1:0 bits are shown. When referring to the OC*nx* state, the reference is to the internal OC*nx* Register and not to the OC*nx* pin. After a system reset the OC*nx* Register will have a value of “0”.

The general I/O port function is overridden by the Output Compare (OC*nx*) from the Waveform Generator if either of the COM*nx*1:0 bits are set. However, the OC*nx* pin direction (input or output) is still controlled by the Data Direction Register (DDR) for the port pin. The Data Direction Register bit for the OC*nx* pin (DDR_OC*nx*) must be set as output before the OC*nx* value is visible on the pin. The port override function is generally independent of the Waveform Generation mode, but there are some exceptions. Refer to Table 18-2, Table 18-3 and Table 18-4 on page 256 for details.

The design of the Output Compare pin logic allows initialization of the OC*nx* state before the output is enabled. Note that some COM*nx*1:0 bit settings are reserved for certain modes of operation (see section "Register Description" on page 266).

The COM*nx*1:0 bits have no effect on the Input Capture unit.

Figure 18-5. Compare Match Output Unit, Schematic



18.8.1 Compare Output Mode and Waveform Generation

The Waveform Generator uses the COM*nx*1:0 bits differently in normal, CTC and PWM modes. A setting of COM*nx*1:0 = 0 tells the Waveform Generator in all modes that no action on the OC*nx* Register is to be performed on the next compare match. For compare output actions in the non-PWM modes refer to Table 18-2. For fast PWM mode refer to Table 18-3 and for phase-correct and phase-and-frequency-correct PWM refer to Table 18-4.

A change of the COM*nx*1:0 bits state will have effect at the first compare match after the bits are written. For non-PWM modes, the action can be forced to have immediate effect by using the FOC*nx* strobe bits.

Table 18-2 shows the COM*nx*1:0 bit functionality when the WGM*n*3:0 bits are set to a normal or a CTC mode (non-PWM).

Table 18-2. Compare Output Mode, non-PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OC*n*A/OC*n*B/OC*n*C disconnected.
0	1	Toggle OC*n*A/OC*n*B/OC*n*C on compare match.
1	0	Clear OC*n*A/OC*n*B/OC*n*C on compare match (set output to low level).
1	1	Set OC*n*A/OC*n*B/OC*n*C on compare match (set output to high level).

Table 18-3 shows the COM*nx*1:0 bit functionality when the WGM*n*3:0 bits are set to the fast PWM mode.

Table 18-3. Compare Output Mode, Fast PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OC*n*A/OC*n*B/OC*n*C disconnected.
0	1	WGM13:0 = 14 or 15: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected.
1	0	Clear OC*n*A/OC*n*B/OC*n*C on compare match; set OC*n*A/OC*n*B/OC*n*C at BOTTOM (non-inverting mode).
1	1	Set OC*n*A/OC*n*B/OC*n*C on compare match, clear OC*n*A/OC*n*B/OC*n*C at BOTTOM (inverting mode).

Note: 1. A special case occurs when OCR*n*A/OCR*n*B/OCR*n*C equals TOP and COM*n*A1/COM*n*B1/COM*n*C1 is set. In this case the compare match is ignored, but the set or clear is done at BOTTOM. See "Fast PWM Mode" on page 258 for more details.

Table 18-4 shows the COM*nx*1:0 bit functionality when the WGM*n*3:0 bits are set to the phase correct and phase and frequency correct PWM mode.

Table 18-4. Compare Output Mode, Phase Correct and Phase/Frequency Correct PWM

COMnA1 COMnB1 COMnC1	COMnA0 COMnB0 COMnC0	Description
0	0	Normal port operation, OC*n*A/OC*n*B/OC*n*C disconnected.
0	1	WGM13:0 =9 or 11: Toggle OC1A on Compare Match, OC1B and OC1C disconnected (normal port operation). For all other WGM1 settings, normal port operation, OC1A/OC1B/OC1C disconnected.
1	0	Clear OC*n*A/OC*n*B/OC*n*C on compare match when up-counting. Set OC*n*A/OC*n*B/OC*n*C on compare match when down-counting.
1	1	Set OC*n*A/OC*n*B/OC*n*C on compare match when up-counting. Clear OC*n*A/OC*n*B/OC*n*C on compare match when down-counting.

Note: 1. A special case occurs when OCR*n*A/OCR*n*B/OCR*n*C equals TOP and COM*n*A1/COM*n*B1/COM*n*C1 is set. See "Phase and Frequency Correct PWM Mode" on page 262 for more details.

18.9 Modes of Operation

The mode of operation i.e., the behavior of the Timer/Counter and the Output Compare pins, is defined by the combination of the Waveform Generation mode (WGM*n*3:0) and the Compare Output mode (COM*nx*1:0) bits. The Compare Output mode bits do not affect the counting sequence, but the Waveform Generation mode bits do. The COM*nx*1:0 bits control whether the PWM output generated should be inverted or not (inverted or non-inverted PWM). For non-PWM modes the COM*nx*1:0 bits control if the output should be set, cleared or toggle at a compare match (See "Compare Match Output Unit" on page 254)

Table 18-5. Waveform Generation Mode Bit Description [(1)]

Mode	WGMn3	WGMn2 (CTCn)	WGMn1 (PWMn1)	WGMn0) (PWMn0)	Timer/Counter Mode of Operation	TOP	Update of OCRnx at	TOVn Flag Set on
0	0	0	0	0	Normal	0xFFFF	Immediate	MAX
1	0	0	0	1	PWM, Phase Correct, 8-bit	0x00FF	TOP	BOTTOM
2	0	0	1	0	PWM, Phase Correct, 9-bit	0x01FF	TOP	BOTTOM
3	0	0	1	1	PWM, Phase Correct, 10-bit	0x3FF	TOP	BOTTOM
4	0	1	0	0	CTC	OCR*n*A	Immediate	MAX
5	0	1	0	1	Fast PWM, 8-bit	0x00FF	BOTTOM	TOP
6	0	1	1	0	Fast PWM, 9-bit	0x01FF	BOTTOM	TOP
7	0	1	1	1	Fast PWM, 10-bit	0x03FF	BOTTOM	TOP
8	1	0	0	0	PWM, Phase and Frequency Correct	ICR*n*	BOTTOM	BOTTOM
9	1	0	0	1	PWM, Phase and Frequency Correct	OCR*n*A	BOTTOM	BOTTOM
10	1	0	1	0	PWM, Phase Correct	ICR*n*	TOP	BOTTOM
11	1	0	1	1	PWM, Phase Correct	OCR*n*A	TOP	BOTTOM
12	1	1	0	0	CTC	ICR*n*	Immediate	MAX
13	1	1	0	1	(Reserved)	–	–	–
14	1	1	1	0	Fast PWM	ICR*n*	BOTTOM	TOP
15	1	1	1	1	Fast PWM	OCR*n*A	BOTTOM	TOP

Notes: 1. The CTC*n* and PWM*n*1:0 bit definition names are obsolete. Use the WGM*n*2:0 definitions. However, the functionality and location of these bits are compatible with previous versions of the timer.

For detailed timing information refer to "Timer/Counter Timing Diagrams" on page 264.

18.9.1 Normal Mode

The simplest mode of operation is the Normal mode (WGM*n*3:0 = 0). In this mode the counting direction is always up (incrementing) and no counter clear is performed. The counter simply overruns when it passes its maximum 16-bit value (MAX = 0xFFFF) and then restarts from the BOTTOM (0x0000). In normal operation the Timer/Counter Overflow Flag (TOV*n*) will be set in the same timer clock cycle as the TCNT*n* becomes zero. In this case the TOV*n* Flag behaves like a 17th bit, except that it is only set and not cleared. However the timer resolution can be increased by software when combined with the timer overflow interrupt that automatically clears the TOV*n* Flag. There are no special cases to consider in the Normal mode. A new counter value can be written anytime.

The Input Capture unit is easy to use in Normal mode. However it is important to note that the maximum interval between the external events must not exceed the resolution of the counter. The timer overflow interrupt or the prescaler must be used to extend the resolution for the capture unit if the intervals between events are too long.

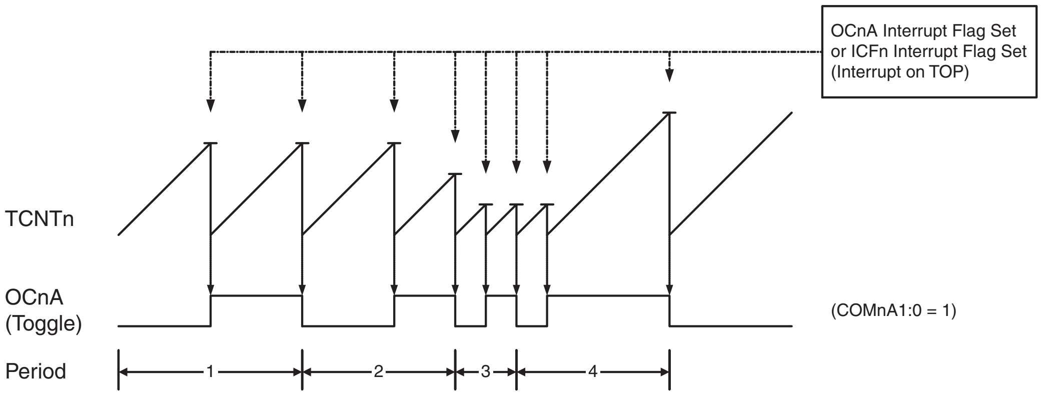
The Output Compare units can be used to generate interrupts at some given time. Using the Output Compare to generate waveforms in Normal mode is not recommended because this will occupy too much CPU time.

18.9.2 Clear Timer on Compare Match (CTC) Mode

In Clear Timer on Compare (CTC) mode (WGM*n*3:0 = 4 or 12), the OCR*n*A or ICR*n* Register are used to manipulate the counter resolution. In CTC mode the counter is cleared to zero when the counter value (TCNT*n*) matches either the OCR*n*A (WGM*n*3:0 = 4) or the ICR*n* (WGM*n*3:0 = 12). The OCR*n*A or ICR*n* define the top value for the counter, hence also its resolution. This mode allows greater control of the compare match output frequency. It also simplifies the operation of counting external events.

The timing diagram for the CTC mode is shown in the following figure. The counter value (TCNT*n*) increases until a compare match occurs with either OCR*n*A or ICR*n*, and then counter (TCNT*n*) is cleared.

Figure 18-6. CTC Mode Timing Diagram



Each time the counter reaches the TOP value an interrupt can be generated by either the OCF*n*A or ICF*n* Flag according to the register used to define the TOP value. If the interrupt is enabled, the interrupt handler routine can be used for updating the TOP value. However, changing TOP to a value close to BOTTOM when the counter is running with no or a low prescaler value must be done with care since the CTC mode does not have the double buffering feature. The counter will miss the compare match if the new value written to OCR*n*A or ICR*n* is lower than the current value of TCNT*n*. The counter will then have to count to its maximum value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. In many cases this feature is not desirable. The fast PWM mode is available as an alternative using OCR*n*A for defining TOP (WGMn3:0 = 15). The OCR*n*A then will be double buffered.

For generating a waveform output in CTC mode, the OC*n*A output can be set to toggle its logical level on each compare match by setting the Compare Output mode bits to toggle mode (COM*n*A1:0 = 1). The OC*n*A value will not be visible on the port pin unless the data direction for the pin is set to output (DDR_OC*n*A = 1). The waveform generated will have a maximum frequency of $f_{OCnA} = f_{clk\_I/O}/2$ when OCR*n*A is set to zero (0x0000). The waveform frequency is given by the following equation:

$$f_{OCnA} = \frac{f_{CLK\_I/O}}{2 \cdot N \cdot (1 + OCRnA)}$$

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

As for the Normal mode of operation, the TOV*n* Flag is set in the same clock cycle of the timer when the counter changes from MAX to 0x0000.

18.9.3 Fast PWM Mode

The fast Pulse Width Modulation (PWM) mode (WGM*n*3:0 = 5, 6, 7, 14 or 15) provides a high frequency PWM waveform generation option. The fast PWM differs from the other PWM options by its single-slope operation. The counter counts from BOTTOM to TOP then restarts from BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC*nx*) is cleared on the compare match between TCNT*n* and OCR*nx*, and

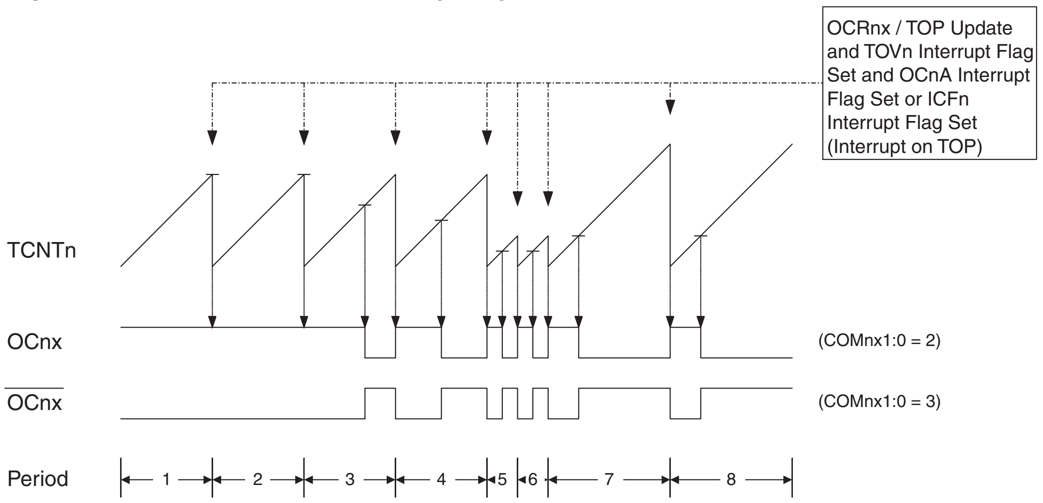
OC*nx* is set at BOTTOM. In inverting Compare Output mode output is set on compare match and cleared at BOTTOM. Due to the single-slope operation, the operating frequency of the fast PWM mode can be twice as high as the phase-correct and phase and frequency correct PWM modes that use dual-slope operation. This high frequency makes the fast PWM mode well suited for power regulation, rectification and DAC applications. High frequency allows physically small sized external components (coils, capacitors), hence reducing total system cost.

The PWM resolution for fast PWM can be fixed to 8-, 9-, or 10-bit, or defined by either ICR*n* or OCR*n*A. The minimum resolution allowed is 2-bit (ICR*n* or OCR*n*A set to 0x0003), and the maximum resolution is 16-bit (ICR*n* or OCR*n*A set to MAX). The PWM resolution R_{FPWM} in bits can be calculated with the following equation:

$$R_{FPWM} = \frac{\log(TOP + 1)}{\log(2)}$$

In fast PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF or 0x03FF (WGM*n*3:0 = 5, 6 or 7), the value in ICR*n* (WGM*n*3:0 = 14) or the value in OCR*n*A (WGM*n*3:0 = 15). The counter is then cleared at the following timer clock cycle. The timing diagram for the fast PWM mode is shown in Figure 18-7. The figure shows fast PWM mode when OCR*n*A or ICR*n* is used to define TOP. The TCNT*n* value is in the timing diagram shown as a histogram for illustrating the single-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT*n* slopes represent compare matches between OCR*nx* and TCNT*n*. The OC*nx* Interrupt Flag will be set when a compare match occurs.

Figure 18-7. Fast PWM Mode Timing Diagram


OCRnx / TOP Update and TOVn Interrupt Flag Set and OCnA Interrupt Flag Set or ICFn Interrupt Flag Set (Interrupt on TOP)
TCNTn
OCnx
(COMnx1:0 = 2)
$\overline{\text{OCnx}}$
(COMnx1:0 = 3)
Period
1
2
3
4
5
6
7
8

The Timer/Counter Overflow Flag (TOV*n*) is set each time the counter reaches TOP. In addition the OC*n*A or ICF*n* Flag is set at the same timer clock cycle as TOV*n* is set when either OCR*n*A or ICR*n* is used to define the TOP value. If one of the interrupts are enabled, the interrupt handler routine can be utilized for updating the TOP and compare values.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. A compare match will never occur between the TCNT*n* and the OCR*nx* if the TOP value is lower than any of the Compare Registers. Note that when working with fixed TOP values the unused bits are masked to zero when any of the OCR*nx* Registers are written.

The procedure for updating ICR*n* differs from updating OCR*n*A when used for defining the TOP value. The ICR*n* Register is not double buffered. This means that if ICR*n* is changed to a low value while the counter is running with no or a low prescaler value, there is a risk that the newly written ICR*n* value is lower than the current value of TCNT*n*. In consequence the counter will miss the compare match at the TOP value. The counter must then count to the MAX value (0xFFFF) and wrap around starting at 0x0000 before the compare match can occur. The OCR*n*A Register is double buffered though. This feature allows writing the OCR*n*A I/O location at anytime. When the OCR*n*A I/O location is written the new value will be put first into the OCR*n*A Buffer Register. The OCR*n*A Compare Register will then be updated with the value in the Buffer Register at the next clock cycle of the timer when TCNT*n* matches TOP. The update is done at the same timer clock cycle as the TCNT*n* is cleared and the TOV*n* Flag is set.

The definition of TOP with the ICR*n* Register works well for fixed TOP values. Combined with ICR*n*, the OCR*n*A Register is free to be used for generating a PWM output on OC*n*A. However, if the base PWM frequency is actively changed (by modifying the TOP value), working with the OCR*n*A as TOP is clearly a better choice due to its double buffer feature.

In fast PWM mode the compare units allow the generation of PWM waveforms on the OC*nx* pins. Setting the COM*nx*1:0 bits to 2 will produce a non-inverted PWM and an inverted PWM output can be generated by setting the COM*nx*1:0 to 3 (see Table 18-3 on page 256). The actual OC*nx* value will only be visible on the port pin if the data direction of the port pin is set to output (DDR_OC*nx*). The PWM waveform is generated by setting (or clearing) the OC*nx* Register at the compare match between OCR*nx* and TCNT*n*, and by clearing (or setting) the OC*nx* Register at the timer clock cycle the counter is cleared (changes from TOP to BOTTOM).

The PWM frequency of the output $f_{OCnxPWM}$ can be calculated with the following equation:

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

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

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

A frequency (with 50% duty cycle) waveform output in fast PWM mode can be achieved by setting OC*n*A to toggle its logical level on each compare match (COM*n*A1:0 = 1). This applies only if OCR1A is used to define the TOP value (WGM13:0 = 15). The waveform generated will have a maximum frequency of $f_{OCnA} = f_{clk\_I/O}/2$ when OCR*n*A is set to zero (0x0000). This feature is similar to the OC*n*A toggle in CTC mode, except the double buffer feature of the Output Compare unit is enabled in the fast PWM mode.

18.9.4 Phase Correct PWM Mode

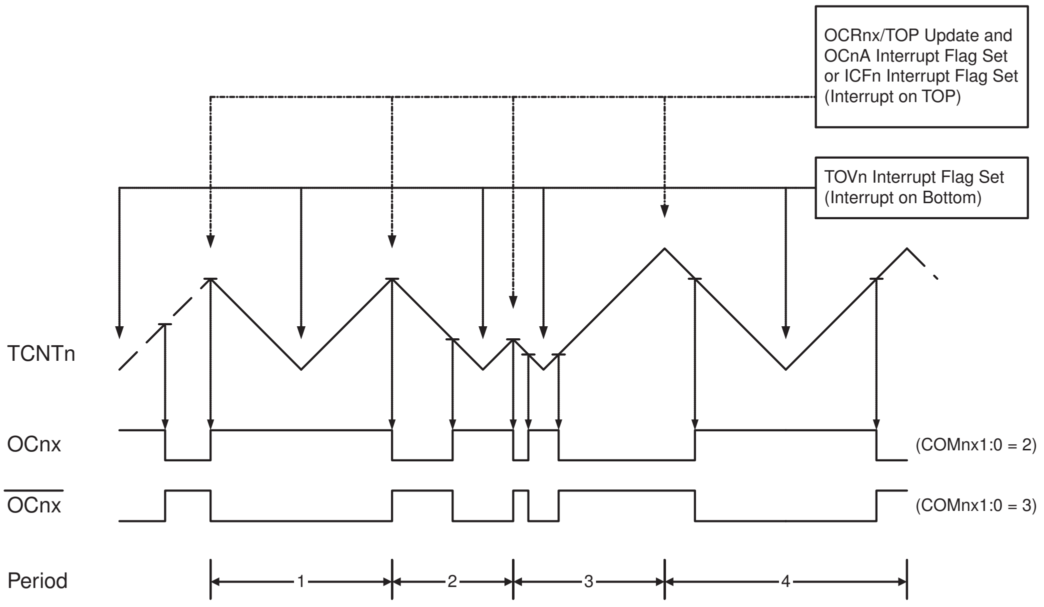
The phase correct Pulse Width Modulation (PWM) mode (WGM*n*3:0 = 1, 2, 3, 10 or 11) provides a high resolution phase correct PWM waveform generation option. The phase correct PWM mode is, like the phase and frequency correct PWM mode, based on a dual-slope operation. The counter counts repeatedly from BOTTOM (0x0000) to TOP and then from TOP to BOTTOM. In non-inverting Compare Output mode, the Output Compare (OC*nx*) is cleared on the compare match between TCNT*n* and OCR*nx* while up-counting, and set on the compare match while down-counting. In inverting Output Compare mode, the operation is inverted. The dual-slope operation has a lower maximum operation frequency than single slope operation. However these modes are preferred for motor control applications due to the symmetric feature of the dual-slope PWM modes.

The PWM resolution for the phase correct PWM mode can be fixed to 8, 9 or 10 bit, or be defined by either ICR*n* or OCR*n*A. The minimum resolution allowed is 2 bit (ICR*n* or OCRnA set to 0x0003), and the maximum resolution is 16-bit (ICR*n* or OCR*n*A set to MAX). The PWM resolution R_{PCPWM} in bits can be calculated with the following equation:

$$R_{PCPWM} = \frac{\log(TOP+1)}{\log(2)}$$

In phase correct PWM mode the counter is incremented until the counter value matches either one of the fixed values 0x00FF, 0x01FF or 0x03FF (WGM*n*3:0 = 1, 2, or 3), the value in ICR*n* (WGM*n*3:0 = 10) or the value in OCR*n*A (WGM*n*3:0 = 11). The counter has then reached the TOP and changes the count direction. The TCNT*n* value will be equal to TOP for one timer clock cycle. The timing diagram for the phase correct PWM mode is shown on Figure 18-8 below. The figure shows phase correct PWM mode when OCR*n*A or ICR*n* is used to define TOP. The TCNT*n* value is shown in the timing diagram as a histogram illustrating the dual-slope operation. The diagram includes non-inverted and inverted PWM outputs. The small horizontal line marks on the TCNT*n* slopes represent compare matches between OCR*nx* and TCNT*n*. The OC*nx* Interrupt Flag will be set when a compare match occurs.

Figure 18-8. Phase Correct PWM Mode Timing Diagram


OCRnx/TOP Update and OCnA Interrupt Flag Set or ICFn Interrupt Flag Set (Interrupt on TOP)
TOVn Interrupt Flag Set (Interrupt on Bottom)
TCNTn
OCnx
OCnx
(COMnx1:0 = 2)
(COMnx1:0 = 3)
Period
1
2
3
4

The Timer/Counter Overflow Flag (TOV*n*) is set each time the counter reaches BOTTOM. When either OCR*n*A or ICR*n* is used for defining the TOP value, the OC*n*A or ICF*n* Flag is set accordingly at the same timer clock cycle as the OCR*nx* Registers are updated with the double buffer value (at TOP). The Interrupt Flags can be used to generate an interrupt each time the counter reaches the TOP or BOTTOM value.

When changing the TOP value the program must ensure that the new TOP value is higher or equal to the value of all of the Compare Registers. If the TOP value is lower than any of the Compare Registers, a compare match will never occur between the