



Chipsmall Limited consists of a professional team with an average of over 10 year of expertise in the distribution of electronic components. Based in Hongkong, we have already established firm and mutual-benefit business relationships with customers from,Europe,America and south Asia,supplying obsolete and hard-to-find components to meet their specific needs.

With the principle of “Quality Parts,Customers Priority,Honest Operation,and Considerate Service”,our business mainly focus on the distribution of electronic components. Line cards we deal with include Microchip,ALPS,ROHM,Xilinx,Pulse,ON,Everlight and Freescale. Main products comprise IC,Modules,Potentiometer,IC Socket,Relay,Connector.Our parts cover such applications as commercial,industrial, and automotives areas.

We are looking forward to setting up business relationship with you and hope to provide you with the best service and solution. Let us make a better world for our industry!



Contact us

Tel: +86-755-8981 8866 Fax: +86-755-8427 6832

Email & Skype: info@chipsmall.com Web: www.chipsmall.com

Address: A1208, Overseas Decoration Building, #122 Zhenhua RD., Futian, Shenzhen, China



Features

- High-performance, Low-power 8/16-bit Atmel® AVR® XMEGA™ Microcontroller
- Non-volatile Program and Data Memories
 - 16 KB - 128 KB of In-System Self-Programmable Flash
 - 4 KB - 8 KB Boot Code Section with Independent Lock Bits
 - 1 KB - 2 KB EEPROM
 - 2 KB - 8 KB Internal SRAM
- Peripheral Features
 - Four-channel DMA Controller with support for external requests
 - Eight-channel Event System
 - Five 16-bit Timer/Counters
 - Three Timer/Counters with 4 Output Compare or Input Capture channels
 - Two Timer/Counters with 2 Output Compare or Input Capture channels
 - High-Resolution Extensions on all Timer/Counters
 - Advanced Waveform Extension on one Timer/Counter
 - Five USARTs
 - IrDA Extension on one USART
 - Two Two-Wire Interfaces with dual address match (I²C and SMBus compatible)
 - Two SPIs (Serial Peripheral Interfaces) peripherals
 - AES and DES Crypto Engine
 - 16-bit Real Time Counter with Separate Oscillator
 - One Twelve-channel, 12-bit, 2 Msps Analog to Digital Converter
 - One Two-channel, 12-bit, 1 Msps Digital to Analog Converter
 - Two Analog Comparators with Window compare function
 - External Interrupts on all General Purpose I/O pins
 - Programmable Watchdog Timer with Separate On-chip Ultra Low Power Oscillator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal and External Clock Options with PLL
 - Programmable Multi-level Interrupt Controller
 - Sleep Modes: Idle, Power-down, Standby, Power-save, Extended Standby
 - Advanced Programming, Test and Debugging Interfaces
 - PDI (Program and Debug Interface) for programming, test and debugging
- I/O and Packages
 - 34 Programmable I/O Lines
 - 44 - lead TQFP
 - 44 - pad VQFN/QFN
 - 49 - ball VFBGA
- Operating Voltage
 - 1.6 – 3.6V
- Speed performance
 - 0 – 12 MHz @ 1.6 – 3.6V
 - 0 – 32 MHz @ 2.7 – 3.6V

Typical Applications

- Industrial control
- Climate control
- Hand-held battery applications
- Factory automation
- ZigBee
- Power tools
- Building control
- Motor control
- HVAC
- Board control
- Networking
- Metering
- White Goods
- Optical
- Medical Applications



8/16-bit AVR® XMEGA A4 Microcontroller

ATxmega128A4
ATxmega64A4
ATxmega32A4
ATxmega16A4

Not recommended for
new designs - Use
XMEGA A4U series

8069R-AVR-06/2013



Not recommended for new designs -
Use XMEGA A4U series

1. Ordering Information

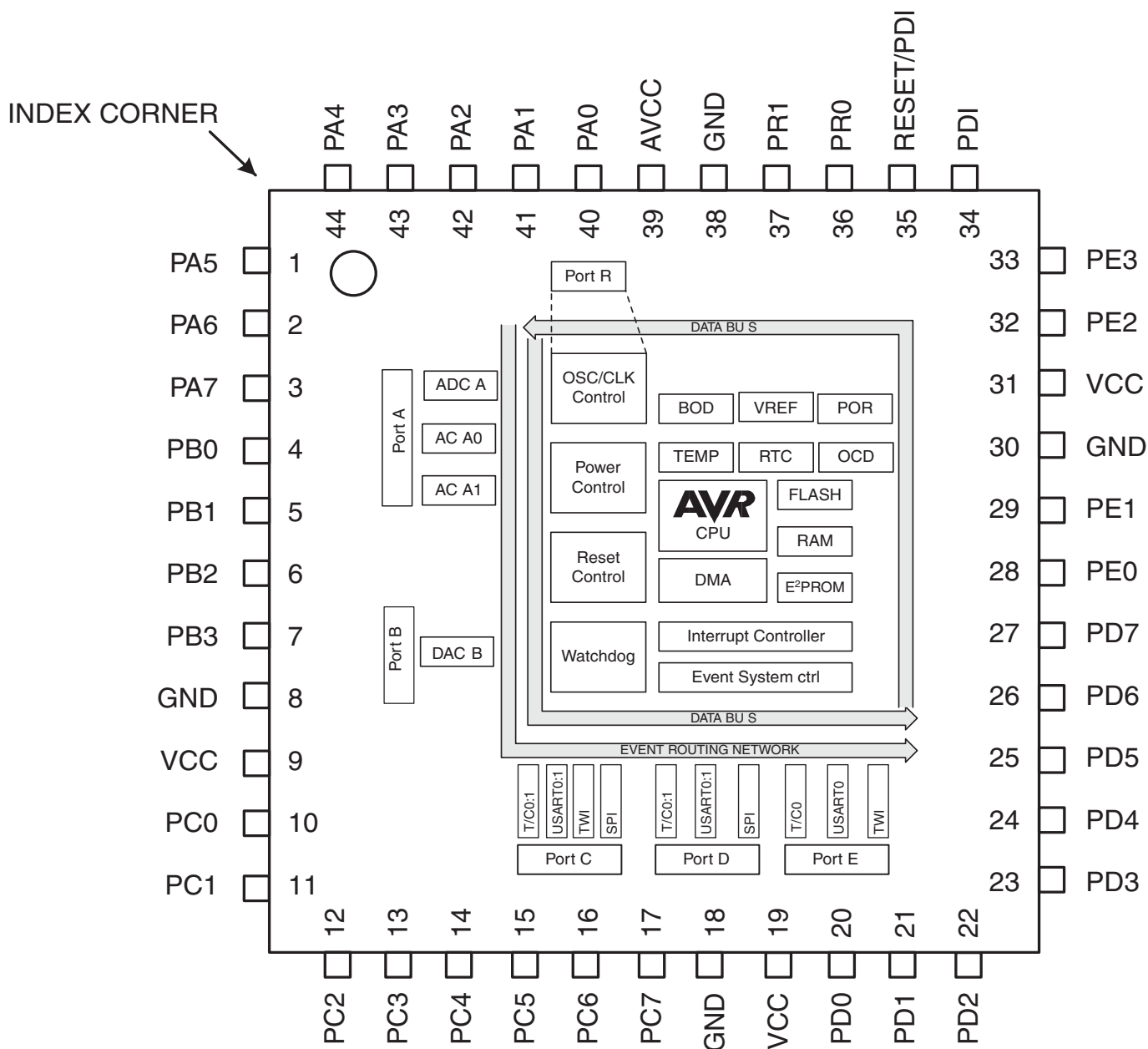
Ordering Code	Flash	E ²	SRAM	Speed (MHz)	Power Supply	Package ⁽¹⁾⁽²⁾⁽³⁾	Temp
ATxmega128A4-AU	128 KB + 8 KB	2 KB	8 KB	32	1.6 - 3.6V	44A	-40°C - 85°C
ATxmega64A4-AU	64 KB + 4 KB	2 KB	4 KB	32	1.6 - 3.6V		
ATxmega32A4-AU	32 KB + 4 KB	1 KB	4 KB	32	1.6 - 3.6V		
ATxmega16A4-AU	16 KB + 4 KB	1 KB	2 KB	32	1.6 - 3.6V		
ATxmega128A4-MH	128 KB + 8 KB	2 KB	8 KB	32	1.6 - 3.6V	44M1	
ATxmega64A4-MH	64 KB + 4 KB	2 KB	4 KB	32	1.6 - 3.6V		
ATxmega32A4-MH	32 KB + 4 KB	1 KB	4 KB	32	1.6 - 3.6V		
ATxmega16A4-MH	16 KB + 4 KB	1 KB	2 KB	32	1.6 - 3.6V		
ATxmega32A4-CU	32 KB + 4K	1 KB	4 KB	32	1.6 - 3.6V	49C2	
ATxmega16A4-CU	16 KB + 4 KB	1 KB	2 KB	32	1.6 - 3.6V		

- Notes:
1. This device can also be supplied in wafer form. Please contact your local Atmel sales office for detailed ordering information.
 2. Pb-free packaging, complies to the European Directive for Restriction of Hazardous Substances (RoHS directive). Also Halide free and fully Green.
 3. For packaging information see "[Packaging information](#)" on page 58.

Package Type	
44A	44-Lead, 10 x 10 mm Body Size, 1.0 mm Body Thickness, 0.8 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)
44M1	44-Pad, 7x7x1 mm Body, Lead Pitch 0.50 mm, 5.20 mm Exposed Pad, Thermally Enhanced Plastic Very Thin Quad No Lead Package (VQFN)
49C2	49-Ball (7 x 7 Array), 0.65 mm Pitch, 5.0 x 5.0 x 1.0 mm, Very Thin, Fine-Pitch Ball Grid Array Package (VFPGA)

2. Pinout/Block Diagram

Figure 2-1. Block Diagram and TQFP/QFN pinout



- Notes:
1. For full details on pinout and pin functions refer to ["Pinout and Pin Functions" on page 49](#).
 2. The large center pad underneath the QFN/MLF package should be soldered to ground on the board to ensure good mechanical stability.

Figure 2-2. VFBGA pinout

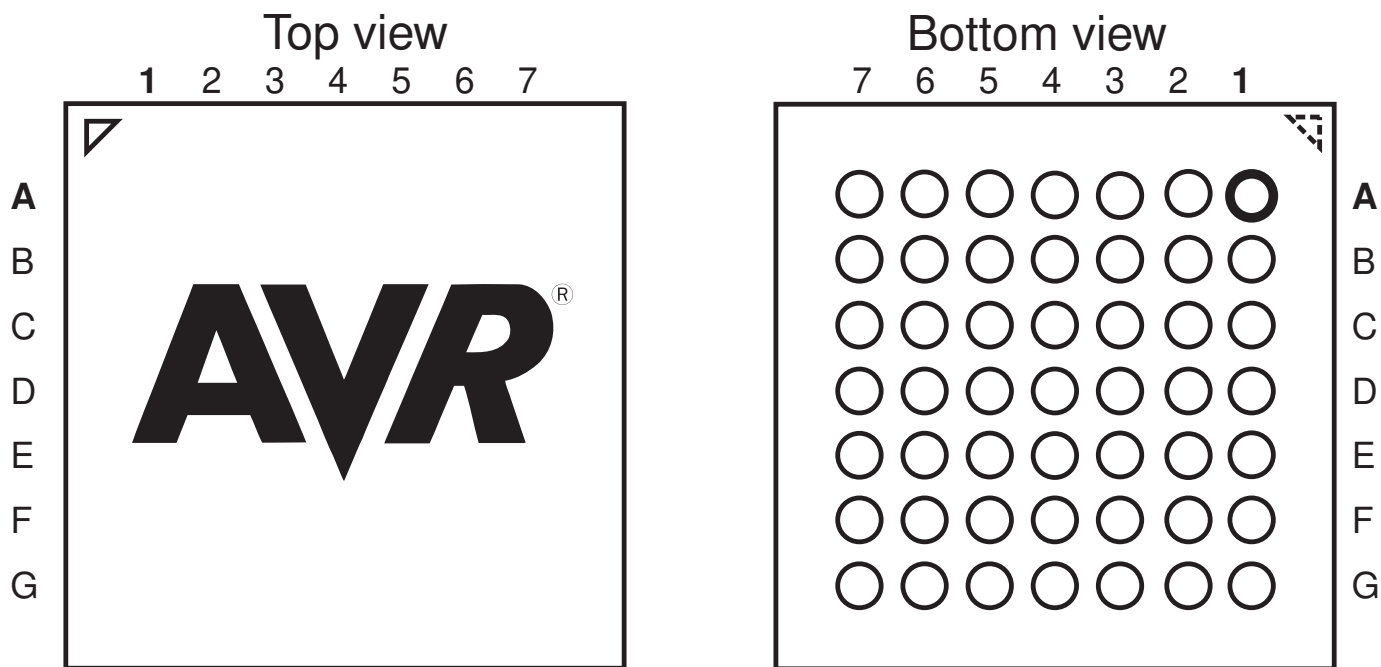


Table 2-1. VFBGA pinout

	1	2	3	4	5	6	
A	PA3	AVCC	GND	PR1	PR0	PDI	PE3
B	PA4	PA1	PA0	GND	RESET/PDI	PE2	VCC
C	PA5	PA2	PA6	PA7	GND	PE1	GND
D	PB1	PB2	PB3	PB0	GND	PD7	PE0
E	GND	GND	PC3	GND	PD4	PD5	PD6
F	VCC	PC0	PC4	PC6	PD0	PD1	PD3
G	PC1	PC2	PC5	PC7	GND	VCC	PD2

3. Overview

The Atmel® AVR® XMEGA™ A4 is a family of low power, high performance and peripheral rich CMOS 8/16-bit microcontrollers based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the XMEGA A4 achieves throughputs approaching 1 Million Instructions Per Second (MIPS) per MHz allowing the system designer to optimize power consumption versus processing speed.

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

The XMEGA A4 devices provide the following features: In-System Programmable Flash with Read-While-Write capabilities, Internal EEPROM and SRAM, four-channel DMA Controller, eight-channel Event System, Programmable Multi-level Interrupt Controller, 34 general purpose I/O lines, 16-bit Real Time Counter (RTC), five flexible 16-bit Timer/Counters with compare modes and PWM, five USARTs, two Two Wire Serial Interfaces (TWIs), two Serial Peripheral Interfaces (SPIs), AES and DES crypto engine, one Twelve-channel, 12-bit ADC with optional differential input with programmable gain, one Two-channel 12-bit DAC, two analog comparators with window mode, programmable Watchdog Timer with separate Internal Oscillator, accurate internal oscillators with PLL and prescaler and programmable Brown-Out Detection.

The Program and Debug Interface (PDI), a fast 2-pin interface for programming and debugging, is available.

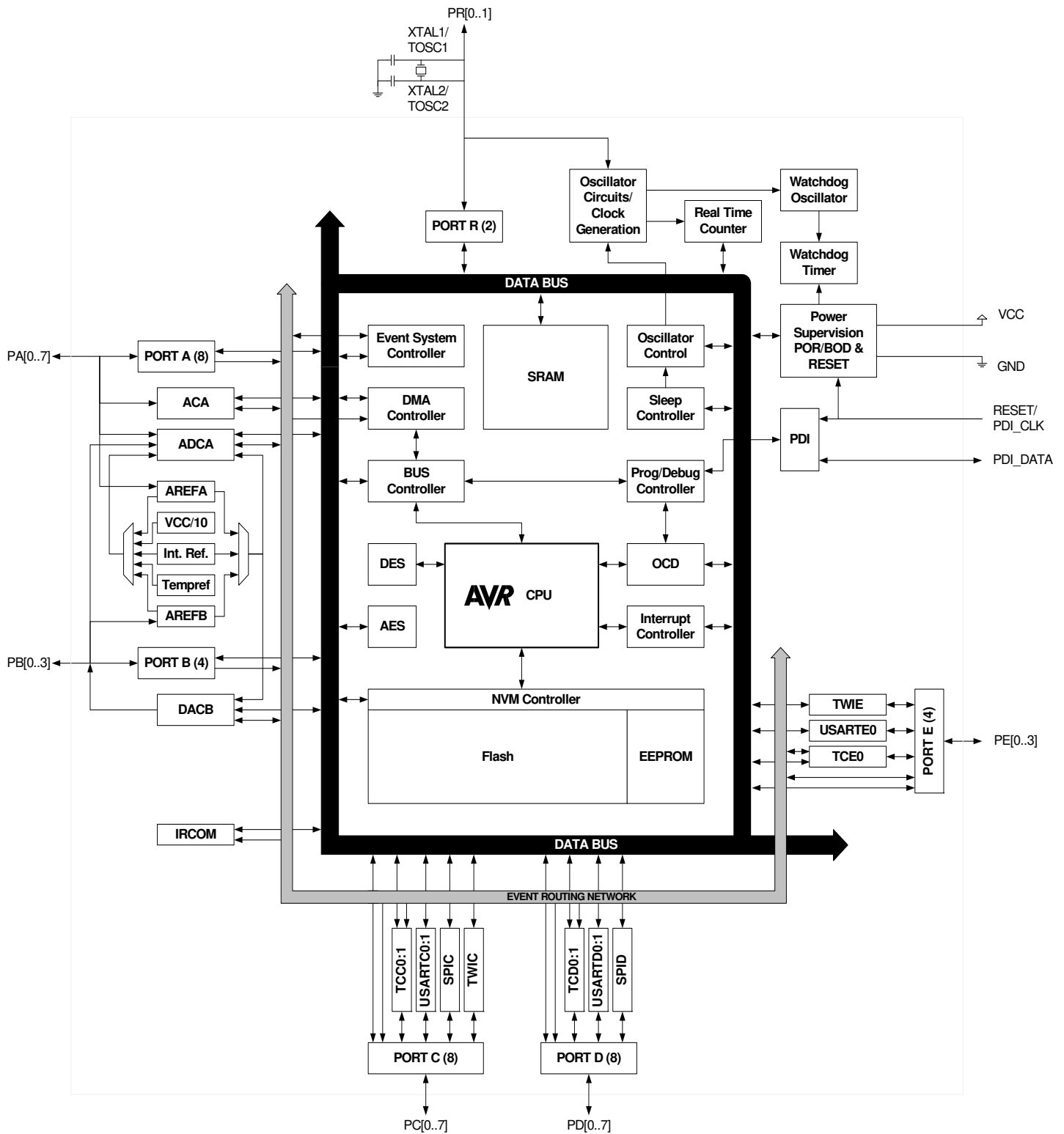
The XMEGA A4 devices have five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, DMA Controller, Event System, Interrupt Controller and all peripherals to continue functioning. The Power-down mode saves the SRAM and register contents but stops the oscillators, disabling all other functions until the next TWI or pin-change interrupt, or Reset. In Power-save mode, the asynchronous Real Time Counter continues to run, allowing the application to maintain a timer base while the rest of the device is sleeping. In Standby mode, the Crystal/Resonator Oscillator is kept running while the rest of the device is sleeping. This allows very fast start-up from external crystal combined with low power consumption. In Extended Standby mode, both the main Oscillator and the Asynchronous Timer continue to run. To further reduce power consumption, the peripheral clock to each individual peripheral can optionally be stopped in Active mode and in Idle sleep mode.

The device is manufactured using Atmel's high-density nonvolatile memory technology. The program Flash memory can be reprogrammed in-system through the PDI. A Bootloader running in the device can use any interface to download the application program to the Flash memory. The Bootloader 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 8/16-bit RISC CPU with In-System Self-Programmable Flash, the Atmel XMEGA A4 is a powerful microcontroller family that provides a highly flexible and cost effective solution for many embedded applications.

The XMEGA A4 devices are supported with a full suite of program and system development tools including: C compilers, macro assemblers, program debugger/simulators, programmers, and evaluation kits.

3.1 Block Diagram

Figure 3-1. XMEGA A4 Block Diagram



4. Resources

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

4.1 Recommended reading

- Atmel AVR XMEGA A Manual
- XMEGA A Application Notes

This device data sheet only contains part specific information and a short description of each peripheral and module. The XMEGA A Manual describes the modules and peripherals in depth. The XMEGA A application notes contain example code and show applied use of the modules and peripherals.

The XMEGA A Manual and Application Notes are available from <http://www.atmel.com/avr>.

5. Disclaimer

For devices that are not available yet, typical values contained in this datasheet are based on simulations and characterization of other AVR XMEGA microcontrollers manufactured on the same process technology. Min. and Max values will be available after the device is characterized.

6. AVR CPU

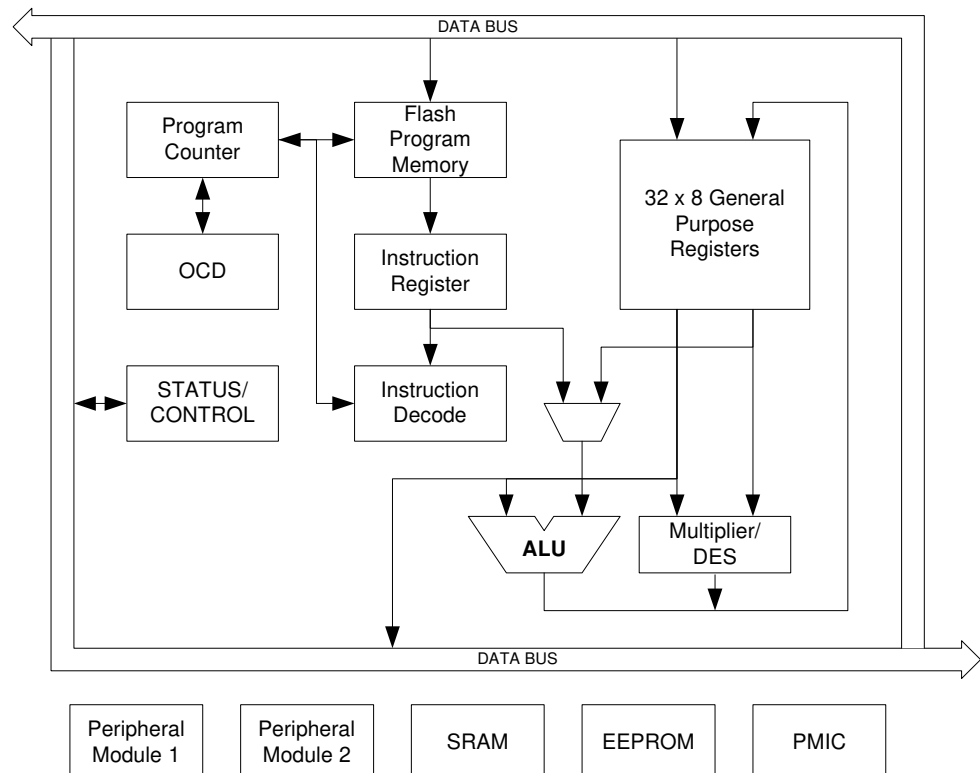
6.1 Features

- 8/16-bit high performance AVR RISC Architecture
 - 138 instructions
 - Hardware multiplier
- 32x8-bit registers directly connected to the ALU
- Stack in RAM
- Stack Pointer accessible in I/O memory space
- Direct addressing of up to 16M Bytes of program and data memory
- True 16/24-bit access to 16/24-bit I/O registers
- Support for 8-, 16- and 32-bit Arithmetic
- Configuration Change Protection of system critical features

6.2 Overview

The XMEGA A4 uses the 8/16-bit AVR CPU. The main function of the CPU is program execution. The CPU must therefore be able to access memories, perform calculations and control peripherals. Interrupt handling is described in a separate section. [Figure 6-1 on page 8](#) shows the CPU block diagram.

Figure 6-1. CPU block diagram



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 pipeline. 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 Re-programmable Flash memory.

6.3 Register File

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

Six of the 32 registers can be used as three 16-bit 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.

6.4 ALU - Arithmetic Logic Unit

The high performance Arithmetic Logic Unit (ALU) supports arithmetic and logic operations between registers or between a constant and a register. Single register operations can also be executed. Within a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed. After an arithmetic or logic operation, the Status Register is updated to reflect information about the result of the operation.

The ALU operations are divided into three main categories – arithmetic, logical, and bit-functions. Both 8- and 16-bit arithmetic is supported, and the instruction set allows for efficient implementation of 32-bit arithmetic. The ALU also provides a powerful multiplier supporting both signed and unsigned multiplication and fractional format.

6.5 Program Flow

When the device is powered on, the CPU starts to execute instructions from the lowest address in the Flash Program Memory '0'. The Program Counter (PC) addresses the next instruction to be fetched. After a reset, the PC is set to location '0'.

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

During interrupts and subroutine calls, the return address 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. After reset the Stack Pointer (SP) points to the highest address in the internal SRAM. The SP is read/write accessible in the I/O memory space, enabling easy implementation of multiple stacks or stack areas. The data SRAM can easily be accessed through the five different addressing modes supported in the AVR CPU.

7. Memories

7.1 Features

- **Flash Program Memory**
 - One linear address space
 - In-System Programmable
 - Self-Programming and Bootloader support
 - Application Section for application code
 - Application Table Section for application code or data storage
 - Boot Section for application code or bootloader code
 - Separate lock bits and protection for all sections
 - Built in fast CRC check of a selectable flash program memory section
- **Data Memory**
 - One linear address space
 - Single cycle access from CPU
 - SRAM
 - EEPROM
 - Byte and page accessible
 - Optional memory mapping for direct load and store
 - I/O Memory
 - Configuration and Status registers for all peripherals and modules
 - 16 bit-accessible General Purpose Register for global variables or flags
 - Bus arbitration
 - Safe and deterministic handling of CPU and DMA Controller priority
 - Separate buses for SRAM, EEPROM, I/O Memory and External Memory access
 - Simultaneous bus access for CPU and DMA Controller
- **Production Signature Row Memory for factory programmed data**
 - Device ID for each microcontroller device type
 - Serial number for each device
 - Oscillator calibration bytes
 - ADC, DAC and temperature sensor calibration data
- **User Signature Row**
 - One flash page in size
 - Can be read and written from software
 - Content is kept after chip erase

7.2 Overview

The AVR architecture has two main memory spaces, the Program Memory and the Data Memory. In addition, the XMEGA A4 features an EEPROM Memory for non-volatile data storage. All three memory spaces are linear and require no paging. The available memory size configurations are shown in ["Ordering Information" on page 2](#). In addition each device has a Flash memory signature row for calibration data, device identification, serial number etc.

Non-volatile memory spaces can be locked for further write or read/write operations. This prevents unrestricted access to the application software.

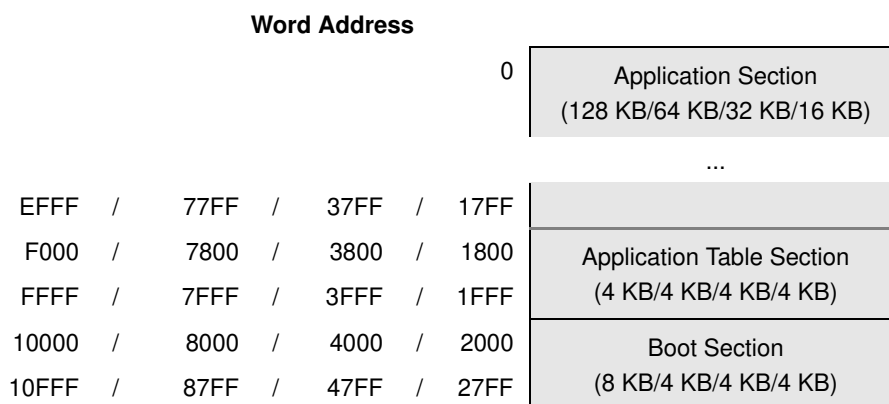
7.3 In-System Programmable Flash Program Memory

The XMEGA A4 devices contain On-chip In-System Programmable Flash memory for program storage, see [Figure 7-1 on page 11](#). Since all AVR instructions are 16- or 32-bits wide, each Flash address location is 16 bits.

The Program Flash memory space is divided into Application and Boot sections. Both sections have dedicated Lock Bits for setting restrictions on write or read/write operations. The Store Program Memory (SPM) instruction must reside in the Boot Section when used to write to the Flash memory.

A third section inside the Application section is referred to as the Application Table section which has separate Lock bits for storage of write or read/write protection. The Application Table section can be used for storing non-volatile data or application software.

Figure 7-1. Flash Program Memory (Hexadecimal address)



The Application Table Section and Boot Section can also be used for general application software.

7.4 Data Memory

The Data Memory consist of the I/O Memory, EEPROM and SRAM memories, all within one linear address space, see [Figure 7-2 on page 12](#). To simplify development, the memory map for all devices in the family is identical and with empty, reserved memory space for smaller devices.

Figure 7-2. Data Memory Map (Hexadecimal address)

Byte Address	ATxmega64A4	Byte Address	ATxmega32A4	Byte Address	ATxmega16A4
0	I/O Registers (4 KB)	0	I/O Registers (4 KB)	0	I/O Registers (4 KB)
FFF		FFF		FFF	
1000	EEPROM (2 KB)	1000	EEPROM (1 KB)	1000	EEPROM (1 KB)
17FF		13FF		13FF	
	RESERVED		RESERVED		RESERVED
2000	Internal SRAM (4 KB)	2000	Internal SRAM (4 KB)	2000	Internal SRAM (2 KB)
2FFF		2FFF		27FF	

Byte Address	ATxmega128A4
0	I/O Registers (4 KB)
FFF	
1000	EEPROM (2 KB)
17FF	
	RESERVED
2000	Internal SRAM (8 KB)
3FFF	

7.4.1 I/O Memory

All peripherals and modules are addressable through I/O memory locations in the data memory space. All I/O memory locations can be accessed by the Load (LD/LDS/LDD) and Store (ST/STS/STD) instructions, transferring data between the 32 general purpose registers in the CPU and the I/O Memory.

The IN and OUT instructions can address I/O memory locations in the range 0x00 - 0x3F directly.

I/O registers within the address range 0x00 - 0x1F are directly bit-accessible using the SBI and CBI instructions. The value of single bits can be checked by using the SBIS and SBIC instructions on these registers.

The I/O memory address for all peripherals and modules in XMEGA A4 is shown in the "[Peripheral Module Address Map](#)" on page 53.

7.4.2 SRAM Data Memory

The XMEGA A4 devices have internal SRAM memory for data storage.

7.4.3 EEPROM Data Memory

The XMEGA A4 devices have internal EEPROM memory for non-volatile data storage. It is addressable either in a separate data space or it can be memory mapped into the normal data memory space. The EEPROM memory supports both byte and page access.

7.5 Production Signature Row

The Production Signature Row is a separate memory section for factory programmed data. It contains calibration data for functions such as oscillators and analog modules.

The production signature row also contains a device ID that identify each microcontroller device type, and a serial number that is unique for each manufactured device. The device ID for the available XMEGA A4 devices is shown in [Table 7-1 on page 13](#). The serial number consist of the production LOT number, wafer number, and wafer coordinates for the device.

The production signature row can not be written or erased, but it can be read from both application software and external programming.

Table 7-1. Device ID bytes for XMEGA A4 devices.

Device	Device ID bytes		
	Byte 2	Byte 1	Byte 0
ATxmega16A4	41	94	1E
ATxmega32A4	41	95	1E
ATxmega64A4	46	96	1E
ATxmega128A4	46	97	1E

7.6 User Signature Row

The User Signature Row is a separate memory section that is fully accessible (read and write) from application software and external programming. The user signature row is one flash page in size, and is meant for static user parameter storage, such as calibration data, custom serial numbers or identification numbers, random number seeds etc. This section is not erased by Chip Erase commands that erase the Flash, and requires a dedicated erase command. This ensures parameter storage during multiple program/erase session and on-chip debug sessions.

7.7 Flash and EEPROM Page Size

The Flash Program Memory and EEPROM data memory are organized in pages. The pages are word accessible for the Flash and byte accessible for the EEPROM.

Table 7-2 on page 14 shows the Flash Program Memory organization. Flash write and erase operations are performed on one page at a time, while reading the Flash is done one byte at a time. For Flash access the Z-pointer (Z[m:n]) is used for addressing. The most significant bits in the address (FPAGE) give the page number and the least significant address bits (FWORD) give the word in the page.

Table 7-2. Number of words and Pages in the Flash.

Devices	Flash Size	Page Size (words)	FWORD	FPAGE	Application		Boot	
					Size	No of Pages	Size	No of Pages
ATxmega16A4	16 KB + 4 KB	128	Z[6:0]	Z[13:7]	16 KB	64	4 KB	16
ATxmega32A4	32 KB + 4 KB	128	Z[6:0]	Z[14:7]	32 KB	128	4 KB	16
ATxmega64A4	64 KB + 4 KB	128	Z[6:0]	Z[15:7]	64 KB	128	4 KB	16
ATxmega128A4	128 KB + 8 KB	256	Z[7:0]	Z[16:8]	128 KB	256	8 KB	16

Table 7-3 on page 14 shows EEPROM memory organization for the XMEGA A4 devices. EEPROM write and erase operations can be performed one page or one byte at a time, while reading the EEPROM is done one byte at a time. For EEPROM access the NVM Address Register (ADDR[m:n]) is used for addressing. The most significant bits in the address (E2PAGE) give the page number and the least significant address bits (E2BYTE) give the byte in the page.

Table 7-3. Number of Bytes and Pages in the EEPROM.

Devices	EEPROM Size	Page Size (Bytes)	E2BYTE	E2PAGE	No of Pages
ATxmega16A4	1 KB	32	ADDR[4:0]	ADDR[10:5]	32
ATxmega32A4	1 KB	32	ADDR[4:0]	ADDR[10:5]	32
ATxmega64A4	2 KB	32	ADDR[4:0]	ADDR[10:5]	64
ATxmega128A4	2 KB	32	ADDR[4:0]	ADDR[10:5]	64

8. DMAC - Direct Memory Access Controller

8.1 Features

- **Allows High-speed data transfer**
 - From memory to peripheral
 - From memory to memory
 - From peripheral to memory
 - From peripheral to peripheral
- **4 Channels**
- **From 1 byte and up to 16 M bytes transfers in a single transaction**
- **Multiple addressing modes for source and destination address**
 - Increment
 - Decrement
 - Static
- **1, 2, 4, or 8 bytes Burst Transfers**
- **Programmable priority between channels**

8.2 Overview

The XMEGA A4 has a Direct Memory Access (DMA) Controller to move data between memories and peripherals in the data space. The DMA controller uses the same data bus as the CPU to transfer data.

It has 4 channels that can be configured independently. Each DMA channel can perform data transfers in blocks of configurable size from 1 to 64K bytes. A repeat counter can be used to repeat each block transfer for single transactions up to 16M bytes. Each DMA channel can be configured to access the source and destination memory address with incrementing, decrementing or static addressing. The addressing is independent for source and destination address. When the transaction is complete the original source and destination address can automatically be reloaded to be ready for the next transaction.

The DMAC can access all the peripherals through their I/O memory registers, and the DMA may be used for automatic transfer of data to/from communication modules, as well as automatic data retrieval from ADC conversions, data transfer to DAC conversions, or data transfer to or from port pins. A wide range of transfer triggers is available from the peripherals, Event System and software. Each DMA channel has different transfer triggers.

To allow for continuous transfer, two channels can be interlinked so that the second takes over the transfer when the first is finished and vice versa.

The DMA controller can read from memory mapped EEPROM, but it cannot write to the EEPROM or access the Flash.

9. Event System

9.1 Features

- Inter-peripheral communication and signalling with minimum latency
- CPU and DMA independent operation
- 8 Event Channels allow for up to 8 signals to be routed at the same time
- Events can be generated by
 - Timer/Counters (TCxn)
 - Real Time Counter (RTC)
 - Analog to Digital Converters (ADCx)
 - Analog Comparators (ACx)
 - Ports (PORTx)
 - System Clock (Clk_{sys})
 - Software (CPU)
- Events can be used by
 - Timer/Counters (TCxn)
 - Analog to Digital Converters (ADCx)
 - Digital to Analog Converters (DACx)
 - Ports (PORTx)
 - DMA Controller (DMAC)
 - IR Communication Module (IRCOM)
- The same event can be used by multiple peripherals for synchronized timing
- Advanced Features
 - Manual Event Generation from software (CPU)
 - Quadrature Decoding
 - Digital Filtering
- Functions in Active and Idle mode

9.2 Overview

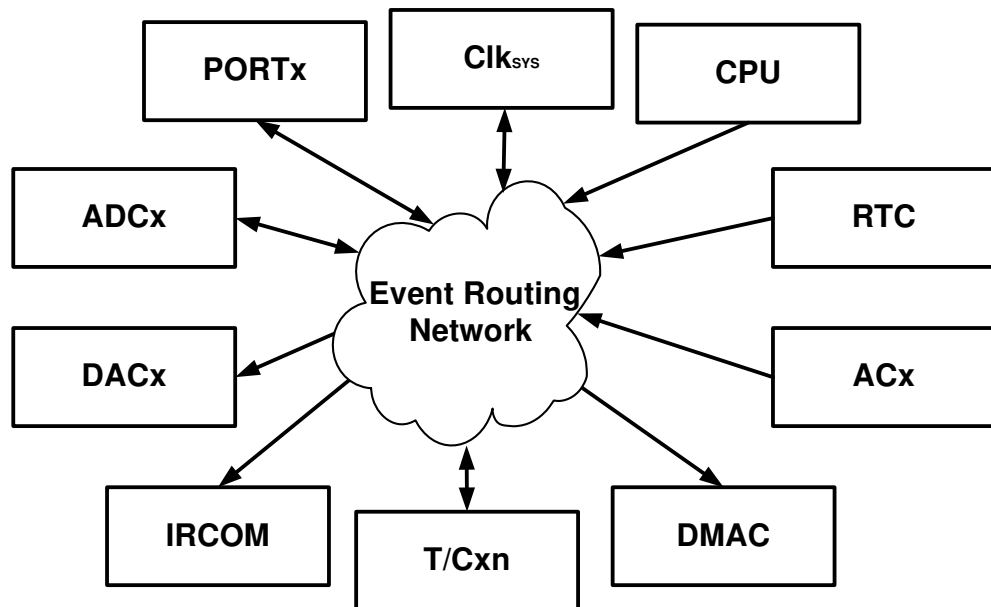
The Event System is a set of features for inter-peripheral communication. It enables the possibility for a change of state in one peripheral to automatically trigger actions in one or more peripherals. Whose changes in a peripheral that will trigger actions in other peripherals are configurable by software. It is a simple, but powerful system as it allows for autonomous control of peripherals without any use of interrupts, CPU or DMA resources.

The indication of a change in a peripheral is referred to as an event, and is usually the same as the interrupt conditions for that peripheral. Events are passed between peripherals using a dedicated routing network called the Event Routing Network. [Figure 9-1 on page 17](#) shows a basic block diagram of the Event System with the Event Routing Network and the peripherals to which it is connected. This highly flexible system can be used for simple routing of signals, pin functions or for sequencing of events.

The maximum latency is two CPU clock cycles from when an event is generated in one peripheral, until the actions are triggered in one or more other peripherals.

The Event System is functional in both Active and Idle modes.

Figure 9-1. Event System Block Diagram



The Event Routing Network can directly connect together ADCs, DACs, Analog Comparators (ACx), I/O ports (PORTx), the Real-time Counter (RTC), Timer/Counters (T/C) and the IR Communication Module (IRCOM). Events can also be generated from software (CPU).

All events from all peripherals are always routed into the Event Routing Network. This consist of eight multiplexers where each can be configured in software to select which event to be routed into that event channel. All eight event channels are connected to the peripherals that can use events, and each of these peripherals can be configured to use events from one or more event channels to automatically trigger a software selectable action.

10. System Clock and Clock options

10.1 Features

- **Fast start-up time**
- **Safe run-time clock switching**
- **Internal Oscillators:**
 - 32 MHz run-time calibrated RC oscillator
 - 2 MHz run-time calibrated RC oscillator
 - 32.768 kHz calibrated RC oscillator
 - 32 kHz Ultra Low Power (ULP) oscillator with 1 kHz output
- **External clock options**
 - 0.4 - 16 MHz Crystal Oscillator
 - 32 kHz Crystal Oscillator
 - External clock
- **PLL with internal and external clock options with 1 to 31x multiplication**
- **Clock Prescalers with 1 to 2048x division**
- **Fast peripheral clock running at 2 and 4 times the CPU clock speed**
- **Automatic Run-Time Calibration of internal oscillators**
- **Crystal Oscillator failure detection**

10.2 Overview

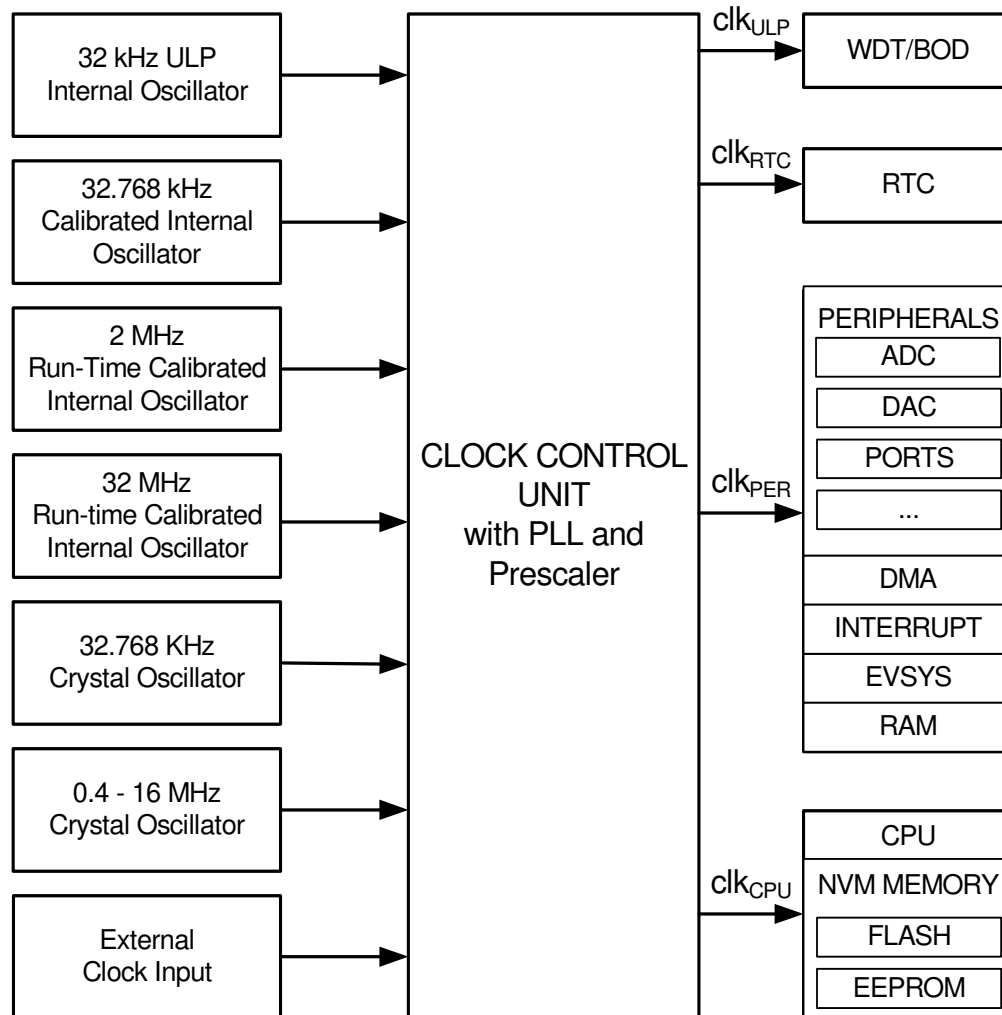
XMEGA A4 has an advanced clock system, supporting a large number of clock sources. It incorporates both integrated oscillators, external crystal oscillators and resonators. A high frequency Phase Locked Loop (PLL) and clock prescalers can be controlled from software to generate a wide range of clock frequencies from the clock source input.

It is possible to switch between clock sources from software during run-time. After reset the device will always start up running from the 2 Mhz internal oscillator.

A calibration feature is available, and can be used for automatic run-time calibration of the internal 2 MHz and 32 MHz oscillators. This reduce frequency drift over voltage and temperature.

A Crystal Oscillator Failure Monitor can be enabled to issue a Non-Maskable Interrupt and switch to internal oscillator if the external oscillator fails. [Figure 10-1 on page 19](#) shows the principal clock system in XMEGA A4.

Figure 10-1. Clock system overview



Each clock source is briefly described in the following sub-sections.

10.3 Clock Options

10.3.1 32 kHz Ultra Low Power Internal Oscillator

The 32 kHz Ultra Low Power (ULP) Internal Oscillator is a very low power consumption clock source. It is used for the Watchdog Timer, Brown-Out Detection and as an asynchronous clock source for the Real Time Counter. This oscillator cannot be used as the system clock source, and it cannot be directly controlled from software.

10.3.2 32.768 kHz Calibrated Internal Oscillator

The 32.768 kHz Calibrated Internal Oscillator is a high accuracy clock source that can be used as the system clock source or as an asynchronous clock source for the Real Time Counter. It is calibrated during production to provide a default frequency which is close to its nominal frequency.

10.3.3 32.768 kHz Crystal Oscillator

The 32.768 kHz Crystal Oscillator is a low power driver for an external watch crystal. It can be used as system clock source or as asynchronous clock source for the Real Time Counter.

10.3.4 0.4 - 16 MHz Crystal Oscillator

The 0.4 - 16 MHz Crystal Oscillator is a driver intended for driving both external resonators and crystals ranging from 400 kHz to 16 MHz.

10.3.5 2 MHz Run-time Calibrated Internal Oscillator

The 2 MHz Run-time Calibrated Internal Oscillator is a high frequency oscillator. It is calibrated during production to provide a default frequency which is close to its nominal frequency. The oscillator can use the 32 kHz Calibrated Internal Oscillator or the 32 kHz Crystal Oscillator as a source for calibrating the frequency run-time to compensate for temperature and voltage drift hereby optimizing the accuracy of the oscillator.

10.3.6 32 MHz Run-time Calibrated Internal Oscillator

The 32 MHz Run-time Calibrated Internal Oscillator is a high frequency oscillator. It is calibrated during production to provide a default frequency which is close to its nominal frequency. The oscillator can use the 32 kHz Calibrated Internal Oscillator or the 32 kHz Crystal Oscillator as a source for calibrating the frequency run-time to compensate for temperature and voltage drift hereby optimizing the accuracy of the oscillator.

10.3.7 External Clock input

The external clock input gives the possibility to connect a clock from an external source.

10.3.8 PLL with Multiplication factor 1 - 31x

The PLL provides the possibility of multiplying a frequency by any number from 1 to 31. In combination with the prescalers, this gives a wide range of output frequencies from all clock sources.

11. Power Management and Sleep Modes

11.1 Features

- 5 sleep modes
 - Idle
 - Power-down
 - Power-save
 - Standby
 - Extended standby
- Power Reduction registers to disable clocks to unused peripherals

11.2 Overview

The XMEGA A4 provides various sleep modes tailored to reduce power consumption to a minimum. All sleep modes are available and can be entered from Active mode. In Active mode the CPU is executing application code. The application code decides when and what sleep mode to enter. 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 mode.

11.3 Sleep Modes

11.3.1 Idle Mode

In Idle mode the CPU and Non-Volatile Memory are stopped, but all peripherals including the Interrupt Controller, Event System and DMA Controller are kept running. Interrupt requests from all enabled interrupts will wake the device.

11.3.2 Power-down Mode

In Power-down mode all system clock sources, and the asynchronous Real Time Counter (RTC) clock source, are stopped. This allows operation of asynchronous modules only. The only interrupts that can wake up the MCU are the Two Wire Interface address match interrupts, and asynchronous port interrupts, e.g pin change.

11.3.3 Power-save Mode

Power-save mode is identical to Power-down, with one exception: If the RTC is enabled, it will keep running during sleep and the device can also wake up from RTC interrupts.

11.3.4 Standby Mode

Standby mode is identical to Power-down with the exception that all enabled system clock sources are kept running, while the CPU, Peripheral and RTC clocks are stopped. This reduces the wake-up time when external crystals or resonators are used.

11.3.5 Extended Standby Mode

Extended Standby mode is identical to Power-save mode with the exception that all enabled system clock sources are kept running while the CPU and Peripheral clocks are stopped. This reduces the wake-up time when external crystals or resonators are used.

12. System Control and Reset

12.1 Features

- **Multiple reset sources for safe operation and device reset**
 - Power-On Reset
 - External Reset
 - Watchdog Reset
 - The Watchdog Timer runs from separate, dedicated oscillator
 - Brown-Out Reset
 - Accurate, programmable Brown-Out levels
 - PDI reset
 - Software reset
- **Asynchronous reset**
 - No running clock in the device is required for reset
- **Reset status register**

12.2 Resetting the AVR

During reset, all I/O registers are set to their initial values. The SRAM content is not reset. Application execution starts from the Reset Vector. The instruction placed at the Reset Vector should be an Absolute Jump (JMP) instruction to the reset handling routine. By default the Reset Vector address is the lowest Flash program memory address, '0', but it is possible to move the Reset Vector to the first address in the Boot Section.

The I/O ports of the AVR are immediately tri-stated when a reset source goes active.

The reset functionality is asynchronous, so no running clock is required to reset the device.

After the device is reset, the reset source can be determined by the application by reading the Reset Status Register.

12.3 Reset Sources

12.3.1 Power-On Reset

The MCU is reset when the supply voltage VCC is below the Power-on Reset threshold voltage.

12.3.2 External Reset

The MCU is reset when a low level is present on the RESET pin.

12.3.3 Watchdog Reset

The MCU is reset when the Watchdog Timer period expires and the Watchdog Reset is enabled. The Watchdog Timer runs from a dedicated oscillator independent of the System Clock. For more details see ["WDT - Watchdog Timer" on page 24](#).

12.3.4 Brown-Out Reset

The MCU is reset when the supply voltage VCC is below the Brown-Out Reset threshold voltage and the Brown-out Detector is enabled. The Brown-out threshold voltage is programmable.

12.3.5 PDI reset

The MCU can be reset through the Program and Debug Interface (PDI).

12.3.6 Software reset

The MCU can be reset by the CPU writing to a special I/O register through a timed sequence.

13. WDT - Watchdog Timer

13.1 Features

- 11 selectable timeout periods, from 8 ms to 8s.
- Two operation modes
 - Standard mode
 - Window mode
- Runs from the 1 kHz output of the 32 kHz Ultra Low Power oscillator
- Configuration lock to prevent unwanted changes

13.2 Overview

The XMEGA A4 has a Watchdog Timer (WDT). The WDT will run continuously when turned on and if the Watchdog Timer is not reset within a software configurable time-out period, the microcontroller will be reset. The Watchdog Reset (WDR) instruction must be run by software to reset the WDT, and prevent microcontroller reset.

The WDT has a Window mode. In this mode the WDR instruction must be run within a specified period called a window. Application software can set the minimum and maximum limits for this window. If the WDR instruction is not executed inside the window limits, the microcontroller will be reset.

A protection mechanism using a timed write sequence is implemented in order to prevent unwanted enabling, disabling or change of WDT settings.

For maximum safety, the WDT also has an Always-on mode. This mode is enabled by programming a fuse. In Always-on mode, application software can not disable the WDT.

14. PMIC - Programmable Multi-level Interrupt Controller

14.1 Features

- Separate interrupt vector for each interrupt
- Short, predictable interrupt response time
- Programmable Multi-level Interrupt Controller
 - 3 programmable interrupt levels
 - Selectable priority scheme within low level interrupts (round-robin or fixed)
 - Non-Maskable Interrupts (NMI)
- Interrupt vectors can be moved to the start of the Boot Section

14.2 Overview

XMEGA A4 has a Programmable Multi-level Interrupt Controller (PMIC). All peripherals can define three different priority levels for interrupts; high, medium or low. Medium level interrupts may interrupt low level interrupt service routines. High level interrupts may interrupt both low- and medium level interrupt service routines. Low level interrupts have an optional round robin scheme to make sure all interrupts are serviced within a certain amount of time.

The built in oscillator failure detection mechanism can issue a Non-Maskable Interrupt (NMI).

14.3 Interrupt vectors

When an interrupt is serviced, the program counter will jump to the interrupt vector address. The interrupt vector is the sum of the peripheral’s base interrupt address and the offset address for specific interrupts in each peripheral. The base addresses for the XMEGA A4 devices are shown in [Table 14-1](#). Offset addresses for each interrupt available in the peripheral are described for each peripheral in the XMEGA A manual. For peripherals or modules that have only one interrupt, the interrupt vector is shown in [Table 14-1](#). The program address is the word address.

Table 14-1. Reset and Interrupt Vectors

Program Address (Base Address)	Source	Interrupt Description
0x000	RESET	
0x002	OSCF_INT_vect	Crystal Oscillator Failure Interrupt vector (NMI)
0x004	PORTC_INT_base	Port C Interrupt base
0x008	PORTR_INT_base	Port R Interrupt base
0x00C	DMA_INT_base	DMA Controller Interrupt base
0x014	RTC_INT_base	Real Time Counter Interrupt base
0x018	TWIC_INT_base	Two-Wire Interface on Port C Interrupt base
0x01C	TCC0_INT_base	Timer/Counter 0 on port C Interrupt base
0x028	TCC1_INT_base	Timer/Counter 1 on port C Interrupt base
0x030	SPIC_INT_vect	SPI on port C Interrupt vector
0x032	USARTC0_INT_base	USART 0 on port C Interrupt base
0x038	USARTC1_INT_base	USART 1 on port C Interrupt base
0x03E	AES_INT_vect	AES Interrupt vector