8. Power Management and Sleep Modes

8.1 Features

- Power management for adjusting power consumption and functions
- Five sleep modes
  - Idle
  - Power down
  - Power save
  - Standby
  - Extended standby
- Power reduction register to disable clock and turn off unused peripherals in active and idle modes

8.2 Overview

Various sleep modes and clock gating are provided in order to tailor power consumption to application requirements. This enables the XMEGA microcontroller to stop unused modules to save power.

All sleep modes are available and can be entered from active mode. In active mode, the CPU is executing application code. When the device enters sleep mode, program execution is stopped and interrupts or a reset is used to wake the device again. The application code decides which sleep mode to enter and when. Interrupts from enabled peripherals and all enabled reset sources can restore the microcontroller from sleep to active mode.

In addition, power reduction registers provide a method to stop the clock to individual peripherals from software. When this is done, the current state of the peripheral is frozen, and there is no power consumption from that peripheral. This reduces the power consumption in active mode and idle sleep modes and enables much more fine-tuned power management than sleep modes alone.

8.3 Sleep Modes

Sleep modes are used to shut down modules and clock domains in the microcontroller in order to save power. XMEGA microcontrollers have five different sleep modes tuned to match the typical functional stages during application execution. A dedicated sleep instruction (SLEEP) is available to enter sleep mode. Interrupts are used to wake the device from sleep, and the available interrupt wake-up sources are dependent on the configured sleep mode. When an enabled interrupt occurs, the device will wake up and execute the interrupt service routine before continuing normal program execution from the first instruction after the SLEEP instruction. If other, higher priority interrupts are pending when the wake-up occurs, their interrupt service routines will be executed according to their priority before the interrupt service routine for the wake-up interrupt is executed. After wake-up, the CPU is halted for four cycles before execution starts.

Table 8-1 on page 104 shows the different sleep modes and the active clock domains, oscillators, and wake-up sources.

Table 8-1. Active clock domains and wake-up sources in the different sleep modes.

	Active Clock Domain			Oscillators		Wake-up Sources				
Sleep Modes	CPU Clock	Peripheral and USB Clock	RTC Clock	System Clock Source	RTC Clock Source	USB Resume	Asynchronous Port Interrupts	TWI Address Match Interrupts	Real Time Clock Interrupts	All Interrupts
Idle		X	X	X	X	X	X	X	X	X
Power down						X	X	X		
Power save			X		X	X	X	X	X	
Standby				X		X	X	X		
Extended standby			X	X	X	X	X	X	X	

The wake-up time for the device is dependent on the sleep mode and the main clock source. The startup time for the system clock source must be added to the wake-up time for sleep modes where the system clock source is not kept running. For details on the startup time for the different oscillator options, refer to “System Clock and Clock Options” on page 82.

The content of the register file, SRAM and registers are kept during sleep. If a reset occurs during sleep, the device will reset, start up, and execute from the reset vector.

8.3.1 Idle Mode

In idle mode the CPU and nonvolatile memory are stopped (note that any ongoing programming will be completed), but all peripherals, including the interrupt controller, event system and DMA controller are kept running. Any enabled interrupt will wake the device.

8.3.2 Power-down Mode

In power-down mode, all clocks, including the real-time counter clock source, are stopped. This allows operation only of asynchronous modules that do not require a running clock. The only interrupts that can wake up the MCU are the two-wire interface address match interrupt, asynchronous port interrupts, and the USB resume interrupt.

8.3.3 Power-save Mode

Power-save mode is identical to power down, with one exception. If the real-time counter (RTC) is enabled, it will keep running during sleep, and the device can also wake up from either an RTC overflow or compare match interrupt.

8.3.4 Standby Mode

Standby mode is identical to power down, with the exception that the enabled system clock sources are kept running while the CPU, peripheral, and RTC clocks are stopped. This reduces the wake-up time.

8.3.5 Extended Standby Mode

Extended standby mode is identical to power-save mode, with the exception that the enabled system clock sources are kept running while the CPU and peripheral clocks are stopped. This reduces the wake-up time.

8.4 Power Reduction Registers

The power reduction (PR) registers provide a method to stop the clock to individual peripherals. When this is done, the current state of the peripheral is frozen and the associated I/O registers cannot be read or written. Resources used by the peripheral will remain occupied; hence, the peripheral should be disabled before stopping the clock. Enabling the clock to a peripheral again puts the peripheral in the same state as before it was stopped. This can be used in idle mode and active modes to reduce the overall power consumption. In all other sleep modes, the peripheral clock is already stopped.

Not all devices have all the peripherals associated with a bit in the power reduction registers. Setting a power reduction bit for a peripheral that is not available will have no effect.

8.5 Minimizing Power Consumption

There are several possibilities to consider when trying to minimize the power consumption in an AVR MCU controlled system. In general, correct sleep modes should be selected and used to ensure that only the modules required for the application are operating.

All unneeded functions should be disabled. In particular, the following modules may need special consideration when trying to achieve the lowest possible power consumption.

8.5.1 Analog-to-Digital Converter - ADC

When entering idle mode, the ADC should be disabled if not used. In other sleep modes, the ADC is automatically disabled. To save power, the ADC should be disabled before entering any sleep mode. When the ADC is turned off and on again, the next conversion will be an extended conversion. Refer to “ADC – Analog-to-Digital Converter” on page 339 for details on ADC operation.

8.5.2 Analog Comparator - AC

When entering idle mode, the analog comparator should be disabled if not used. 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, irrespective of sleep mode. Refer to “AC – Analog Comparator” on page 377 for details on how to configure the analog comparator.

8.5.3 Brownout Detector

If the brownout detector is not needed by the application, this module should be turned off. If the brownout detector is enabled by the BODLEVEL fuses, it will be enabled in all sleep modes, and always consume power. In the deeper sleep modes, it can be turned off and set in sampled mode to reduce current consumption. Refer to “Brownout Detection” on page 112 for details on how to configure the brownout detector.

8.5.4 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. Refer to “WDT – Watchdog Timer” on page 125 for details on how to configure the watchdog timer.

8.5.5 Port Pins

When entering a sleep mode, all port pins should be configured to use minimum power. Most important is to ensure that no pins drive resistive loads. In sleep modes where the Peripheral Clock (Clk_{PER}) is stopped, the input buffers of the device will be disabled. This ensures that no power is consumed by the input logic when not needed.

8.6 Register Description – Sleep

8.6.1 CTRL – Control register

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

- **Bit 7:4 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 3:1 – SMODE[2:0]: Sleep Mode Selection**
  These bits select sleep modes according to Table 8-2 on page 106.

Table 8-2. Sleep mode.

SMODE[2:0]	Group configuration	Description
000	IDLE	Idle mode
001	–	Reserved
010	PDOWN	Power-down mode
011	PSAVE	Power-save mode
100	–	Reserved
101	–	Reserved
110	STDBY	Standby mode
111	ESTDBY	Extended standby mode

- **Bit 0 – SEN: Sleep Enable**
  This bit must be set to make the MCU enter the selected sleep mode when the SLEEP instruction is executed. To avoid unintentional entering of sleep modes, it is recommended to write SEN just before executing the SLEEP instruction and clear it immediately after waking up.

8.7 Register Description – Power Reduction

8.7.1 PRGEN – General Power Reduction register

Bit	7	6	5	4	3	2	1	0
+0x00	–	USB	–	AES	EBI	RTC	EVSYS	DMA
Read/Write	R	R/W	R	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written.
- **Bit 6 – USB: USB Module**
  Setting this bit stops the clock to the USB module. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 5 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written.
- **Bit 4 – AES: AES Module**
  Setting this bit stops the clock to the AES module. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 3 – EBI: External Bus Interface**
  Setting this bit stops the clock to the external bus interface. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 2 – RTC: Real-Time Counter**
  Setting this bit stops the clock to the real-time counter. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 1 – EVSYS: Event System**
  Setting this stops the clock to the event system. When this bit is cleared, the module will continue as before it was stopped.
- **Bit 0 – DMA: DMA Controller**
  Setting this bit stops the clock to the DMA controller. This bit can be set only if the DMA controller is disabled.

8.7.2 PRPA/B – Power Reduction Port A/B register

Bit	7	6	5	4	3	2	1	0
+0x01/+0x02	–	–	–	–	–	DAC	ADC	AC
Read/Write	R	R	R	R	R	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Note: Disabling of analog modules stops the clock to the analog blocks themselves and not only the interfaces.

- **Bit 7:3 – Reserved**
  These bits are unused and reserved for future use. For compatibility with future devices, always write these bits to zero when this register is written.

- **Bit 2 – DAC: Power Reduction DAC**
  Setting this bit stops the clock to the DAC. The DAC should be disabled before stopped.
- **Bit 1 – ADC: Power Reduction ADC**
  Setting this bit stops the clock to the ADC. The ADC should be disabled before stopped.
- **Bit 0 – AC: Power Reduction Analog Comparator**
  Setting this bit stops the clock to the analog comparator. The AC should be disabled before shutdown.

8.7.3 PRPC/D/E/F – Power Reduction Port C/D/E/F register

Bit	7	6	5	4	3	2	1	0
+0x03/+0x04/+0x05/+0x06	–	TWI	USART1	USART0	SPI	HIRES	TC1	TC0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

- **Bit 7 – Reserved**
  This bit is unused and reserved for future use. For compatibility with future devices, always write this bit to zero when this register is written.
- **Bit 6 – TWI: Two-Wire Interface**
  Setting this bit stops the clock to the two-wire interface. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 5 – USART1**
  Setting this bit stops the clock to USART1. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 4 – USART0**
  Setting this bit stops the clock to USART0. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 3 – SPI: Serial Peripheral Interface**
  Setting this bit stops the clock to the SPI. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 2 – HIRES: High-Resolution Extension**
  Setting this bit stops the clock to the high-resolution extension for the timer/counters. When this bit is cleared, the peripheral should be reinitialized to ensure proper operation.
- **Bit 1 – TC1: Timer/Counter 1**
  Setting this bit stops the clock to timer/counter 1. When this bit is cleared, the peripheral will continue like before the shut down.
- **Bit 0 – TC0: Timer/Counter 0**
  Setting this bit stops the clock to timer/counter 0. When this bit is cleared, the peripheral will continue like before the shut down.

8.8 Register summary – Sleep

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
+0x00	CTRL	–	–		–	SMODE[2:0]			SEN	106

8.9 Register summary – Power reduction

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Page
+0x00	PRGEN	–	USB	–	AES	EBI	RTC	EVSYS	DMA	106
+0x01	PRPA	–	–	–	–	–	DAC	ADC	AC	107
+0x02	PRPB	–	–	–	–	–	DAC	ADC	AC	107
+0x03	PRPC	–	TWI	USART1	USART0	SPI	HIRES	TC1	TC0	108
+0x04	PRPD	–	TWI	USART1	USART0	SPI	HIRES	TC1	TC0	108
+0x05	PRPE	–	TWI	USART1	USART0	SPI	HIRES	TC1	TC0	108
+0x06	PRPF	–	TWI	USART1	USART0	SPI	HIRES	TC1	TC0	108
+0x07	Reserved	–	–	–	–	–	–	–	–	