Atmel®

Not recommended for new designs -

Use XMEGA A1U series

8/16-bit XMEGA A1 Microcontroller

ATxmega128A1 / ATxmega64A1

Preliminary

Features

- High-performance, low-power Atmel® AVR® XMEGA® 8/16-bit Microcontroller
- Nonvolatile program and data memories
  - 64K - 128KBytes of in-system self-programmable flash
  - 4K - 8KBytes boot section
  - 2 KBBytes EEPROM
  - 4 KB - 8 KBBytes internal SRAM
    - External bus interface for up to 16Mbytes SRAM
    - External bus interface for up to 128Mbit SDRAM
- Peripheral features
  - Four-channel DMA controller
  - Eight-channel event system
  - Eight 16-bit timer/counters
    - Four timer/counters with 4 output compare or input capture channels
    - Four timer/counters with 2 output compare or input capture channels
    - High resolution extension on all timer/counters
    - Advanced waveform extension (AWeX) on two timer/counters
  - Eight USARTs with IrDA support for one USART
  - Four two-wire interfaces with dual address match (I^2C and SMBus compatible)
  - Four serial peripheral interfaces (SPIs)
  - AES and DES crypto engine
  - 16-bit real time counter (RTC) with separate oscillator
  - Two sixteen channel, 12-bit, 2msps Analog to Digital Converters
  - Two two-channel, 12-bit, 1msps Digital to Analog Converters
  - Four Analog Comparators (ACs) with window compare function, and current sources
  - External interrupts on all general purpose I/O pins
  - Programmable watchdog timer with separate on-chip ultra low power oscillator
  - QTouch® library support
    - Capacitive touch buttons, sliders and wheels
- Special microcontroller features
  - Power-on reset and programmable brown-out detection
  - Internal and external clock options with PLL and prescaler
  - Programmable multilevel interrupt controller
  - Five sleep modes
  - Programming and debug interfaces
    - JTAG (IEEE 1149.1 compliant) interface, including boundary scan
    - PDI (Program and Debug Interface)
- I/O and packages
  - 78 Programmable I/O pins
  - 100 lead TQFP
  - 100 ball BGA
  - 100 ball VFBGA
- Operating voltage
  - 1.6 – 3.6V
- Operating frequency
  - 0 – 12MHz from 1.6V
  - 0 – 32MHz from 2.7V

8067O–AVR–06/2013

1. Ordering Information

Ordering Code	Flash (B)	E^2	SRAM	Speed (MHz)	Power Supply	Package[1][2][3]	Temp
ATxmega128A1-AU	128K + 8K	2 KB	8 KB	32	1.6 - 3.6V	100A	-40°C - 85°C
ATxmega128A1-AUR							
ATxmega64A1-AU	64K + 4K	2 KB	4 KB				
ATxmega64A1-AUR							
ATxmega128A1-CU	128K + 8K	2 KB	8 KB			100C1	
ATxmega128A1CUR							
ATxmega64A1-CU	64K + 4K	2 KB	4 KB				
ATxmega64A1-CUR							
ATxmega128A1-C7U	128K + 8K	2 KB	8 KB			100C2	
ATxmega128A1-C7UR							
ATxmega64A1-C7U	64K + 4K	2 KB	4 KB				
ATxmega64A1-C7UR							

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 70.

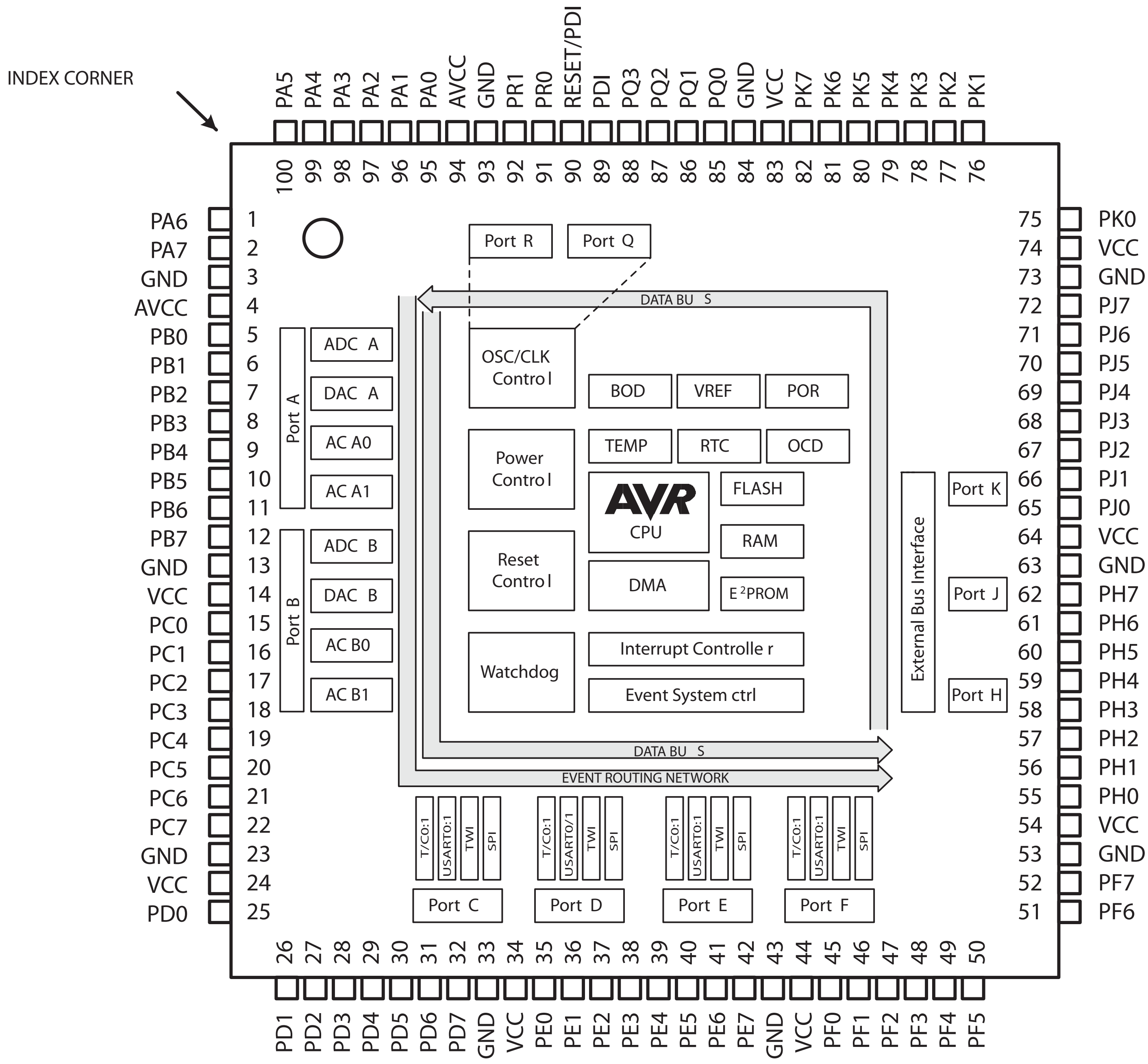
Package Type	
100A	100-lead, 14 x 14 x 1.0mm, 0.5mm lead pitch, thin profile plastic quad flat package (TQFP)
100C1	100-ball, 9 x 9 x 1.2mm body, ball pitch 0.88mm, chip ball grid array (CBGA)
100C2	100-ball, 7 x 7 x 1.0mm body, ball pitch 0.65mm, very thin fine-pitch ball grid array (VFBGA)

Typical Applications

Industrial control	Climate control	Low power battery applications
Factory automation	RF and ZigBee®	Power tools
Building control	Sensor control	HVAC
Board control	Optical	Utility metering
White goods	Medical applications	

2. Pinout/Block Diagram

Figure 2-1. Block diagram and pinout



Notes:
1. For full details on pinout and pin functions refer to "Pinout and Pin Functions" on page 55.
2. VCC/GND on pin 83/84 are swapped compared to other VCC/GND to allow easier routing of GND to 32kHz crystal.

Figure 2-2. CBGA-pinout

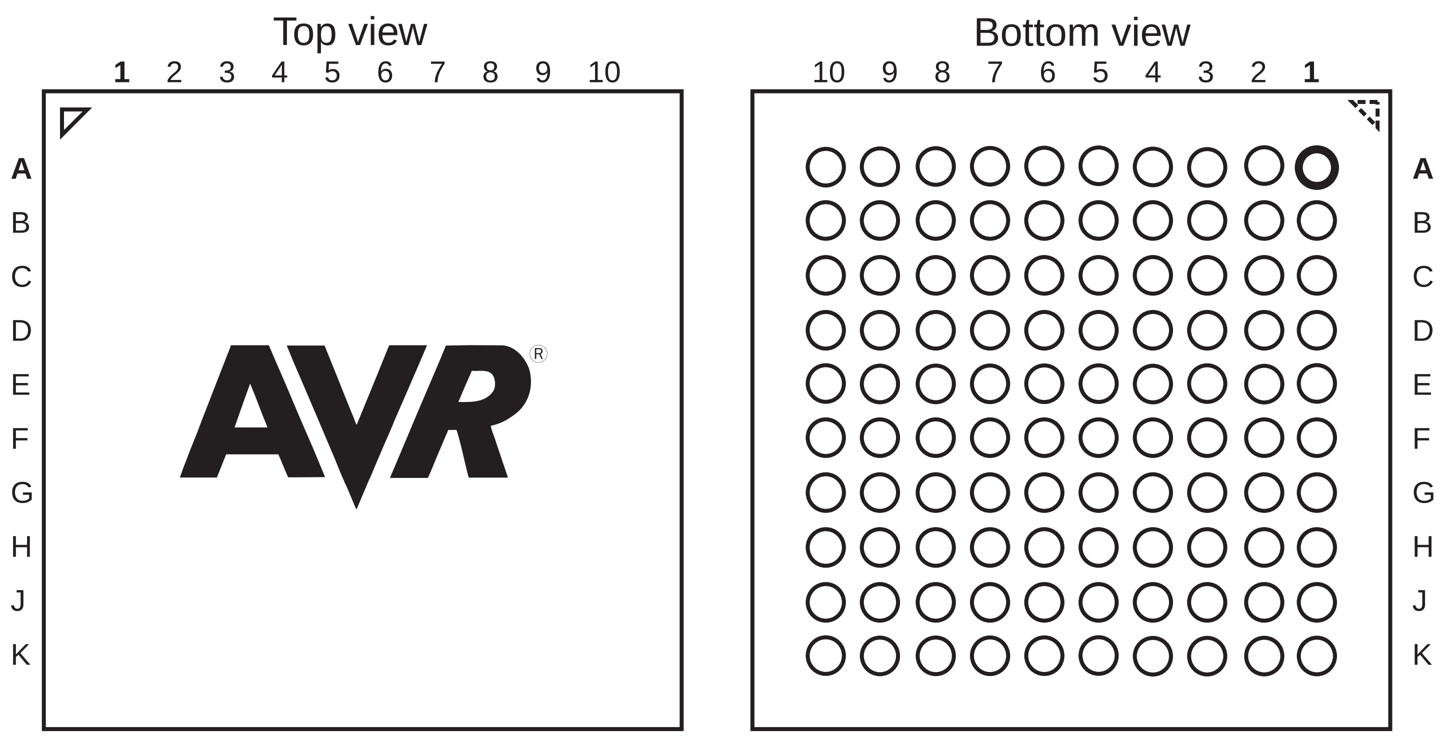


Table 2-1. CBGA-pinout.

	1	2	3	4	5	6	7	8	9	10
A	PK0	VCC	GND	PJ3	VCC	GND	PH1	GND	VCC	PF7
B	PK3	PK2	PK1	PJ4	PH7	PH4	PH2	PH0	PF6	PF5
C	VCC	PK5	PK4	PJ5	PJ0	PH5	PH3	PF2	PF3	VCC
D	GND	PK6	PK7	PJ6	PJ1	PH6	PF0	PF1	PF4	GND
E	PQ0	PQ1	PQ2	PJ7	PJ2	PE7	PE6	PE5	PE4	PE3
F	PR1	PR0	RESET/ PDI	PDI	PQ3	PC2	PE2	PE1	PE0	VCC
G	GND	PA1	PA4	PB3	PB4	PC1	PC6	PD7	PD6	GND
H	AVCC	PA2	PA5	PB2	PB5	PC0	PC5	PD5	PD4	PD3
J	PA0	PA3	PB0	PB1	PB6	PC3	PC4	PC7	PD2	PD1
K	PA6	PA7	GND	AVCC	PB7	VCC	GND	VCC	GND	PD0

3. Overview

The Atmel AVR XMEGA is a family of low power, high performance, and peripheral rich 8/16-bit microcontrollers based on the AVR enhanced RISC architecture. By executing instructions in a single clock cycle, the AVR XMEGA devices achieve CPU throughput approaching one million instructions per second (MIPS) per megahertz, allowing the system designer to optimize power consumption versus processing speed.

The Atmel AVR CPU combines a rich instruction set with 32 general purpose working registers. All 32 registers are directly connected to the arithmetic logic unit (ALU), allowing two independent registers to be accessed in a 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 AVR XMEGA A1 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 and programmable multilevel interrupt controller, 78 general purpose I/O lines, 16-bit real-time counter (RTC); eight flexible, 16-bit timer/counters with compare and PWM channels, eight USARTs; four two-wire serial interfaces (TWIs); four serial peripheral interfaces (SPIs); AES and DES cryptographic engine; two 16-channel, 12-bit ADCs with programmable gain; two 2-channel, 12-bit DACs; four Analog Comparators (ACs) 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, two-pin interface for programming and debugging, is available. The devices also have an IEEE std. 1149.1 compliant JTAG interface, and this can also be used for boundary scan, on-chip debug and programming.

The XMEGA A1 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 external crystal oscillator keeps running while the rest of the device is sleeping. This allows very fast startup from the 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 idle sleep mode.

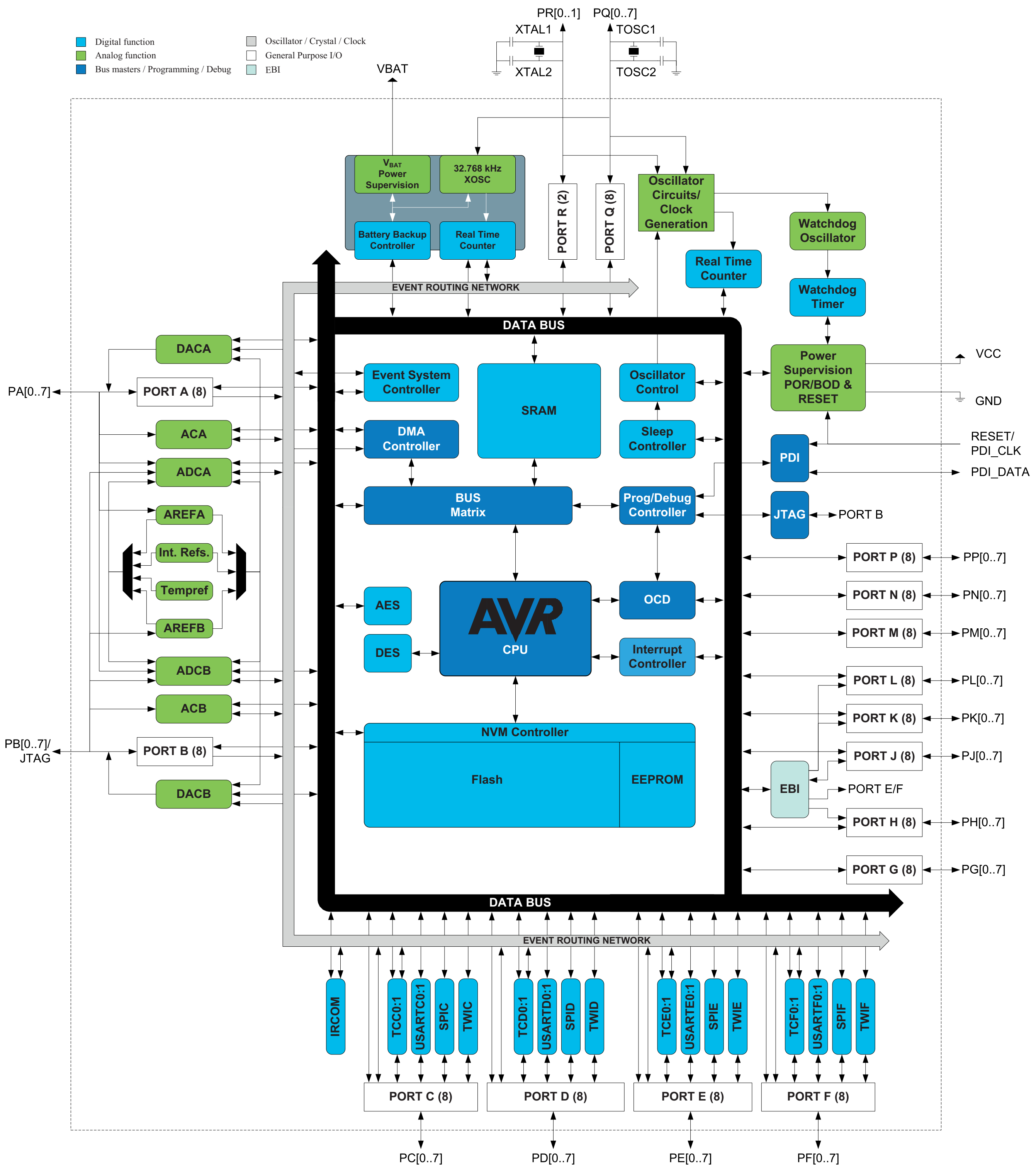
Atmel offers a free QTouch library for embedding capacitive touch buttons, sliders and wheels functionality into AVR microcontrollers.

The device are manufactured using Atmel high-density, nonvolatile memory technology. The program flash memory can be reprogrammed in-system through the PDI or JTAG interfaces. A boot loader running in the device can use any interface to download the application program to the flash memory. The boot loader 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 AVR XMEGA is a powerful microcontroller family that provides a highly flexible and cost effective solution for many embedded applications.

All Atmel AVR XMEGA 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 A1 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

- 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. Capacitive touch sensing

The Atmel QTouch library provides a simple to use solution to realize touch sensitive interfaces on most Atmel AVR microcontrollers. The patented charge-transfer signal acquisition offers robust sensing and includes fully debounced reporting of touch keys and includes Adjacent Key Suppression® (AKS®) technology for unambiguous detection of key events. The QTouch library includes support for the QTouch and QMatrix acquisition methods.

Touch sensing can be added to any application by linking the appropriate Atmel QTouch library for the AVR microcontroller. This is done by using a simple set of APIs to define the touch channels and sensors, and then calling the touch sensing API's to retrieve the channel information and determine the touch sensor states.

The QTouch library is FREE and downloadable from the Atmel website at the following location: www.atmel.com/qtouchlibrary. For implementation details and other information, refer to the QTouch library user guide - also available for download from the Atmel website.

6. 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.

7. AVR CPU

7.1 Features

- 8/16-bit high performance AVR RISC Architecture
  - 138 instructions
  - Hardware multiplier
- 32x8-bit registers directly connected to the ALU
- Stack in SRAM
- 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

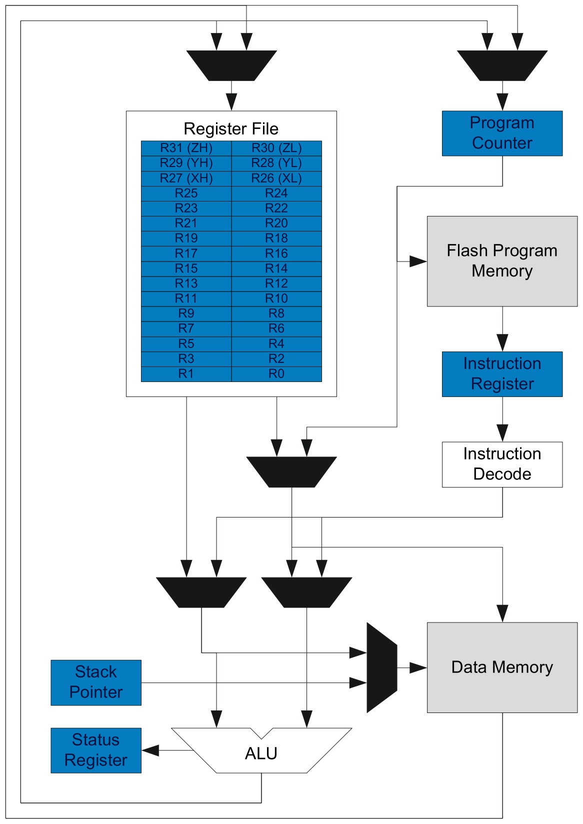
7.2 Overview

All Atmel AVR XMEGA devices use the 8/16-bit AVR CPU. The main function of the CPU is to execute the code and perform all calculations. The CPU is able to access memories, perform calculations, control peripherals, and execute the program in the flash memory. Interrupt handling is described in a separate section, refer to “Interrupts and Programmable Multilevel Interrupt Controller” on page 29.

7.3 Architectural Overview

In order to maximize performance and parallelism, the AVR CPU uses a Harvard architecture with separate memories and buses for program and data. Instructions in the program memory are executed with single-level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. This enables instructions to be executed on every clock cycle. For details of all AVR instructions, refer to http://www.atmel.com/avr.

Figure 7-1. Block diagram of the AVR CPU architecture.



The 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 in the ALU. After an arithmetic operation, the status register is updated to reflect information about the result of the operation.

The ALU is directly connected to the fast-access register file. The 32 x 8-bit general purpose working registers all have single clock cycle access time allowing single-cycle arithmetic logic unit (ALU) operation between registers or between a register and an immediate. Six of the 32 registers can be used as three 16-bit address pointers for program and data space addressing, enabling efficient address calculations.

The memory spaces are linear. The data memory space and the program memory space are two different memory spaces.

The data memory space is divided into I/O registers, SRAM, and external RAM. In addition, the EEPROM can be memory mapped in the data memory.

All I/O status and control registers reside in the lowest 4KB addresses of the data memory. This is referred to as the I/O memory space. The lowest 64 addresses can be accessed directly, or as the data space locations from 0x00 to 0x3F. The rest is the extended I/O memory space, ranging from 0x0040 to 0x0FFF. I/O registers here must be accessed as data space locations using load (LD/LDS/LDD) and store (ST/STS/STD) instructions.

The SRAM holds data. Code execution from SRAM is not supported. It can easily be accessed through the five different addressing modes supported in the AVR architecture. The first SRAM address is 0x2000.

Data addresses 0x1000 to 0x1FFF are reserved for memory mapping of EEPROM.

The program memory is divided in two sections, the application program section and the boot program section. Both sections have dedicated lock bits for write and read/write protection. The SPM instruction that is used for self-programming of the application flash memory must reside in the boot program section. The application section contains an application table section with separate lock bits for write and read/write protection. The application table section can be used for safe storing of nonvolatile data in the program memory.

7.4 ALU - Arithmetic Logic Unit

The 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. The ALU operates in direct connection with all 32 general purpose registers. In a single clock cycle, arithmetic operations between general purpose registers or between a register and an immediate are executed and the result is stored in the register file. After an arithmetic or logic operation, the status register is updated to reflect information about the result of the operation.

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 aritmetic. The hardware multiplier supports signed and unsigned multiplication and fractional format.

7.4.1 Hardware Multiplier

The multiplier is capable of multiplying two 8-bit numbers into a 16-bit result. The hardware multiplier supports different variations of signed and unsigned integer and fractional numbers:

- Multiplication of unsigned integers
- Multiplication of signed integers
- Multiplication of a signed integer with an unsigned integer
- Multiplication of unsigned fractional numbers
- Multiplication of signed fractional numbers
- Multiplication of a signed fractional number with an unsigned one

A multiplication takes two CPU clock cycles.

7.5 Program Flow

After reset, 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.

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 use a 32-bit format.

During interrupts and subroutine calls, the return address PC is stored on the stack. The stack is 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.

7.6 Status Register

The status register (SREG) contains information about the result of the most recently executed arithmetic or logic 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 nor restored when returning from an interrupt. This must be handled by software.

The status register is accessible in the I/O memory space.

7.6.1 Stack and Stack Pointer

The stack is used for storing return addresses after interrupts and subroutine calls. It can also be used for storing temporary data. The stack pointer (SP) register always points to the top of the stack. It is implemented as two 8-bit registers that are accessible in the I/O memory space. Data are pushed and popped from the stack using the PUSH and POP instructions. The stack grows from a higher memory location to a lower memory location. This implies that pushing data onto the stack decreases the SP, and popping data off the stack increases the SP. The SP is automatically loaded after reset, and the initial value is the highest address of the internal SRAM. If the SP is changed, it must be set to point above address 0x2000, and it must be defined before any subroutine calls are executed or before interrupts are enabled.

During interrupts or subroutine calls, the return address is automatically pushed on the stack. The return address can be two or three bytes, depending on program memory size of the device. For devices with 128KB or less of program memory, the return address is two bytes, and hence the stack pointer is decremented/incremented by two. For devices with more than 128KB of program memory, the return address is three bytes, and hence the SP is decremented/incremented by three. The return address is popped off the stack when returning from interrupts using the RETI instruction, and from subroutine calls using the RET instruction.

The SP is decremented by one when data are pushed on the stack with the PUSH instruction, and incremented by one when data is popped off the stack using the POP instruction.

To prevent corruption when updating the stack pointer from software, a write to SPL will automatically disable interrupts for up to four instructions or until the next I/O memory write.

After reset the stack pointer is initialized to the highest address of the SRAM. See Table 8-2 on page 15.

7.7 Register File

The register file consists of 32 x 8-bit general purpose working registers with single clock cycle access time. The register file supports the following input/output schemes:

- 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

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 lookup tables in flash program memory.

8. Memories

8.1 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
  - External Memory support
    - SRAM
    - SDRAM
    - Memory mapped external hardware
  - 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

8.2 Overview

The Atmel AVR architecture has two main memory spaces, the program memory and the data memory. Executable code can reside only in the program memory, while data can be stored in the program memory and the data memory. The data memory includes the internal SRAM, and EEPROM for nonvolatile data storage. All memory spaces are linear and require no memory bank switching. Nonvolatile memory (NVM) spaces can be locked for further write and read/write operations. This prevents unrestricted access to the application software.

A separate memory section contains the fuse bytes. These are used for configuring important system functions, and can only be written by an external programmer.

The available memory size configurations are shown in "Ordering Information" on page 2. In addition each device has a flash memory signature rows for calibration data, device identification, serial number etc.

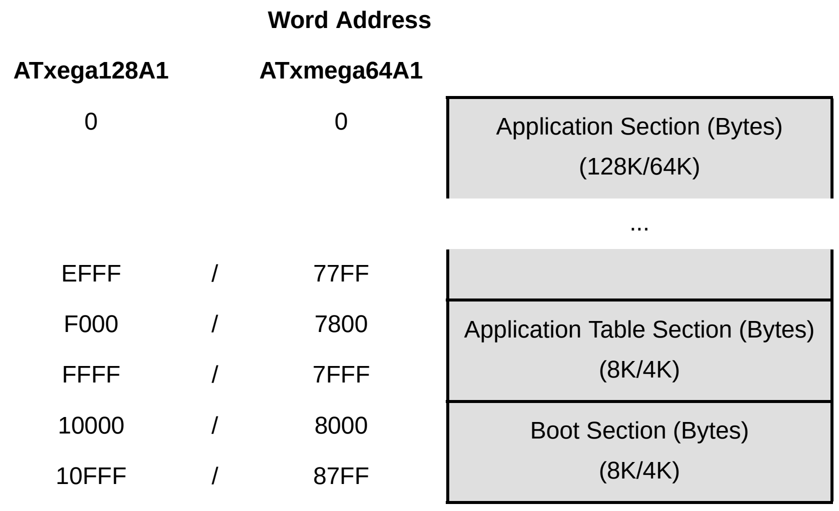
8.3 In-System Programmable Flash Program Memory

he Atmel AVR XMEGA devices contain on-chip, in-system reprogrammable flash memory for program storage. The flash memory can be accessed for read and write from an external programmer through the PDI or from application software running in the device.

All AVR CPU instructions are 16 or 32 bits wide, and each flash location is 16 bits wide. The flash memory is organized in two main sections, the application section and the boot loader section. The sizes of the different sections are fixed, but device-dependent. These two sections have separate lock bits, and can have different levels of protection. The store program memory (SPM) instruction, which is used to write to the flash from the application software, will only operate when executed from the boot loader section.

The application section contains an application table section with separate lock settings. This enables safe storage of nonvolatile data in the program memory.

Figure 8-1. Flash Program Memory (Hexadecimal address)



8.3.1 Application Section

The Application section is the section of the flash that is used for storing the executable application code. The protection level for the application section can be selected by the boot lock bits for this section. The application section can not store any boot loader code since the SPM instruction cannot be executed from the application section.

8.3.2 Application Table Section

The application table section is a part of the application section of the flash memory that can be used for storing data. The size is identical to the boot loader section. The protection level for the application table section can be selected by the boot lock bits for this section. The possibilities for different protection levels on the application section and the application table section enable safe parameter storage in the program memory. If this section is not used for data, application code can reside here.

8.3.3 Boot Loader Section

While the application section is used for storing the application code, the boot loader software must be located in the boot loader section because the SPM instruction can only initiate programming when executing from this section. The SPM instruction can access the entire flash, including the boot loader section itself. The protection level for the boot loader section can be selected by the boot loader lock bits. If this section is not used for boot loader software, application code can be stored here.

8.3.4 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. Some of the calibration values will be automatically loaded to the corresponding module or peripheral unit during reset. Other values must be loaded from the signature row and written to the corresponding peripheral registers from software. For details on calibration conditions, refer to “Electrical Characteristics” on page 76.

The production signature row also contains an ID that identifies each microcontroller device type and a serial number for each manufactured device. The serial number consists of the production lot number, wafer number, and wafer coordinates for the device. The device ID for the available devices is shown in Table 8-1.

The production signature row cannot be written or erased, but it can be read from application software and external programmers.

Table 8-1. Device ID bytes.

Device	Device ID bytes		
	Byte 2	Byte 1	Byte 0
ATxmega64A1	4E	96	1E
ATxmega128A1	4C	97	1E

8.3.5 User Signature Row

The user signature row is a separate memory section that is fully accessible (read and write) from application software and external programmers. It is one flash page in size, and is meant for static user parameter storage, such as calibration data, custom serial number, 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 operations and on-chip debug sessions.

8.4 Fuses and Lock bits

The fuses are used to configure important system functions, and can only be written from an external programmer. The application software can read the fuses. The fuses are used to configure reset sources such as brownout detector and watchdog, startup configuration, JTAG enable, and JTAG user ID.

The lock bits are used to set protection levels for the different flash sections (that is, if read and/or write access should be blocked). Lock bits can be written by external programmers and application software, but only to stricter protection levels. Chip erase is the only way to erase the lock bits. To ensure that flash contents are protected even during chip erase, the lock bits are erased after the rest of the flash memory has been erased.

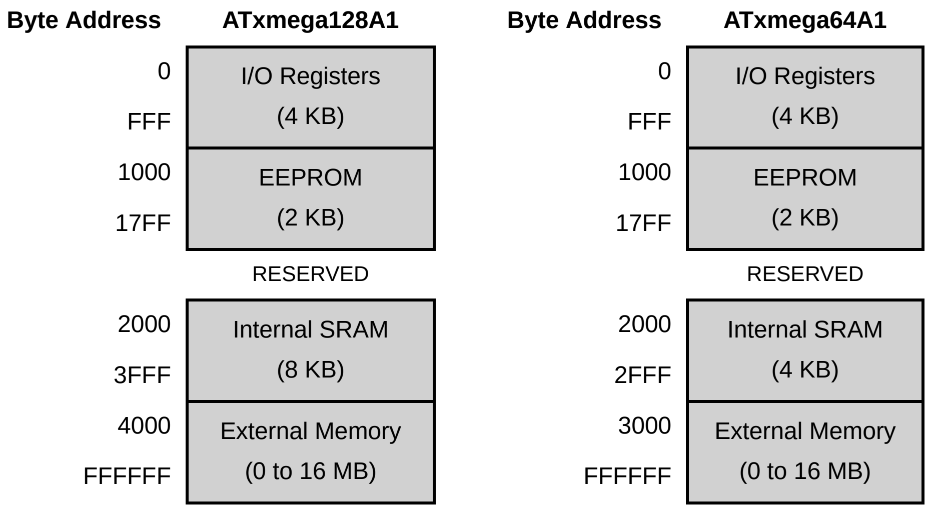
An unprogrammed fuse or lock bit will have the value one, while a programmed fuse or lock bit will have the value zero.

Both fuses and lock bits are reprogrammable like the flash program memory.

8.5 Data Memory

The data memory contains the I/O memory, internal SRAM, optionally memory mapped EEPROM, and external memory if available. The data memory is organized as one continuous memory section, see Figure 8-2 on page 15. To simplify development, I/O Memory, EEPROM and SRAM will always have the same start addresses for all Atmel AVR XMEGA devices. The address space for External Memory will always start at the end of Internal SRAM and end at address 0xFFFFFF.

Figure 8-2. Data Memory Map (Hexadecimal address)



8.6 EEPROM

XMEGA AU devices have EEPROM for nonvolatile data storage. It is either addressable in a separate data space (default) or memory mapped and accessed in normal data space. The EEPROM supports both byte and page access. Memory mapped EEPROM allows highly efficient EEPROM reading and EEPROM buffer loading. When doing this, EEPROM is accessible using load and store instructions. Memory mapped EEPROM will always start at hexadecimal address 0x1000.

8.7 I/O Memory

The status and configuration registers for peripherals and modules, including the CPU, are addressable through I/O memory locations. All I/O locations can be accessed by the load (LD/LDS/LDD) and store (ST/STS/STD) instructions, which is used to transfer data between the 32 registers in the register file and the I/O memory. The IN and OUT instructions can address I/O memory locations in the range 0x00 - 0x3F directly. In the address range 0x00 - 0x1F, single- cycle instructions for manipulation and checking of individual bits are available.

The I/O memory address for all peripherals and modules in XMEGA A1U is shown in the "Peripheral Module Address Map" on page 62.

8.7.1 General Purpose I/O Registers

The lowest 16 I/O memory addresses are reserved as general purpose I/O registers. These registers can be used for storing global variables and flags, as they are directly bit-accessible using the SBI, CBI, SBIS, and SBIC instructions.

8.8 External Memory

Four ports can be used for external memory, supporting external SRAM, SDRAM, and memory mapped peripherals such as LCD displays. Refer to "EBI – External Bus Interface" on page 47. The external memory address space will always start at the end of internal SRAM.

8.9 Data Memory and Bus Arbitration

Since the data memory is organized as four separate sets of memories, the different bus masters (CPU, DMA controller read and DMA controller write, etc.) can access different memory sections at the same time.

8.10 Memory Timing

Read and write access to the I/O memory takes one CPU clock cycle. A write to SRAM takes one cycle, and a read from SRAM takes two cycles. For burst read (DMA), new data are available every cycle. EEPROM page load (write) takes one cycle, and three cycles are required for read. For burst read, new data are available every second cycle. External memory has multi-cycle read and write. The number of cycles depends on the type of memory and configuration of the external bus interface. Refer to the instruction summary for more details on instructions and instruction timing.

8.11 Device ID and Revision

Each device has a three-byte device ID. This ID identifies Atmel as the manufacturer of the device and the device type. A separate register contains the revision number of the device.

8.12 I/O Memory Protection

Some features in the device are regarded as critical for safety in some applications. Due to this, it is possible to lock the I/O register related to the clock system, the event system, and the advanced waveform extensions. As long as the lock is enabled, all related I/O registers are locked and they can not be written from the application software. The lock registers themselves are protected by the configuration change protection mechanism.

8.13 JTAG Disable

It is possible to disable the JTAG interface from the application software. This will prevent all external JTAG access to the device until the next device reset or until JTAG is enabled again from the application software. As long as JTAG is disabled, the I/O pins required for JTAG can be used as normal I/O pins.

8.14 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 8-2 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) gives the page number and the least significant address bits (FWORD) gives the word in the page.

Table 8-2. Number of words and Pages in the Flash.

Device	PC size	Flash	Page Size	FWORD	FPAGE	Application		Boot	
	bits	bytes	words			Size	No of pages	Size	No of pages
ATxmega64A1	16	64K + 4K	128	Z[7:1]	Z[16:8]	64K	256	4K	16
ATxmega128A1	17	128K+ 8K	256	Z[8:1]	Z[17:9]	128K	256	8K	16

Table 8-3 shows EEPROM memory organization for the Atmel AVR XMEGA A1U 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 8-3. Number of Bytes and Pages in the EEPROM.

Device	EEPROM	Page Size	E2BYTE	E2PAGE	No of pages
	Size	**bytes**			
ATxmega64A1	2 KB	32	ADDR[4:0]	ADDR[10:5]	64
ATxmega128A1	2 KB	32	ADDR[4:0	ADDR[10:5]	64

8.14.1 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 A1 is shown in the "Peripheral Module Address Map" on page 62.

9. DMAC - Direct Memory Access Controller

9.1 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 16M bytes transfers in a single transaction
- Multiple addressing modes for source and destination address
  - Increment
  - Decrement
  - Static
- 1, 2, 4, or 8 byte Burst Transfers
- Programmable priority between channels

9.2 Overview

The four-channel direct memory access (DMA) controller can transfer data between memories and peripherals, and thus offload these tasks from the CPU. It enables high data transfer rates with minimum CPU intervention, and frees up CPU time. The four DMA channels enable up to four independent and parallel transfers.

The DMA controller can move data between SRAM and peripherals, between SRAM locations and directly between peripheral registers. With access to all peripherals, the DMA controller can handle automatic transfer of data to/from communication modules. The DMA controller can also read from memory mapped EEPROM.

Data transfers are done in continuous bursts of 1, 2, 4, or 8 bytes. They build block transfers of configurable size from 1 byte to 64KB. A repeat counter can be used to repeat each block transfer for single transactions up to 16MB. Source and destination addressing can be static, incremental or decremental. Automatic reload of source and/or destination addresses can be done after each burst or block transfer, or when a transaction is complete. Application software, peripherals, and events can trigger DMA transfers.

The four DMA channels have individual configuration and control settings. This include source, destination, transfer triggers, and transaction sizes. They have individual interrupt settings. Interrupt requests can be generated when a transaction is complete or when the DMA controller detects an error on a DMA channel.

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

10. Event System

10.1 Features

- Inter-peripheral communication and signalling with minimum latency
- CPU and DMA independent operation
- 8 Event Channels allows 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

10.2 Overview

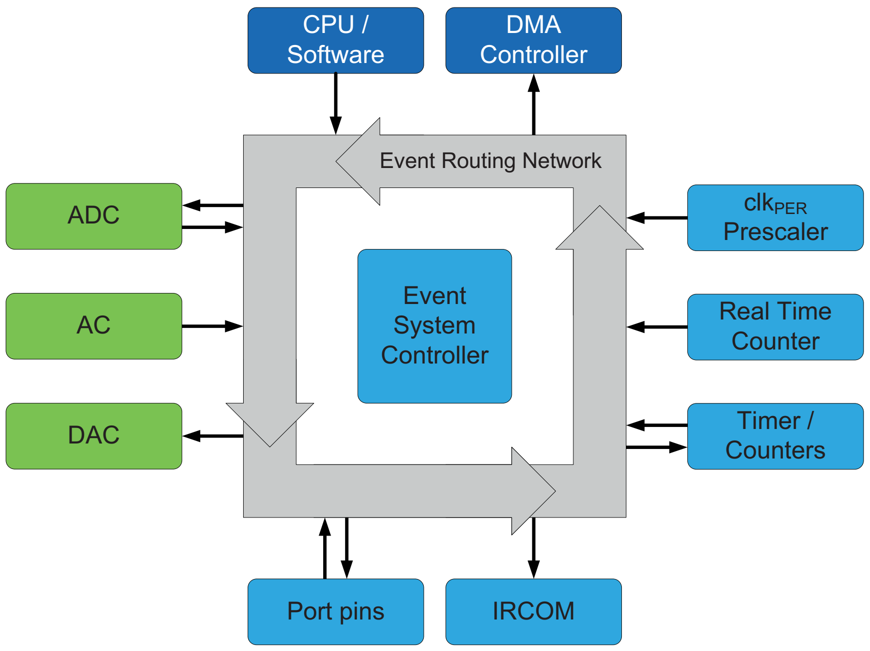
The Event System is a set of features for inter-peripheral communication. It enables the possibility for a change of state in one peripheral to automatically trigger actions in one or more peripherals. These changes in a peripheral that will trigger actions in other peripherals are configurable by software. 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 10-1 on page 20 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 10-1. Event system block diagram.



he event routing network consists of eight software-configurable multiplexers that control how events are routed and used. These are called event channels, and allow for up to eight parallel event routing configurations. The maximum routing latency is two peripheral clock cycles. The event system works in both active mode and idle sleep mode.

11. System Clock and Clock options

11.1 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 ouput
- 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 1x to 2048x division
- Fast peripheral clock running at two and four times the CPU clock speed
- Automatic Run-Time Calibration of internal oscillators
- Crystal Oscillator failure detection

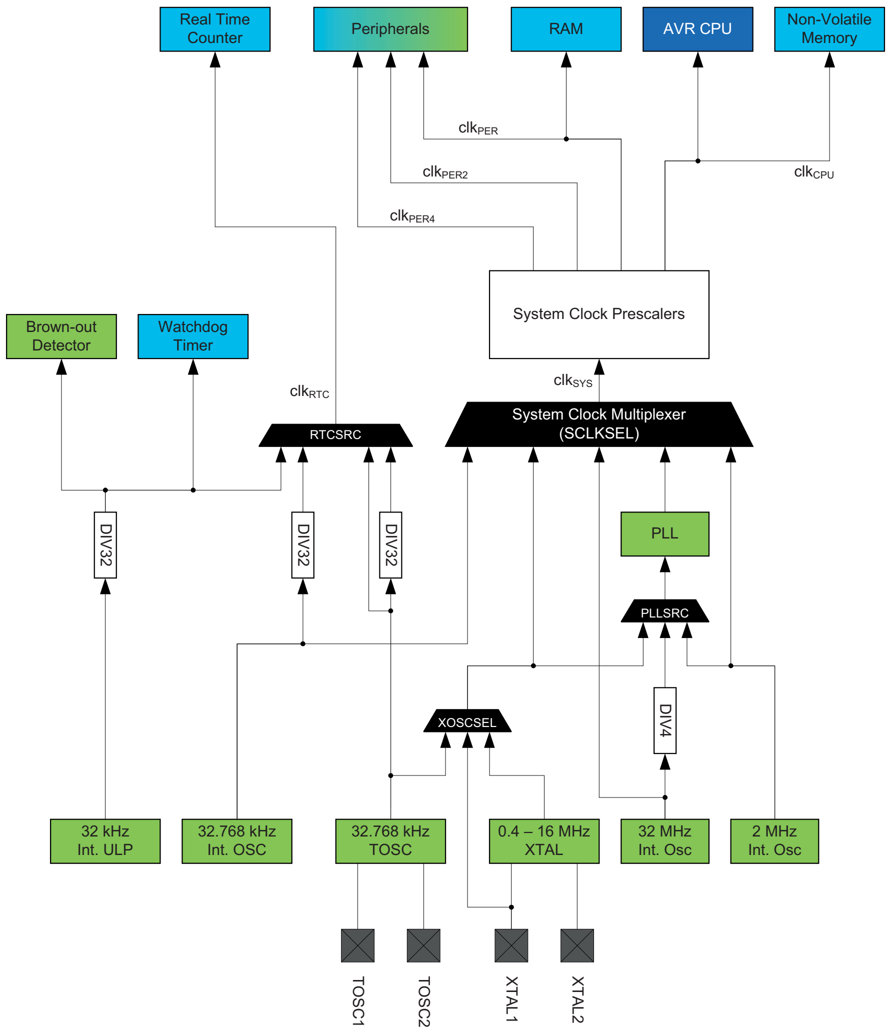
11.2 Overview

Atmel AVR XMEGA devices have a flexible clock system supporting a large number of clock sources. It incorporates both accurate internal oscillators and external crystal oscillator and resonator support. A high-frequency phase locked loop (PLL) and clock prescalers can be used to generate a wide range of clock frequencies. An oscillator failure monitor can be enabled to issue a non-maskable interrupt and switch to the internal oscillator if the external oscillator or PLL fails.

When a reset occurs, all clock sources except the 32kHz ultra low power oscillator are disabled. After reset, the device will always start up running from the 2MHz internal oscillator. During normal operation, the system clock source and prescalers can be changed from software at any time.

Figure 11-1 on page 22 presents the principal clock system in the XMEGA A1U family devices. Not all of the clocks need to be active at a given time. The clocks for the CPU and peripherals can be stopped using sleep modes and power reduction registers as described in "Power Management and Sleep Modes" on page 24.

Figure 11-1. The clock system, clock sources and clock distribution



11.3 Clock Options

The clock sources are divided in two main groups: internal oscillators and external clock sources. Most of the clock sources can be directly enabled and disabled from software, while others are automatically enabled or disabled, depending on peripheral settings. After reset, the device starts up running from the 2MHz internal oscillator. The other clock sources and PLL are turned off by default.

The internal oscillators do not require any external components to run. For details on characteristics and accuracy of the internal oscillators, refer to the device datasheet.

11.3.1 32 kHz Ultra Low Power Internal Oscillator

This oscillator provides an approximate 32kHz clock. The 32kHz ultra low power (ULP) internal oscillator is a very low power clock source, and it is not designed for high accuracy. The oscillator employs a built-in prescaler that provides a 1kHz output. The oscillator is automatically enabled/disabled when it is used as clock source for any part of the device. This oscillator can be selected as the clock source for the RTC.

11.3.2 32.768 kHz Calibrated Internal Oscillator

This oscillator provides an approximate 32.768kHz clock. It is calibrated during production to provide a default frequency close to its nominal frequency. The calibration register can also be written from software for run-time calibration of the oscillator frequency. The oscillator employs a built-in prescaler, which provides both a 32.768kHz output and a 1.024kHz output.

11.3.3 32.768 kHz Crystal Oscillator

A 32.768kHz crystal oscillator can be connected between the 1 and 2 pins and enables a dedicated low frequency oscillator input circuit. A low power mode with reduced voltage swing on 2 is available. This oscillator can be used as a clock source for the system clock and RTC.

11.3.4 0.4 - 16 MHz Crystal Oscillator

This oscillator can operate in four different modes optimized for different frequency ranges, all within 0.4 - 16MHz.

11.3.5 2 MHz Run-time Calibrated Internal Oscillator

The 2MHz 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 32kHz Calibrated Internal Oscillator or the 32kHz 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.

11.3.6 32 MHz Run-time Calibrated Internal Oscillator

The 32MHz 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 32kHz Calibrated Internal Oscillator or the 32kHz 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.

11.3.7 External Clock input

The XTAL1 and XTAL2 pins can be used to drive an external oscillator, either a quartz crystal or a ceramic resonator. XTAL1 can be used as input for an external clock signal. The 1 and 2 pins is dedicated to driving a 32.768kHz crystal oscillator.

11.3.8 PLL with Multiplication factor 1 - 31x

The built-in phase locked loop (PLL) can be used to generate a high-frequency system clock. The PLL has a user-selectable multiplication factor of from 1 to 31. In combination with the prescalers, this gives a wide range of output frequencies from all clock sources.

12. Power Management and Sleep Modes

12.1 Features

- Power management for adjusting power consumption and functions
- 5 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

12.2 Overview

Various sleep modes and clock gating are provided in order to tailor power consumption to application requirements. This enables the Atmel AVR 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.

12.3 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.

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.

12.3.1 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.

12.3.2 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 and asynchronous port interrupts, e.g pin change.

12.3.3 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.

12.3.4 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.

12.3.5 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.

13. System Control and Reset

13.1 Features

- Multiple reset sources for safe operation and device reset
  - Power-On Reset
  - External Reset
  - Watchdog Reset
  - Brown-Out Reset
  - PDI reset
  - Software reset
- Asynchronous reset
  - No running clock in the device is required for reset
- Reset status register

13.2 Overview

The reset system issues a microcontroller reset and sets the device to its initial state. This is for situations where operation should not start or continue, such as when the microcontroller operates below its power supply rating. If a reset source goes active, the device enters and is kept in reset until all reset sources have released their reset. The I/O pins are immediately tri-stated. The program counter is set to the reset vector location, and all I/O registers are set to their initial values. The SRAM content is kept. However, if the device accesses the SRAM when a reset occurs, the content of the accessed location can not be guaranteed.

After reset is released from all reset sources, the default oscillator is started and calibrated before the device starts running from the reset vector address. By default, this is the lowest program memory address, 0, but it is possible to move the reset vector to the lowest address in the boot section.

The reset functionality is asynchronous, and so no running system clock is required to reset the device. The software reset feature makes it possible to issue a controlled system reset from the user software.

The reset status register has individual status flags for each reset source. It is cleared at power-on reset, and shows which sources have issued a reset since the last power-on.

13.3 Reset Sequence

A reset request from any reset source will immediately reset the device and keep it in reset as long as the request is active. When all reset requests are released, the device will go through three stages before the device starts running again:

- Reset counter delay
- Oscillator startup
- Oscillator calibration

If another reset requests occurs during this process, the reset sequence will start over again.

13.4 Reset Sources

13.4.1 Power-On Reset

TA power-on reset (POR) is generated by an on-chip detection circuit. The POR is activated when the V_{CC} rises and reaches the POR threshold voltage (V_{POT}), and this will start the reset sequence.

The POR is also activated to power down the device properly when the V_{CC} falls and drops below the V_{POT} level.

The V_{POT} level is higher for falling V_{CC} than for rising V_{CC}. Consult the datasheet for POR characteristics data.

13.4.2 Brownout Detection

The on-chip brownout detection (BOD) circuit monitors the V_{CC} level during operation by comparing it to a fixed, programmable level that is selected by the BODLEVEL fuses. If disabled, BOD is forced on at the lowest level during chip erase and when the PDI is enabled.

13.4.3 External Reset

The external reset circuit is connected to the external $\overline{\text{RESET}}$ pin. The external reset will trigger when the RESET pin is driven below the $\overline{\text{RESET}}$ pin threshold voltage, V_{RST}, for longer than the minimum pulse period, t_{EXT}. The reset will be held as long as the pin is kept low. The $\overline{\text{RESET}}$ pin includes an internal pull-up resistor.

13.4.4 Watchdog Reset

The watchdog timer (WDT) is a system function for monitoring correct program operation. If the WDT is not reset from the software within a programmable timeout period, a watchdog reset will be given. The watchdog reset is active for one to two clock cycles of the 2MHz internal oscillator. For more details see "WDT - Watchdog Timer" on page 28.

13.4.5 Software reset

The software reset makes it possible to issue a system reset from software by writing to the software reset bit in the reset control register.The reset will be issued within two CPU clock cycles after writing the bit. It is not possible to execute any instruction from when a software reset is requested until it is issued.

13.4.6 Program and Debug Interface Reset

The program and debug interface reset contains a separate reset source that is used to reset the device during external programming and debugging. This reset source is accessible only from external debuggers and programmers.

13.5 WDT - Watchdog Timer

13.5.1 Features

- Issues a device reset if the timer is not reset before its timeout period
- Asynchronous operation from dedicated oscillator
- 1kHz output of the 32kHz ultra low power oscillator
- 11 selectable timeout periods, from 8ms to 8s
- Two operation modes:
  - Normal mode
  - Window mode
- Configuration lock to prevent unwanted changes

13.6 Overview

The watchdog timer (WDT) is a system function for monitoring correct program operation. It makes it possible to recover from error situations such as runaway or deadlocked code. The WDT is a timer, configured to a predefined timeout period, and is constantly running when enabled. If the WDT is not reset within the timeout period, it will issue a microcontroller reset. The WDT is reset by executing the WDR (watchdog timer reset) instruction from the application code.

The window mode makes it possible to define a time slot or window inside the total timeout period during which WDT must be reset. If the WDT is reset outside this window, either too early or too late, a system reset will be issued. Compared to the normal mode, this can also catch situations where a code error causes constant WDR execution.

The WDT will run in active mode and all sleep modes, if enabled. It is asynchronous, runs from a CPU-independent clock source, and will continue to operate to issue a system reset even if the main clocks fail.

The configuration change protection mechanism ensures that the WDT settings cannot be changed by accident. For increased safety, a fuse for locking the WDT settings is also available.

14. Interrupts and Programmable Multilevel Interrupt Controller

14.1 Features

- Short and predictable interrupt response time
- Separate interrupt configuration and vector address for each interrupt
- Programmable multilevel interrupt controller
  - Interrupt prioritizing according to level and vector address
  - Three selectable interrupt levels for all interrupts: low, medium and high
  - Selectable, round-robin priority scheme within low-level interrupts
  - Non-maskable interrupts for critical functions
- Interrupt vectors optionally placed in the application section or the boot loader section

14.2 Overview

Interrupts signal a change of state in peripherals, and this can be used to alter program execution. Peripherals can have one or more interrupts, and all are individually enabled and configured. When an interrupt is enabled and configured, it will generate an interrupt request when the interrupt condition is present. The programmable multilevel interrupt controller (PMIC) controls the handling and prioritizing of interrupt requests. When an interrupt request is acknowledged by the PMIC, the program counter is set to point to the interrupt vector, and the interrupt handler can be executed.

All peripherals can select between three different priority levels for their interrupts: low, medium, and high. Interrupts are prioritized according to their level and their interrupt vector address. Medium-level interrupts will interrupt low-level interrupt handlers. High-level interrupts will interrupt both medium- and low-level interrupt handlers. Within each level, the interrupt priority is decided from the interrupt vector address, where the lowest interrupt vector address has the highest interrupt priority. Low-level interrupts have an optional round-robin scheduling scheme to ensure that all interrupts are serviced within a certain amount of time.

Non-maskable interrupts (NMI) are also supported, and can be used for system critical functions.

14.3 Interrupt vectors

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 Atmel AVR XMEGA A1U devices are shown in Table 14-1. Offset addresses for each interrupt available in the peripheral are described for each peripheral in the XMEGA AU 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

Program Address (Base Address)	Source	Interrupt Description
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
0x040	NVM_INT_base	Non-Volatile Memory Interrupt base
0x044	PORTB_INT_base	Port B Interrupt base
0x048	ACB_INT_base	Analog Comparator on Port B Interrupt base
0x04E	ADCB_INT_base	Analog to Digital Converter on Port B Interrupt base
0x056	PORTE_INT_base	Port E Interrupt base
0x05A	TWIE_INT_base	Two-Wire Interface on Port E Interrupt base
0x05E	TCE0_INT_base	Timer/Counter 0 on port E Interrupt base
0x06A	TCE1_INT_base	Timer/Counter 1 on port E Interrupt base
0x072	SPIE_INT_vect	SPI on port E Interrupt vector
0x074	USARTE0_INT_base	USART 0 on port E Interrupt base
0x07A	USARTE1_INT_base	USART 1 on port E Interrupt base
0x080	PORTD_INT_base	Port D Interrupt base
0x084	PORTA_INT_base	Port A Interrupt base
0x088	ACA_INT_base	Analog Comparator on Port A Interrupt base
0x08E	ADCA_INT_base	Analog to Digital Converter on Port A Interrupt base
0x096	TWID_INT_base	Two-Wire Interface on Port D Interrupt base
0x09A	TCD0_INT_base	Timer/Counter 0 on port D Interrupt base
0x0A6	TCD1_INT_base	Timer/Counter 1 on port D Interrupt base
0x0AE	SPID_INT_vector	SPI on port D Interrupt vector
0x0B0	USARTD0_INT_base	USART 0 on port D Interrupt base
0x0B6	USARTD1_INT_base	USART 1 on port D Interrupt base
0x0BC	PORTQ_INT_base	Port Q INT base
0x0C0	PORTH_INT_base	Port H INT base
0x0C4	PORTJ_INT_base	Port J INT base
0x0C8	PORTK_INT_base	Port K INT base
0x0D0	PORTF_INT_base	Port F INT base
0x0D4	TWIF_INT_base	Two-Wire Interface on Port F INT base

Program Address (Base Address)	Source	Interrupt Description
0x0D8	TCF0_INT_base	Timer/Counter 0 on port F Interrupt base
0x0E4	TCF1_INT_base	Timer/Counter 1 on port F Interrupt base
0x0EC	SPIF_INT_vector	SPI ion port F Interrupt base
0x0EE	USARTF0_INT_base	USART 0 on port F Interrupt base
0x0F4	USARTF1_INT_base	USART 1 on port F Interrupt base

15. I/O Ports

15.1 Features

- 78 General purpose input and output pins with individual configuration
- Output driver with configurable driver and pull settings:
  - Totem-pole
  - Wired-AND
  - Wired-OR
  - Bus-keeper
  - Inverted I/O
- Input with synchronous and/or asynchronous sensing with interrupts and events
  - Sense both edges
  - Sense rising edges
  - Sense falling edges
  - Sense low level
- Optional pull-up and pull-down resistor on input and Wired-OR/AND configurations
- Optional slew rate control
- Asynchronous pin change sensing that can wake the device from all sleep modes
- Two port interrupts with pin masking per I/O port
- Efficient and safe access to port pins
  - Hardware read-modify-write through dedicated toggle/clear/set registers
  - Configuration of multiple pins in a single operation
  - Mapping of port registers into bit-accessible I/O memory space
- Peripheral clocks output on port pin
- Real-time counter clock output to port pin
- Event channels can be output on port pin
- Remapping of digital peripheral pin functions
  - Selectable USART, SPI, and timer/counter input/output pin locations

15.2 Overview

One port consists of up to eight port pins: pin 0 to 7. Each port pin can be configured as input or output with configurable driver and pull settings. They also implement synchronous and asynchronous input sensing with interrupts and events for selectable pin change conditions. Asynchronous pin-change sensing means that a pin change can wake the device from all sleep modes, included the modes where no clocks are running.

All functions are individual and configurable per pin, but several pins can be configured in a single operation. The pins have hardware read-modify-write (RMW) functionality for safe and correct change of drive value and/or pull resistor configuration. The direction of one port pin can be changed without unintentionally changing the direction of any other pin.

The port pin configuration also controls input and output selection of other device functions. It is possible to have both the peripheral clock and the real-time clock output to a port pin, and available for external use. The same applies to events from the event system that can be used to synchronize and control external functions. Other digital peripherals, such as USART, SPI, and timer/counters, can be remapped to selectable pin locations in order to optimize pin-out versus application needs.

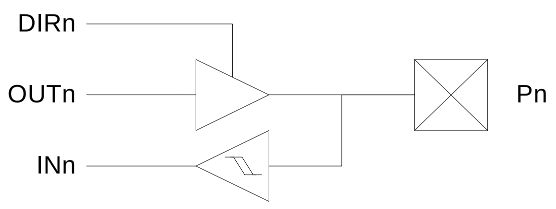
The notation of these ports are PORTA, PORTB, PORTC, PORTD, PORTE, PORTF, PORTH, PORTJ, PORTK, PORTQ and PORTR.

15.3 Output Driver

All port pins (Pn) have programmable output configuration. The port pins also have configurable slew rate limitation to reduce electromagnetic emission.

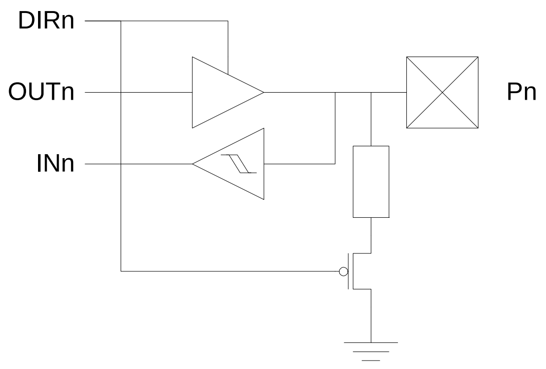
15.3.1 Push-pull

Figure 15-1. I/O configuration - Totem-pole



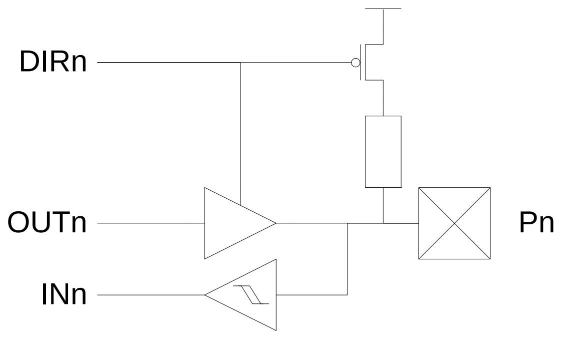
15.3.2 Pull-down

Figure 15-2. I/O configuration - Totem-pole with pull-down (on input)



15.3.3 Pull-up

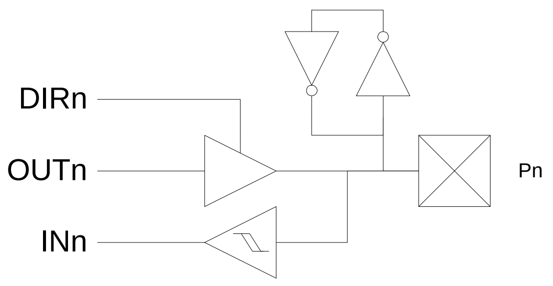
Figure 15-3. I/O configuration - Totem-pole with pull-up (on input)



15.3.4 Bus-keeper

The bus-keeper's weak output produces the same logical level as the last output level. It acts as a pull-up if the last level was '1', and pull-down if the last level was '0'.

Figure 15-4. I/O configuration - Totem-pole with bus-keeper



15.3.5 Others

Figure 15-5. Output configuration - Wired-OR with optional pull-down

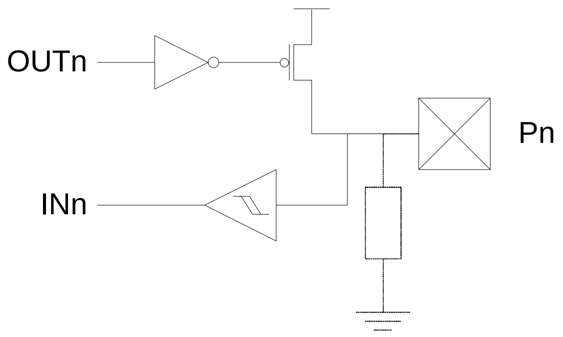
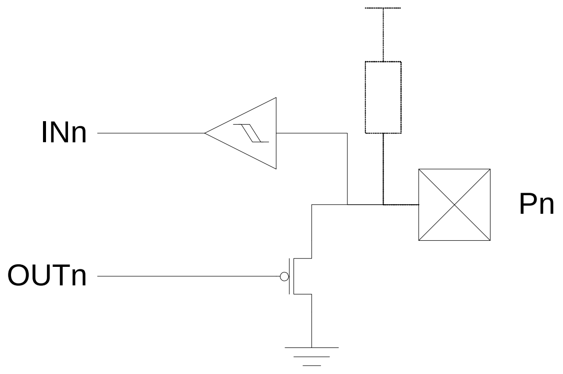


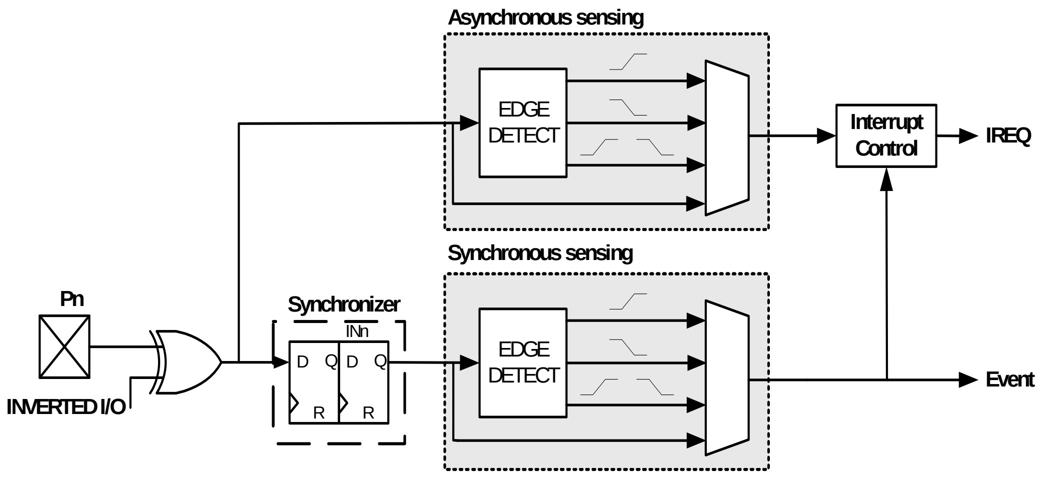
Figure 15-6. I/O configuration - Wired-AND with optional pull-up



15.4 Input sensing

Input sensing is synchronous or asynchronous depending on the enabled clock for the ports, and the configuration is shown in Figure 15-7 on page 35.

Figure 15-7. Input sensing system overview



When a pin is configured with inverted I/O the pin value is inverted before the input sensing.

15.5 Port Interrupt

Each ports have two interrupts with seperate priority and interrupt vector. All pins on the port can be individually selected as source for each of the interrupts. The interrupts are then triggered according to the input sense configuration for each pin configured as source for the interrupt.

15.6 Alternate Port Functions

In addition to the input/output functions on all port pins, most pins have alternate functions. This means that other modules or peripherals connected to the port can use the port pins for their functions, such as communication or pulse-width modulation. "Pinout and Pin Functions" on page 55 shows which modules on peripherals that enables alternate functions on a pin, and what alternate functions that is available on a pin.

16. T/C - 16-bit Timer/Counter

16.1 Features

- Eight 16-bit Timer/Counters
  - Four Timer/Counters of type 0
  - Four Timer/Counters of type 1
- Four Compare or Capture (CC) Channels in Timer/Counter 0
- Two Compare or Capture (CC) Channels in Timer/Counter 1
- Double Buffered Timer Period Setting
- Double Buffered Compare or Capture Channels
- Waveform Generation:
  - Single Slope Pulse Width Modulation
  - Dual Slope Pulse Width Modulation
  - Frequency Generation
- Input Capture:
  - Input Capture with Noise Cancelling
  - Frequency capture
  - Pulse width capture
  - 32-bit input capture
- Event Counter with Direction Control
- Timer Overflow and Timer Error Interrupts and Events
- One Compare Match or Capture Interrupt and Event per CC Channel
- Supports DMA Operation
- Hi-Resolution Extension (Hi-Res)
- Advanced Waveform Extension (AWEX)

16.2 Overview

Atmel AVR XMEGA devices have a set of eight flexible 16-bit timer/counters (TC). Their capabilities include accurate program execution timing, frequency and waveform generation, and input capture with time and frequency measurement of digital signals. Two timer/counters can be cascaded to create a 32-bit timer/counter with optional 32-bit capture.

A timer/counter consists of a base counter and a set of compare or capture (CC) channels. The base counter can be used to count clock cycles or events. It has direction control and period setting that can be used for timing. The CC channels can be used together with the base counter to do compare match control, frequency generation, and pulse width waveform modulation, as well as various input capture operations. A timer/counter can be configured for either capture or compare functions, but cannot perform both at the same time.

A timer/counter can be clocked and timed from the peripheral clock with optional prescaling or from the event system. The event system can also be used for direction control and capture trigger or to synchronize operations.

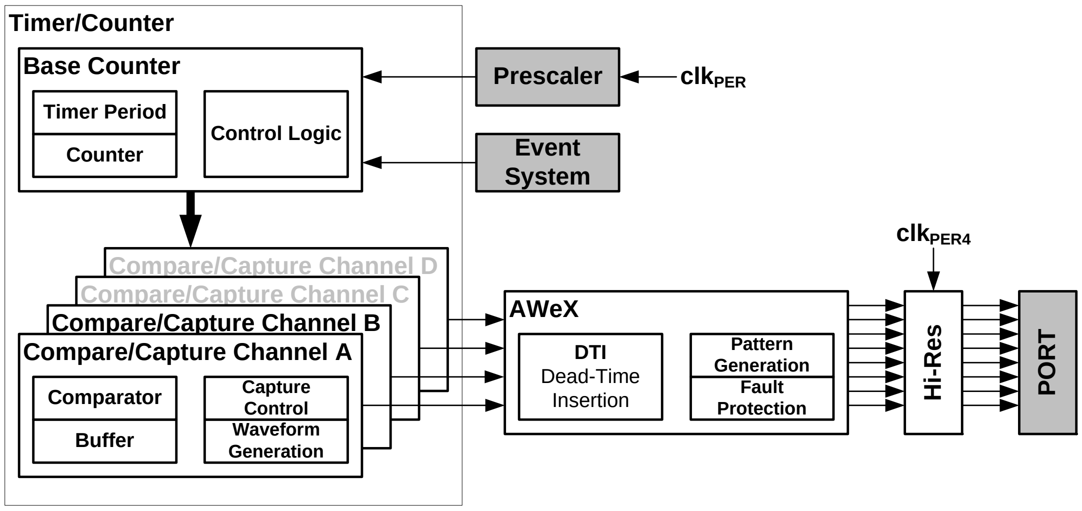
There are two differences between timer/counter type 0 and type 1. Timer/counter 0 has four CC channels, and timer/counter 1 has two CC channels. All information related to CC channels 3 and 4 is valid only for timer/counter 0. Only Timer/Counter 0 has the split mode feature that split it into 2 8-bit Timer/Counters with four compare channels each.

Some timer/counters have extensions to enable more specialized waveform and frequency generation. The advanced waveform extension (AWeX) is intended for motor control and other power control applications. It enables low- and high-side output with dead-time insertion, as well as fault protection for disabling and shutting down external drivers. It can also generate a synchronized bit pattern across the port pins.

The advanced waveform extension can be enabled to provide extra and more advanced features for the Timer/Counter. This is only available for Timer/Counter 0. See "AWeX - Advanced Waveform Extension" on page 38 for more details.

The high-resolution (hi-res) extension can be used to increase the waveform output resolution by four or eight times by using an internal clock source running up to four times faster than the peripheral clock. See "Hi-Res - High Resolution Extension" on page 39 for more details.

Figure 16-1. Overview of a Timer/Counter and closely related peripherals



PORTC, PORTD, PORTE and PORTF each has one Timer/Counter 0 and one Timer/Counter1. Notation of these Timer/Counters are TCC0 (Time/Counter C0), TCC1, TCD0, TCD1, TCE0, TCE1, TCF0, and TCF1, respectively.

17. AWeX - Advanced Waveform Extension

17.1 Features

- Output with complementary output from each Capture channel
- Four Dead Time Insertion (DTI) Units, one for each Capture channel
- 8-bit DTI Resolution
- Separate High and Low Side Dead-Time Setting
- Double Buffered Dead-Time
- Event Controlled Fault Protection
- Single Channel Multiple Output Operation (for BLDC motor control)
- Double Buffered Pattern Generation

17.2 Overview

The advanced waveform extension (AWeX) provides extra functions to the timer/counter in waveform generation (WG) modes. It is primarily intended for use with different types of motor control and other power control applications. It enables low- and high side output with dead-time insertion and fault protection for disabling and shutting down external drivers. It can also generate a synchronized bit pattern across the port pins.

Each of the waveform generator outputs from the Timer/Counter 0 are split into a complimentary pair of outputs when any AWeX features are enabled. These output pairs go through a dead-time insertion (DTI) unit that generates the non-inverted low side (LS) and inverted high side (HS) of the WG output with dead-time insertion between LS and HS switching. The DTI output will override the normal port value according to the port override setting.

The pattern generation unit can be used to generate a synchronized bit pattern on the port it is connected to. In addition, the WG output from compare channel A can be distributed to and override all the port pins. When the pattern generator unit is enabled, the DTI unit is bypassed.

The fault protection unit is connected to the event system, enabling any event to trigger a fault condition that will disable the AWeX output. The event system ensures predictable and instant fault reaction, and gives great flexibility in the selection of fault triggers.

The AWeX is available for TCC0 and TCE0. The notation of these are AWEXC and AWEXE.

18. Hi-Res - High Resolution Extension

18.1 Features

- Increases Waveform Generator resolution by 2-bits (4x)
- Supports Frequency, single- and dual-slope PWM operation
- Supports the AWeX when this is enabled and used for the same Timer/Counter

18.2 Overview

TThe high-resolution (hi-res) extension can be used to increase the resolution of the waveform generation output from a timer/counter by four or eight. It can be used for a timer/counter doing frequency, single-slope PWM, or dual-slope PWM generation. It can also be used with the AWeX if this is used for the same timer/counter.

The hi-res extension uses the peripheral 4x clock (ClkPER4). The system clock prescalers must be configured so the peripheral 4x clock frequency is four times higher than the peripheral and CPU clock frequency when the hi-res extension is enabled.

There are four hi-res extensions that each can be enabled for each timer/counters pair on PORTC, PORTD, PORTE and PORTF. The notation of these peripherals are HIRESC, HIRESD, HIRESE and HIRESF, respectively.

19. RTC - 16-bit Real-Time Counter

19.1 Features

- 16-bit resolution
- Selectable clock source
  - 32.768kHz external crystal
  - External clock
  - 32.768kHz internal oscillator
  - 32kHz internal ULP oscillator
- Programmable 10-bit clock prescaling
- One compare register
- One period register
- Clear counter on period overflow
- Optional interrupt/event on overflow and compare match

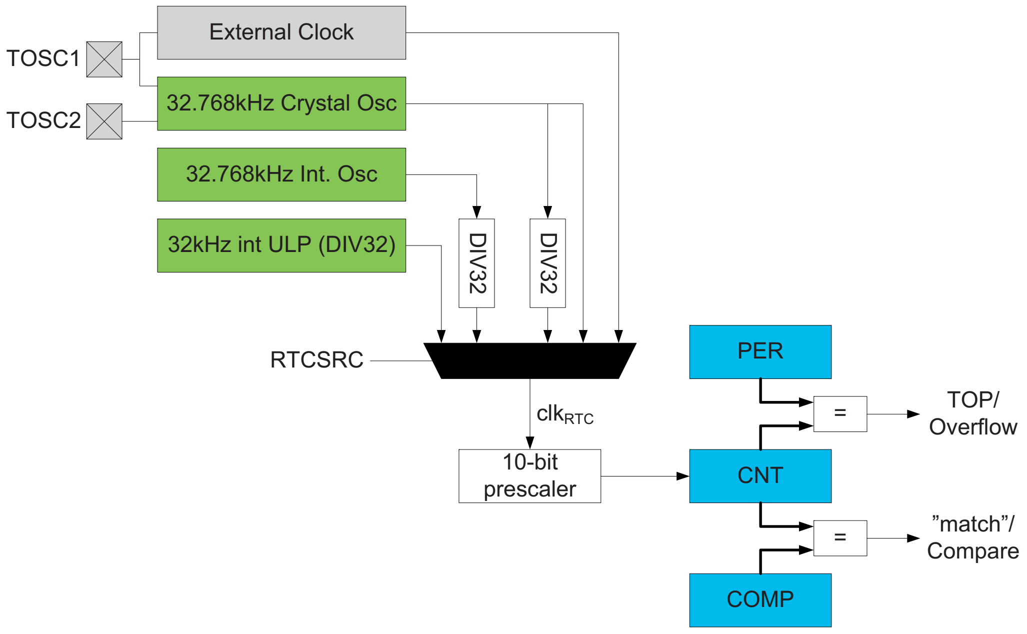
19.2 Overview

The 16-bit real-time counter (RTC) is a counter that typically runs continuously, including in low-power sleep modes, to keep track of time. It can wake up the device from sleep modes and/or interrupt the device at regular intervals.

The reference clock is typically the 1.024kHz output from a high-accuracy crystal of 32.768kHz, and this is the configuration most optimized for low power consumption. The faster 32.768kHz output can be selected if the RTC needs a resolution higher than 1ms. The RTC can also be clocked from an external clock signal, the 32.768kHz internal oscillator or the 32kHz internal ULP oscillator.

The RTC includes a 10-bit programmable prescaler that can scale down the reference clock before it reaches the counter. A wide range of resolutions and time-out periods can be configured. With a 32.768kHz clock source, the maximum resolution is 30.5µs, and time-out periods can range up to 2000 seconds. With a resolution of 1s, the maximum timeout period is more than18 hours (65536 seconds). The RTC can give a compare interrupt and/or event when the counter equals the compare register value, and an overflow interrupt and/or event when it equals the period register value.

Figure 19-1. Real Time Counter overview



20. TWI - Two-Wire Interface

20.1 Features

- Four identical two-wire interface peripherals
- Bidirectional two-wire communication interface
  - Phillips I2C compatible
  - System Management Bus (SMBus) compatible
- Bus master and slave operation supported
  - Slave operation
  - Single bus master operation
  - Bus master in multi-master bus environment
  - Multi-master arbitration
- Flexible slave address match functions
  - 7-bit and general call address recognition in hardware
  - 10-bit addressing supported
  - Address mask register for dual address match or address range masking
  - Optional software address recognition for unlimited number of addresses
- Slave can operate in all sleep modes, including power-down
- Slave address match can wake device from all sleep modes, including power-down
- 100kHz and 400kHz bus frequency support
- Slew-rate limited output drivers
- Input filter for bus noise and spike suppression
- Support arbitration between start repeated start and data bit (SMBus)
- Slave arbitration allows support for address resolve protocol (ARP) (SMBus)

20.2 Overview

The two-wire interface (TWI) is a bidirectional, two-wire communication interface. It is I2C and System Management Bus (SMBus) compatible. The only external hardware needed to implement the bus is one pull-up resistor on each bus line.

A device connected to the bus must act as a master or a slave. The master initiates a data transaction by addressing a slave on the bus and telling whether it wants to transmit or receive data. One bus can have many slaves and one or several masters that can take control of the bus. An arbitration process handles priority if more than one master tries to transmit data at the same time. Mechanisms for resolving bus contention are inherent in the protocol.

The TWI module supports master and slave functionality. The master and slave functionality are separated from each other, and can be enabled and configured separately. The master module supports multi-master bus operation and arbitration. It contains the baud rate generator. Both 100kHz and 400kHz bus frequency is supported. Quick command and smart mode can be enabled to auto-trigger operations and reduce software complexity.

The slave module implements 7-bit address match and general address call recognition in hardware. 10-bit addressing is also supported. A dedicated address mask register can act as a second address match register or as a register for address range masking. The slave continues to operate in all sleep modes, including power-down mode. This enables the slave to wake up the device from all sleep modes on TWI address match. It is possible to disable the address matching to let this be handled in software instead.

The TWI module will detect START and STOP conditions, bus collisions, and bus errors. Arbitration lost, errors, collision, and clock hold on the bus are also detected and indicated in separate status flags available in both master and slave modes.

It is possible to disable the TWI drivers in the device, and enable a four-wire digital interface for connecting to an external TWI bus driver. This can be used for applications where the device operates from a different VCC voltage than used by the TWI bus.

PORTC, PORTD, PORTE, and PORTF each has one TWI. Notation of these peripherals are TWIC, TWID, TWIE, and TWIF.

21. SPI - Serial Peripheral Interface

21.1 Features

- Four identical SPI peripherals
- Full-duplex, three-wire synchronous data transfer
- Master or slave operation
- Lsb first or msb first data transfer
- Eight programmable bit rates
- Interrupt flag at the end of transmission
- Write collision flag to indicate data collision
- Wake up from idle sleep mode
- Double speed master mode

21.2 Overview

The Serial Peripheral Interface (SPI) is a high-speed synchronous data transfer interface using three or four pins. It allows fast communication between an Atmel AVR XMEGA device and peripheral devices or between several microcontrollers. The SPI supports full-duplex communication.

A device connected to the bus must act as a master or slave. The master initiates and controls all data transactions.

PORTC, PORTD, PORTE, and PORTF each has one SPI. Notation of these peripherals are SPIC, SPID, SPIE, and SPIF.

22. USART

22.1 Features

- Eight identical USART peripherals
- Full-duplex operation
- Asynchronous or synchronous operation
  - Synchronous clock rates up to 1/2 of the device clock frequency
  - Asynchronous clock rates up to 1/8 of the device clock frequency
- Supports serial frames with 5, 6, 7, 8, or 9 data bits and 1 or 2 stop bits
- Fractional baud rate generator
  - Can generate desired baud rate from any system clock frequency
  - No need for external oscillator with certain frequencies
- Built-in error detection and correction schemes
  - Odd or even parity generation and parity check
  - Data overrun and framing error detection
  - Noise filtering includes false start bit detection and digital low-pass filter
- Separate interrupts for
  - Transmit complete
  - Transmit data register empty
  - Receive complete
- Multiprocessor communication mode
  - Addressing scheme to address a specific devices on a multidevice bus
  - Enable unaddressed devices to automatically ignore all frames
- Master SPI mode
  - Double buffered operation
  - Operation up to 1/2 of the peripheral clock frequency
- IRCOM module for IrDA compliant pulse modulation/demodulation

22.2 Overview

The universal synchronous and asynchronous serial receiver and transmitter (USART) is a fast and flexible serial communication module. The USART supports full-duplex communication and asynchronous and synchronous operation. The USART can be configured to operate in SPI master mode and used for SPI communication.

Communication is frame based, and the frame format can be customized to support a wide range of standards. The USART is buffered in both directions, enabling continued data transmission without any delay between frames. Separate interrupts for receive and transmit complete enable fully interrupt driven communication. Frame error and buffer overflow are detected in hardware and indicated with separate status flags. Even or odd parity generation and parity check can also be enabled.

The clock generator includes a fractional baud rate generator that is able to generate a wide range of USART baud rates from any system clock frequencies. This removes the need to use an external crystal oscillator with a specific frequency to achieve a required baud rate. It also supports external clock input in synchronous slave operation.

When the USART is set in master SPI mode, all USART-specific logic is disabled, leaving the transmit and receive buffers, shift registers, and baud rate generator enabled. Pin control and interrupt generation are identical in both modes. The registers are used in both modes, but their functionality differs for some control settings.

An IRCOM module can be enabled for one USART to support IrDA 1.4 physical compliant pulse modulation and demodulation for baud rates up to 115.2Kbps.

PORTC, PORTD, PORTE, and PORTF each has two USARTs. Notation of these peripherals are USARTC0, USARTC1, USARTD0, USARTD1, USARTE0, USARTE1, USARTF0 and USARTF1.

23. IRCOM - IR Communication Module

23.1 Features

- Pulse modulation/demodulation for infrared communication
- IrDA compatible for baud rates up to 115.2Kbps
- Selectable pulse modulation scheme
  - 3/16 of the baud rate period
  - Fixed pulse period, 8-bit programmable
  - Pulse modulation disabled
- Built-in filtering
- Can be connected to and used by any USART

23.2 Overview

Atmel AVR XMEGA devices contain an infrared communication module (IRCOM) that is IrDA compatible for baud rates up to 115.2Kbps. It can be connected to any USART to enable infrared pulse encoding/decoding for that USART.

24. AES and DES Crypto Engine

24.1 Features

- Data Encryption Standard (DES) CPU instruction
- Advanced Encryption Standard (AES) crypto module
- DES Instruction
  - Encryption and decryption
  - DES supported
  - Encryption/decryption in 16 CPU clock cycles per 8-byte block
- AES crypto module
  - Encryption and decryption
  - Supports 128-bit keys
  - Supports XOR data load mode to the state memory
  - Encryption/decryption in 375 clock cycles per 16-byte block

24.2 Overview

The Advanced Encryption Standard (AES) and Data Encryption Standard (DES) are two commonly used standards for cryptography. These are supported through an AES peripheral module and a DES CPU instruction, and the communication interfaces and the CPU can use these for fast, encrypted communication and secure data storage.

DES is supported by an instruction in the AVR CPU. The 8-byte key and 8-byte data blocks must be loaded into the register file, and then the DES instruction must be executed 16 times to encrypt/decrypt the data block.

The AES crypto module encrypts and decrypts 128-bit data blocks with the use of a 128-bit key. The key and data must be loaded into the key and state memory in the module before encryption/decryption is started. It takes 375 peripheral clock cycles before the encryption/decryption is done. The encrypted/encrypted data can then be read out, and an optional interrupt can be generated. The AES crypto module also has DMA support with transfer triggers when encryption/decryption is done and optional auto-start of encryption/decryption when the state memory is fully loaded.

25. EBI – External Bus Interface

25.1 Features

- Supports SRAM up to:
  - 512KB using 3-port EBI configuration
  - 16MB using 3-port EBI configuration
- Supports SDRAM up to:
  - 128Mb using 3-port EBI configuration
- Four software configurable chip selects
- Software configurable wait state insertion
- Can run from the 2x peripheral clock frequency for fast access

25.2 Overview

The External Bus Interface (EBI) is used to connect external peripherals and memory for access through the data memory space. When the EBI is enabled, data address space outside the internal SRAM becomes available using dedicated EBI pins.

The EBI can interface external SRAM, SDRAM, and peripherals, such as LCD displays and other memory mapped devices.

The address space for the external memory is selectable from 256 bytes (8-bit) up to 16MB (24-bit). Various multiplexing modes for address and data lines can be selected for optimal use of pins when more or fewer pins are available for the EBI. The complete memory will be mapped into one linear data address space continuing from the end of the internal SRAM.

The EBI has four chip selects, each with separate configuration. Each can be configured for SRAM, SRAM low pin count (LPC), or SDRAM.

The EBI is clocked from the fast, 2x peripheral clock, running up to two times faster than the CPU.

Four-bit and eight-bit SDRAM are supported, and SDRAM configurations, such as CAS latency and refresh rate, are configurable in software.

26. ADC - 12-bit Analog to Digital Converter

26.1 Features

- Two ADCs with 12-bit resolution
- 2Msps sample rate for each ADC
- Signed and unsigned conversions
- 4 result registers with individual input channel control for each ADC
- 8 single ended inputs for each ADC
- 8x4 differential inputs for each ADC
- 4 internal inputs:
  - Integrated Temperature Sensor
  - DAC Output
  - VCC voltage divided by 10
  - Bandgap voltage
- Software selectable gain of 2, 4, 8, 16, 32 or 64
- Software selectable resolution of 8- or 12-bit.
- Internal or External Reference selection
- Event triggered conversion for accurate timing
- DMA transfer of conversion results
- Interrupt/Event on compare result

26.2 Overview

XMEGA A1 devices have two Analog to Digital Converters (ADC), see Figure 26-1 on page 49. The two ADC modules can be operated simultaneously, individually or synchronized.

The ADC converts analog voltages to digital values. The ADC has 12-bit resolution and is capable of converting up to 2 million samples per second. The input selection is flexible, and both single-ended and differential measurements can be done. For differential measurements an optional gain stage is available to increase the dynamic range. In addition several internal signal inputs are available. The ADC can provide both signed and unsigned results.

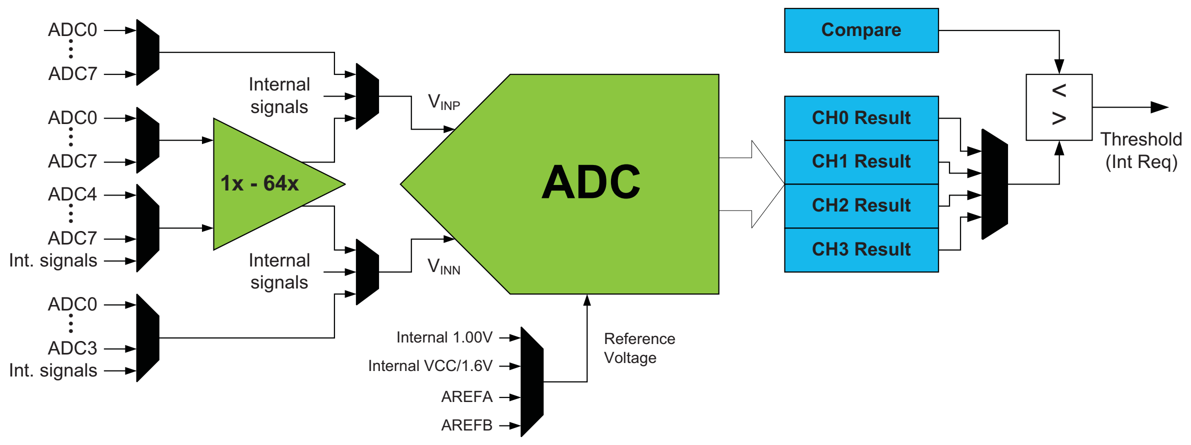
This is a pipeline ADC. A pipeline ADC consists of several consecutive stages, where each stage convert one part of the result. The pipeline design enables high sample rate at low clock speeds, and remove limitations on samples speed versus propagation delay. This also means that a new analog voltage can be sampled and a new ADC measurement started while other ADC measurements are ongoing.

ADC measurements can either be started by application software or an incoming event from another peripheral in the device. Four different result registers with individual input selection (MUX selection) are provided to make it easier for the application to keep track of the data. Each result register and MUX selection pair is referred to as an ADC Channel. It is possible to use DMA to move ADC results directly to memory or peripherals when conversions are done.

Both internal and external analog reference voltages can be used. An accurate internal 1.0V reference is available.

An integrated temperature sensor is available and the output from this can be measured with the ADC. The output from the DAC, VCC/10 and the Bandgap voltage can also be measured by the ADC.

Figure 26-1. ADC overview



Each ADC has four MUX selection registers with a corresponding result register. This means that four channels can be sampled within 1.5 µs without any intervention by the application other than starting the conversion. The results will be available in the result registers.

The ADC may be configured for 8- or 12-bit result, reducing the minimum conversion time (propagation delay) from 3.5 µs for 12-bit to 2.5 µs for 8-bit result.

ADC conversion results are provided left- or right adjusted with optional ‘1’ or ‘0’ padding. This eases calculation when the result is represented as a signed integer (signed 16-bit number).

PORTA and PORTB each has one ADC. Notation of these peripherals are ADCA and ADCB, respectively.

27. DAC - 12-bit Digital to Analog Converter

27.1 Features

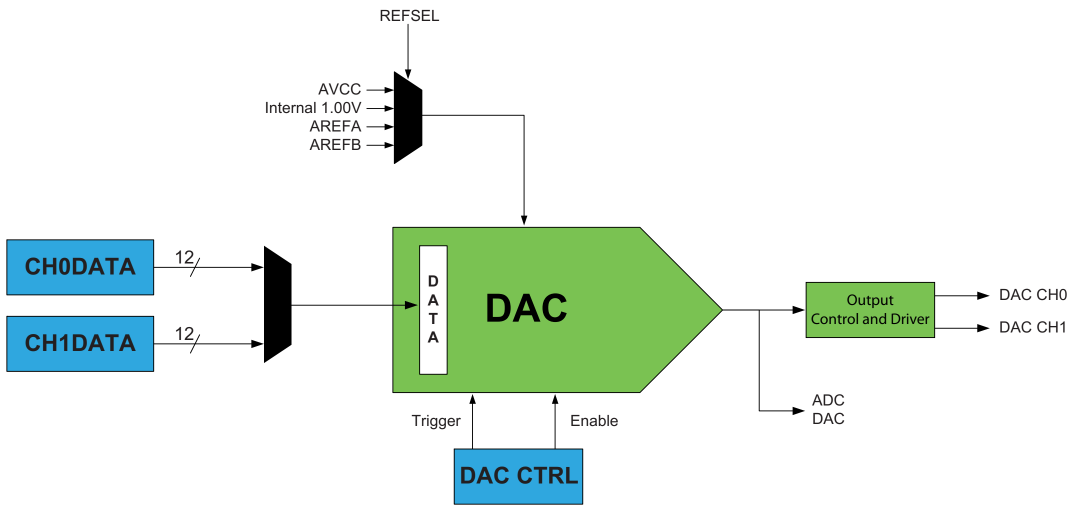
- 12-bit resolution
- Two independent, continuous-drive output channels
- Up to one million samples per second conversion rate
- Built-in calibration that removes:
  - Offset error
  - Gain error
- Multiple conversion trigger sources
  - On new available data
  - Events from the event system
- High drive capabilities and support for
  - Resistive loads
  - Capacitive loads
  - Combined resistive and capacitive loads
- Internal and external reference options
- DAC output available as input to analog comparator and ADC
- Low-power mode, with reduced drive strength
- Optional DMA transfer of data

27.2 Overview

The XMEGA A1 devices features two 12-bit, 1 Msps DACs with built-in offset and gain calibration, see Figure 27-1 on page 50.

A DAC converts a digital value into an analog signal. The DAC may use an internal 1.0 voltage as the upper limit for conversion, but it is also possible to use the supply voltage or any applied voltage in-between. The external reference input is shared with the ADC reference input.

Figure 27-1. DAC overview



Each DAC has one continuous output with high drive capabilities for both resistive and capacitive loads. It is also possible to split the continuous time channel into two Sample and Hold (S/H) channels, each with separate data conversion registers.

A DAC conversion may be started from the application software by writing the data conversion registers. The DAC can also be configured to do conversions triggered by the Event System to have regular timing, independent of the application software. DMA may be used for transferring data from memory locations to DAC data registers.

The DAC has a built-in calibration system to reduce offset and gain error when loading with a calibration value from software.

PORTA and PORTB each has one DAC. Notation of these peripherals are DACA and DACB. respectively.

28. AC - Analog Comparator

28.1 Features

- Four Analog Comparators
- Selectable propagation delay versus current consumption
- Selectable hysteresis
  - No
  - Small
  - Large
- Analog comparator output available on pin
- Flexible input selection
  - All pins on the port
  - Output from the DAC
  - Bandgap reference voltage
  - A 64-level programmable voltage scaler of the internal VCC voltage
- Interrupt and event generation on:
  - Rising edge
  - Falling edge
  - Toggle
- Window function interrupt and event generation on:
  - Signal above window
  - Signal inside window
  - Signal below window
- Constant current source with configurable output pin selection

28.2 Overview

The analog comparator (AC) compares the voltage levels on two inputs and gives a digital output based on this comparison. The analog comparator may be configured to generate interrupt requests and/or events upon several different combinations of input change.

Two important properties of the analog comparator's dynamic behavior are: hysteresis and propagation delay. Both of these parameters may be adjusted in order to achieve the optimal operation for each application.

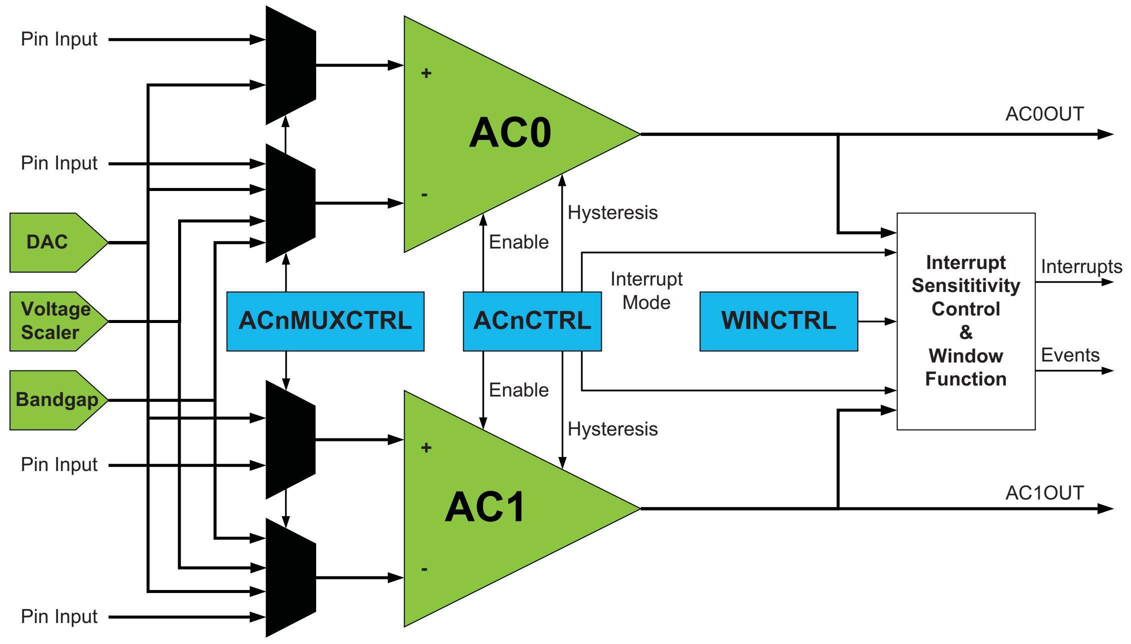
The input selection includes analog port pins, several internal signals, and a 64-level programmable voltage scaler. The analog comparator output state can also be output on a pin for use by external devices.

A constant current source can be enabled and output on a selectable pin. This can be used to replace, for example, external resistors used to charge capacitors in capacitive touch sensing applications.

The analog comparators are always grouped in pairs on each port. These are called analog comparator 0 (AC0) and analog comparator 1 (AC1). They have identical behavior, but separate control registers. Used as pair, they can be set in window mode to compare a signal to a voltage range instead of a voltage level.

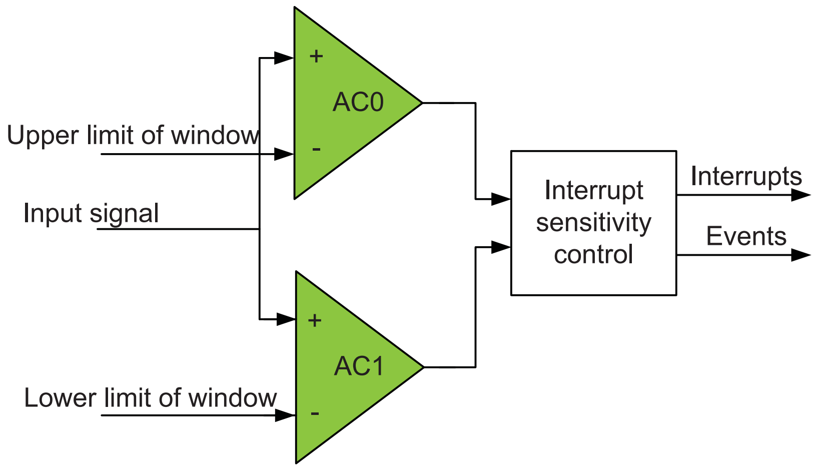
PORTA and PORTB each has one AC pair. Notations are ACA and ACB, respectively.

Figure 28-1. Analog comparator overview



The window function is realized by connecting the external inputs of the two analog comparators in a pair as shown in Figure 28-2.

Figure 28-2. Analog comparator window function



29. Programming and Debugging

29.1 Features

- Programming
  - External programming through PDI or JTAG interfaces
    - Minimal protocol overhead for fast operation
    - Built-in error detection and handling for reliable operation
  - Boot loader support for programming through any communication interface
- Debugging
  - Nonintrusive, real-time, on-chip debug system
  - No software or hardware resources required from device except pin connection
  - Program flow control
    - Go, Stop, Reset, Step Into, Step Over, Step Out, Run-to-Cursor
  - Unlimited number of user program breakpoints
  - Unlimited number of user data breakpoints, break on:
    - Data location read, write, or both read and write
    - Data location content equal or not equal to a value
    - Data location content is greater or smaller than a value
    - Data location content is within or outside a range
  - No limitation on device clock frequency
- Program and Debug Interface (PDI)
  - Two-pin interface for external programming and debugging
  - Uses the Reset pin and a dedicated pin
  - No I/O pins required during programming or debugging
- JTAG interface
  - Four-pin, IEEE Std. 1149.1 compliant interface for programming and debugging
  - Boundary scan capabilities according to IEEE Std. 1149.1 (JTAG)

29.2 Overview

The Program and Debug Interface (PDI) is an Atmel proprietary interface for external programming and on-chip debugging of a device.

The PDI supports fast programming of nonvolatile memory (NVM) spaces; flash, EEPOM, fuses, lock bits, and the user signature row.

Debug is supported through an on-chip debug system that offers nonintrusive, real-time debug. It does not require any software or hardware resources except for the device pin connection. Using the Atmel tool chain, it offers complete program flow control and support for an unlimited number of program and complex data breakpoints. Application debug can be done from a C or other high-level language source code level, as well as from an assembler and disassembler level.

Programming and debugging can be done through two physical interfaces. The primary one is the PDI physical layer, which is available on all devices. This is a two-pin interface that uses the Reset pin for the clock input (PDI_CLK) and one other dedicated pin for data input and output (PDI_DATA). A JTAG interface is also available on most devices, and this can be used for programming and debugging through the four-pin JTAG interface. The JTAG interface is IEEE Std. 1149.1 compliant, and supports boundary scan. Any external programmer or on-chip debugger/emulator can be directly connected to either of these interfaces. Unless otherwise stated, all references to the PDI assume access through the PDI physical layer.

30. Pinout and Pin Functions

The pinout of XMEGA A1 is shown in "Pinout/Block Diagram" on page 3. In addition to general I/O functionality, each pin may have several functions. This will depend on which peripheral is enabled and connected to the actual pin. Only one of the alternate pin functions can be used at time.

30.1 Alternate Pin Function Description

The tables below shows the notation for all pin functions available and describes its function.

30.1.1 Operation/Power Supply

V_{CC}	Digital supply voltage
AV_{CC}	Analog supply voltage
GND	Ground

30.1.2 Port Interrupt functions

SYNC	Port pin with full synchronous and limited asynchronous interrupt function
ASYNC	Port pin with full synchronous and full asynchronous interrupt function

30.1.3 Analog functions

ACn	Analog Comparator input pin n
AC0OUT	Analog Comparator 0 Output
ADCn	Analog to Digital Converter input pin n
DACn	Digital to Analog Converter output pin n
AREF	Analog Reference input pin

30.1.4 EBI functions

An	Address line n	
Dn	Data line n	
$\overline{\text{CSn}}$	Chip Select n	
ALEn	Address Latch Enable pin n	(SRAM)
$\overline{\text{RE}}$	Read Enable	(SRAM)
$\overline{\text{WE}}$	External Data Memory Write	(SRAM /SDRAM)
BAn	Bank Address	(SDRAM)
$\overline{\text{CAS}}$	Column Access Strobe	(SDRAM)
CKE	SDRAM Clock Enable	(SDRAM)

CLK	SDRAM Clock	(SDRAM)
$\overline{\text{DQM}}$	Data Mask Signal/Output Enable	(SDRAM)
$\overline{\text{RAS}}$	Row Access Strobe	(SDRAM)
2P	2 Port Interface	
3P	3 Port Interface	

30.1.5 Timer/Counter and AWEX functions

OCnx	Output Compare Channel x for Timer/Counter n
$\overline{\text{OCnx}}$	Inverted Output Compare Channel x for Timer/Counter n
OCnxLS	Output Compare Channel x Low Side for Timer/Counter n
OCnxHS	Output Compare Channel x High Side for Timer/Counter n

30.1.6 Communication functions

SCL	Serial Clock for TWI
SDA	Serial Data for TWI
SCLIN	Serial Clock In for TWI when external driver interface is enabled
SCLOUT	Serial Clock Out for TWI when external driver interface is enabled
SDAIN	Serial Data In for TWI when external driver interface is enabled
SDAOUT	Serial Data Out for TWI when external driver interface is enabled
XCKn	Transfer Clock for USART n
RXDn	Receiver Data for USART n
TXDn	Transmitter Data for USART n
$\overline{\text{SS}}$	Slave Select for SPI
MOSI	Master Out Slave In for SPI
MISO	Master In Slave Out for SPI
SCK	Serial Clock for SPI

30.1.7 Oscillators, Clock and Event

n	Timer Oscillator pin n
XTALn	Input/Output for inverting Oscillator pin n
CLKOUT	Peripheral Clock Output
EVOUT	Event Channel 0 Output

30.1.8 Debug/System functions

$\overline{\text{RESET}}$	Reset pin
PDI_CLK	Program and Debug Interface Clock pin
PDI_DATA	Program and Debug Interface Data pin
TCK	JTAG Test Clock
TDI	JTAG Test Data In
TDO	JTAG Test Data Out
TMS	JTAG Test Mode Select

30.2 Alternate Pin Functions

The tables below show the primary/default function for each pin on a port in the first column, the pin number in the second column, and then all alternate pin functions in the remaining columns. The head row shows what peripheral that enable and use the alternate pin functions.

Table 30-1. Port A - Alternate functions.

PORT A	PIN #	INTERRUPT	ADCA POS	ADCA NEG	ADCA GAINPOS	ADCA GAINNEG	ACA POS	ACA NEG	ACA OUT	DACA	REFA
GND	93										
AVCC	94										
PA0	95	SYNC	ADC0	ADC0	ADC0		AC0	AC0			AREF
PA1	96	SYNC	ADC1	ADC1	ADC1		AC1	AC1			
PA2	97	SYNC/ASYNC	ADC2	ADC2	ADC2		AC2			DAC0	
PA3	98	SYNC	ADC3	ADC3	ADC3		AC3	AC3		DAC1	
PA4	99	SYNC	ADC4		ADC4	ADC4	AC4				
PA5	100	SYNC	ADC5		ADC5	ADC5	AC5	AC5			
PA6	1	SYNC	ADC6		ADC6	ADC6	AC6				
PA7	2	SYNC	ADC7		ADC7	ADC7		AC7	AC0OUT		

Table 30-2. Port B - Alternate functions.

PORT B	PIN #	INTERRUPT	ADCB POS	ADCB NEG	ADCB GAINPOS	ADCB GAINNEG	ACB POS	ACB NEG	ACB OUT	DACB	REFB	JTAG
GND	3											
AVCC	4											
PB0	5	SYNC	ADC0	ADC0	ADC0		AC0	AC0			AREF	
PB1	6	SYNC	ADC1	ADC1	ADC1		AC1	AC1				
PB2	7	SYNC/ASYNC	ADC2	ADC2	ADC2		AC2			DAC0		
PB3	8	SYNC	ADC3	ADC3	ADC3		AC3	AC3		DAC1		
PB4	9	SYNC	ADC4		ADC4	ADC4	AC4					TMS
PB5	10	SYNC	ADC5		ADC5	ADC5	AC5	AC5				TDI
PB6	11	SYNC	ADC6		ADC6	ADC6	AC6					TCK
PB7	12	SYNC	ADC7		ADC7	ADC7		AC7	AC0OUT			TDO

Table 30-3. Port C - Alternate functions.

PORT C	PIN #	INTERRUPT	TCC0	AWEXC	TCC1	USARTC0	USARTC1	SPIC	TWIC	CLOCKOUT	EVENTOUT
GND	13										
VCC	14										
PC0	15	SYNC	OC0A	OC0ALS					SDA		
PC1	16	SYNC	OC0B	OC0AHS		XCK0			SCL		
PC2	17	SYNC/ASYNC	OC0C	OC0BLS		RXD0					

PORT C	PIN #	INTERRUPT	TCC0	AWEXC	TCC1	USARTC0	USARTC1	SPIC	TWIC	CLOCKOUT	EVENTOUT
PC3	18	SYNC	OC0D	OC0BHS		TXD0					
PC4	19	SYNC		OC0CLS	OC1A			$\overline{\text{SS}}$			
PC5	20	SYNC		OC0CHS	OC1B		XCK1	MOSI			
PC6	21	SYNC		OC0DLS			RXD1	MISO			
PC7	22	SYNC		OC0DHS			TXD1	SCK		CLKOUT	EVOUT

Table 30-4. Port D - Alternate functions.

PORT D	PIN #	INTERRUPT	TCD0	TCD1	USARTD0	USARTD1	SPID	TWID	CLOCKOUT	EVENTOUT
GND	23									
VCC	24									
PD0	25	SYNC	OC0A					SDA		
PD1	26	SYNC	OC0B		XCK0			SCL		
PD2	27	SYNC/ASYNC	OC0C		RXD0					
PD3	28	SYNC	OC0D		TXD0					
PD4	29	SYNC		OC1A			$\overline{\text{SS}}$			
PD5	30	SYNC		OC1B		XCK1	MOSI			
PD6	31	SYNC				RXD1	MISO			
PD7	32	SYNC				TXD1	SCK		CLKOUT	EVOUT

Table 30-5. Port E - Alternate functions.

PORT E	PIN #	INTERRUPT	TCE0	AWEXE	TCE1	USARTE0	USARTE1	SPIE	TWIE	CLOCKOUT	EVENTOUT
GND	33										
VCC	34										
PE0	35	SYNC	OC0A	OC0ALS					SDA		
PE1	36	SYNC	OC0B	OC0AHS		XCK0			SCL		
PE2	37	SYNC/ASYNC	OC0C	OC0BLS		RXD0					
PE3	38	SYNC	OC0D	OC0BHS		TXD0					
PE4	39	SYNC		OC0CLS	OC1A			$\overline{\text{SS}}$			
PE5	40	SYNC		OC0CHS	OC1B		XCK1	MOSI			
PE6	41	SYNC		OC0DLS			RXD1	MISO			
PE7	42	SYNC		OC0DHS			TXD1	SCK		CLKOUT	EVOUT

Table 30-6. Port F - Alternate functions.

PORT F	PIN #	INTERRUPT	TCF0	TCF1	USARTF0	USARTF1	SPIF	TWIF
GND	43							
VCC	44							
PF0	45	SYNC	OC0A					SDA

PORT F	PIN #	INTERRUPT	TCF0	TCF1	USARTF0	USARTF1	SPIF	TWIF
PF1	46	SYNC	OC0B		XCK0			SCL
PF2	47	SYNC/ASYNC	OC0C		RXD0			
PF3	48	SYNC	OC0D		TXD0			
PF4	49	SYNC		OC1A			$\overline{SS}$	
PF5	50	SYNC		OC1B		XCK1	MOSI	
PF6	51	SYNC				RXD1	MISO	
PF7	52	SYNC				TXD1	SCK	

Table 30-7. Port H - Alternate functions.

PORT H	PIN #	INTERRUPT	SDRAM 3P	SRAM ALE1 3P	SRAM ALE12 3P	LPC ALE1 3P	LPC ALE1 2P	LPC ALE12 2P
GND	53							
VCC	54							
PH0	55	SYNC	$\overline{WE}$	$\overline{WE}$	$\overline{WE}$	$\overline{WE}$	$\overline{WE}$	$\overline{WE}$
PH1	56	SYNC	$\overline{CAS}$	$\overline{RE}$	$\overline{RE}$	$\overline{RE}$	$\overline{RE}$	$\overline{RE}$
PH2	57	SYNC/ASYNC	$\overline{RAS}$	ALE1	ALE1	ALE1	ALE1	ALE1
PH3	58	SYNC	$\overline{DQM}$		ALE2			ALE2
PH4	59	SYNC	BA0	$\overline{CS0}$/A16	$\overline{CS0}$	$\overline{CS0}$/A16	$\overline{CS0}$	$\overline{CS0}$/A16
PH5	60	SYNC	BA1	$\overline{CS1}$/A17	$\overline{CS1}$	$\overline{CS1}$/A17	$\overline{CS1}$	$\overline{CS1}$/A17
PH6	61	SYNC	CKE	$\overline{CS2}$/A18	$\overline{CS2}$	$\overline{CS2}$/A18	$\overline{CS2}$	$\overline{CS2}$/A18
PH7	62	SYNC	CLK	$\overline{CS3}$/A19	$\overline{CS3}$	$\overline{CS3}$/A19	$\overline{CS3}$	$\overline{CS3}$/A19

Table 30-8. Port J - Alternate functions.

PORT J	PIN #	INTERRUPT	SDRAM 3P	SRAM ALE1 3P	SRAM ALE12 3P	LPC ALE1 3P	LPC ALE1 2P	LPC ALE12 2P
GND	63							
VCC	64							
PJ0	65	SYNC	D0	D0	D0	D0/A0	D0/A0	D0/A0/A8
PJ1	66	SYNC	D1	D1	D1	D1/A1	D1/A1	D1/A1/A9
PJ2	67	SYNC/ASYNC	D2	D2	D2	D2/A2	D2/A2	D2/A2/A10
PJ3	68	SYNC	D3	D3	D3	D3/A3	D3/A3	D3/A3/A11
PJ4	69	SYNC	A8	D4	D4	D4/A4	D4/A4	D4/A4/A12
PJ5	70	SYNC	A9	D5	D5	D5/A5	D5/A5	D5/A5/A13
PJ6	71	SYNC	A10	D6	D6	D6/A6	D6/A6	D6/A6/A14
PJ7	72	SYNC	A11	D7	D7	D7/A7	D7/A7	D7/A7/A15

Table 30-9. Port K - Alternate functions.

PORT K	PIN #	INTERRUPT	SDRAM 3P	SRAM ALE1 3P	SRAM ALE12 3P	LPC ALE1 3P	LPC ALE1 2P	LPC ALE12 2P
GND	73							
VCC	74							
PK0	75	SYNC	A0	A0/A8	A0/A8/A16	A8		
PK1	76	SYNC	A1	A1/A9	A1/A9/A17	A9		
PK2	77	SYNC/ASYNC	A2	A2/A10	A2/A10/A18	A10		
PK3	78	SYNC	A3	A3/A11	A3/A11/A19	A11		
PK4	79	SYNC	A4	A4/A12	A4/A12/A20	A12		
PK5	80	SYNC	A5	A5/A13	A5/A13/A21	A13		
PK6	81	SYNC	A6	A6/A14	A6/A14/A22	A14		
PK7	82	SYNC	A7	A7/A15	A7/A15/A23	A15		

Table 30-10. Port Q - Alternate functions.

PORT Q	PIN #	INTERRUPT	
VCC	83		
GND	84		
PQ0	85	SYNC	TOSC1 (Input)
PQ1	86	SYNC	TOSC2 (Output)
PQ2	87	SYNC/ASYNC	
PQ3	88	SYNC	

Table 30-11. Port R - Alternate functions.

PORT R	PIN #	INTERRUPT	PDI	XTAL
PDI	89		PDI_DATA	
$\overline{\text{RESET}}$	90		PDI_CLOCK	
PRO	91	SYNC		XTAL2
PR1	92	SYNC		XTAL1

31. Peripheral Module Address Map

The address maps show the base address for each peripheral and module in XMEGA A1. For complete register description and summary for each peripheral module, refer to the XMEGA A Manual.

Table 31-1. Peripheral Module Address Map

Base Address	Name	Description
0x0000	GPIO	General Purpose IO Registers
0x0010	VPORT0	Virtual Port 0
0x0014	VPORT1	Virtual Port 1
0x0018	VPORT2	Virtual Port 2
0x001C	VPORT3	Virtual Port 3
0x0030	CPU	CPU
0x0040	CLK	Clock Control
0x0048	SLEEP	Sleep Controller
0x0050	OSC	Oscillator Control
0x0060	DFLLRC32M	DFLL for the 32 MHz Internal RC Oscillator
0x0068	DFLLRC2M	DFLL for the 2 MHz RC Oscillator
0x0070	PR	Power Reduction
0x0078	RST	Reset Controller
0x0080	WDT	Watch-Dog Timer
0x0090	MCU	MCU Control
0x00A0	PMIC	Programmable Multilevel Interrupt Controller
0x00B0	PORTCFG	Port Configuration
0x00C0	AES	AES Module
0x0100	DMA	DMA Controller
0x0180	EVSYS	Event System
0x01C0	NVM	Non Volatile Memory (NVM) Controller
0x0200	ADCA	Analog to Digital Converter on port A
0x0240	ADCB	Analog to Digital Converter on port B
0x0300	DACA	Digital to Analog Converter on port A
0x0320	DACB	Digital to Analog Converter on port B
0x0380	ACA	Analog Comparator pair on port A
0x0390	ACB	Analog Comparator pair on port B
0x0400	RTC	Real Time Counter
0x0440	EBI	External Bus Interface
0x0480	TWIC	Two Wire Interface on port C
0x0490	TWID	Two Wire Interface on port D

Base Address	Name	Description
0x04A0	TWIE	Two Wire Interface on port E
0x04B0	TWIF	Two Wire Interface on port F
0x0600	PORTA	Port A
0x0620	PORTB	Port B
0x0640	PORTC	Port C
0x0660	PORTD	Port D
0x0680	PORTE	Port E
0x06A0	PORTF	Port F
0x06E0	PORTH	Port H
0x0700	PORTJ	Port J
0x0720	PORTK	Port K
0x07C0	PORTQ	Port Q
0x07E0	PORTR	Port R
0x0800	TCC0	Timer/Counter 0 on port C
0x0840	TCC1	Timer/Counter 1 on port C
0x0880	AWEXC	Advanced Waveform Extension on port C
0x0890	HIRESC	High Resolution Extension on port C
0x08A0	USARTC0	USART 0 on port C
0x08B0	USARTC1	USART 1 on port C
0x08C0	SPIC	Serial Peripheral Interface on port C
0x08F8	IRCOM	Infrared Communication Module
0x0900	TCD0	Timer/Counter 0 on port D
0x0940	TCD1	Timer/Counter 1 on port D
0x0990	HIRESD	High Resolution Extension on port D
0x09A0	USARTD0	USART 0 on port D
0x09B0	USARTD1	USART 1 on port D
0x09C0	SPID	Serial Peripheral Interface on port D
0x0A00	TCE0	Timer/Counter 0 on port E
0x0A40	TCE1	Timer/Counter 1 on port E
0x0A80	AWEXE	Advanced Waveform Extension on port E
0x0A90	HIRESE	High Resolution Extension on port E
0x0AA0	USARTE0	USART 0 on port E
0x0AB0	USARTE1	USART 1 on port E
0x0AC0	SPIE	Serial Peripheral Interface on port E
0x0B00	TCF0	Timer/Counter 0 on port F

Base Address	Name	Description
0x0B40	TCF1	Timer/Counter 1 on port F
0x0B90	HIRESF	High Resolution Extension on port F
0x0BA0	USARTF0	USART 0 on port F
0x0BB0	USARTF1	USART 1 on port F
0x0BC0	SPIF	Serial Peripheral Interface on port F

32. Instruction Set Summary

Mnemonics	Operands	Description	Operation			Flags	#Clocks
Arithmetic and Logic Instructions							
ADD	Rd, Rr	Add without Carry	Rd	←	Rd + Rr	Z,C,N,V,S,H	1
ADC	Rd, Rr	Add with Carry	Rd	←	Rd + Rr + C	Z,C,N,V,S,H	1
ADIW	Rd, K	Add Immediate to Word	Rd	←	Rd + 1:Rd + K	Z,C,N,V,S	2
SUB	Rd, Rr	Subtract without Carry	Rd	←	Rd - Rr	Z,C,N,V,S,H	1
SUBI	Rd, K	Subtract Immediate	Rd	←	Rd - K	Z,C,N,V,S,H	1
SBC	Rd, Rr	Subtract with Carry	Rd	←	Rd - Rr - C	Z,C,N,V,S,H	1
SBCI	Rd, K	Subtract Immediate with Carry	Rd	←	Rd - K - C	Z,C,N,V,S,H	1
SBIW	Rd, K	Subtract Immediate from Word	Rd + 1:Rd	←	Rd + 1:Rd - K	Z,C,N,V,S	2
AND	Rd, Rr	Logical AND	Rd	←	Rd • Rr	Z,N,V,S	1
ANDI	Rd, K	Logical AND with Immediate	Rd	←	Rd • K	Z,N,V,S	1
OR	Rd, Rr	Logical OR	Rd	←	Rd v Rr	Z,N,V,S	1
ORI	Rd, K	Logical OR with Immediate	Rd	←	Rd v K	Z,N,V,S	1
EOR	Rd, Rr	Exclusive OR	Rd	←	Rd ⊕ Rr	Z,N,V,S	1
COM	Rd	One's Complement	Rd	←	$FF - Rd	Z,C,N,V,S	1
NEG	Rd	Two's Complement	Rd	←	$00 - Rd	Z,C,N,V,S,H	1
SBR	Rd,K	Set Bit(s) in Register	Rd	←	Rd v K	Z,N,V,S	1
CBR	Rd,K	Clear Bit(s) in Register	Rd	←	Rd • ($FFh - K)	Z,N,V,S	1
INC	Rd	Increment	Rd	←	Rd + 1	Z,N,V,S	1
DEC	Rd	Decrement	Rd	←	Rd - 1	Z,N,V,S	1
TST	Rd	Test for Zero or Minus	Rd	←	Rd • Rd	Z,N,V,S	1
CLR	Rd	Clear Register	Rd	←	Rd ⊕ Rd	Z,N,V,S	1
SER	Rd	Set Register	Rd	←	$FF	None	1
MUL	Rd,Rr	Multiply Unsigned	R1:R0	←	Rd x Rr (UU)	Z,C	2
MULS	Rd,Rr	Multiply Signed	R1:R0	←	Rd x Rr (SS)	Z,C	2
MULSU	Rd,Rr	Multiply Signed with Unsigned	R1:R0	←	Rd x Rr (SU)	Z,C	2
FMUL	Rd,Rr	Fractional Multiply Unsigned	R1:R0	←	Rd x Rr<<1 (UU)	Z,C	2
FMULS	Rd,Rr	Fractional Multiply Signed	R1:R0	←	Rd x Rr<<1 (SS)	Z,C	2
FMULSU	Rd,Rr	Fractional Multiply Signed with Unsigned	R1:R0	←	Rd x Rr<<1 (SU)	Z,C	2

DES	K	Data Encryption	if (H = 0) then R15:R0
			else if (H = 1) then R15:R0
				←
				←
					Encrypt(R15:R0, K)
					Decrypt(R15:R0, K)
							1/2

Branch Instructions							
RJMP	k	Relative Jump	PC	←	PC + k + 1	None	2

IJMP		Indirect Jump to (Z)	PC(15:0)
			PC(21:16)
				←
				←
					Z,
					0
						None	2

EIJMP		Extended Indirect Jump to (Z)	PC(15:0)
			PC(21:16)
				←
				←
					Z,
					EIND
						None	2

JMP	k	Jump	PC	←	k	None	3

Mnemonics	Operands	Description	Operation			Flags	#Clocks
RCALL	k	Relative Call Subroutine	PC	←	PC + k + 1	None	2 / 3(1)

ICALL		Indirect Call to (Z)	PC(15:0)
			PC(21:16)
				←
				←
					Z,
					0
						None	2 / 3(1)

EICALL		Extended Indirect Call to (Z)	PC(15:0)
			PC(21:16)
				←
				←
					Z,
					EIND
						None	3(1)

CALL	k	call Subroutine	PC	←	k	None	3 / 4(1)
RET		Subroutine Return	PC	←	STACK	None	4 / 5(1)
RETI		Interrupt Return	PC	←	STACK	I	4 / 5(1)
CPSE	Rd,Rr	Compare, Skip if Equal	if (Rd = Rr) PC	←	PC + 2 or 3	None	1 / 2 / 3
CP	Rd,Rr	Compare	Rd - Rr			Z,C,N,V,S,H	1
CPC	Rd,Rr	Compare with Carry	Rd - Rr - C			Z,C,N,V,S,H	1
CPI	Rd,K	Compare with Immediate	Rd - K			Z,C,N,V,S,H	1
SBRC	Rr, b	Skip if Bit in Register Cleared	if (Rr(b) = 0) PC	←	PC + 2 or 3	None	1 / 2 / 3
SBRS	Rr, b	Skip if Bit in Register Set	if (Rr(b) = 1) PC	←	PC + 2 or 3	None	1 / 2 / 3
SBIC	A, b	Skip if Bit in I/O Register Cleared	if (I/O(A,b) = 0) PC	←	PC + 2 or 3	None	2 / 3 / 4
SBIS	A, b	Skip if Bit in I/O Register Set	If (I/O(A,b) =1) PC	←	PC + 2 or 3	None	2 / 3 / 4
BRBS	s, k	Branch if Status Flag Set	if (SREG(s) = 1) then PC	←	PC + k + 1	None	1 / 2
BRBC	s, k	Branch if Status Flag Cleared	if (SREG(s) = 0) then PC	←	PC + k + 1	None	1 / 2
BREQ	k	Branch if Equal	if (Z = 1) then PC	←	PC + k + 1	None	1 / 2
BRNE	k	Branch if Not Equal	if (Z = 0) then PC	←	PC + k + 1	None	1 / 2
BRCS	k	Branch if Carry Set	if (C = 1) then PC	←	PC + k + 1	None	1 / 2
BRCC	k	Branch if Carry Cleared	if (C = 0) then PC	←	PC + k + 1	None	1 / 2
BRSH	k	Branch if Same or Higher	if (C = 0) then PC	←	PC + k + 1	None	1 / 2
BRLO	k	Branch if Lower	if (C = 1) then PC	←	PC + k + 1	None	1 / 2
BRMI	k	Branch if Minus	if (N = 1) then PC	←	PC + k + 1	None	1 / 2
BRPL	k	Branch if Plus	if (N = 0) then PC	←	PC + k + 1	None	1 / 2
BRGE	k	Branch if Greater or Equal, Signed	if (N ⊕ V= 0) then PC	←	PC + k + 1	None	1 / 2
BRLT	k	Branch if Less Than, Signed	if (N ⊕ V= 1) then PC	←	PC + k + 1	None	1 / 2
BRHS	k	Branch if Half Carry Flag Set	if (H = 1) then PC	←	PC + k + 1	None	1 / 2
BRHC	k	Branch if Half Carry Flag Cleared	if (H = 0) then PC	←	PC + k + 1	None	1 / 2
BRTS	k	Branch if T Flag Set	if (T = 1) then PC	←	PC + k + 1	None	1 / 2
BRTC	k	Branch if T Flag Cleared	if (T = 0) then PC	←	PC + k + 1	None	1 / 2
BRVS	k	Branch if Overflow Flag is Set	if (V = 1) then PC	←	PC + k + 1	None	1 / 2
BRVC	k	Branch if Overflow Flag is Cleared	if (V = 0) then PC	←	PC + k + 1	None	1 / 2
BRIE	k	Branch if Interrupt Enabled	if (I = 1) then PC	←	PC + k + 1	None	1 / 2
BRID	k	Branch if Interrupt Disabled	if (I = 0) then PC	←	PC + k + 1	None	1 / 2
Data Transfer Instructions							
MOV	Rd, Rr	Copy Register	Rd	←	Rr	None	1
MOVW	Rd, Rr	Copy Register Pair	Rd+1:Rd	←	Rr+1:Rr	None	1

Mnemonics	Operands	Description	Operation			Flags	#Clocks
LDI	Rd, K	Load Immediate	Rd	←	K	None	1
LDS	Rd, k	Load Direct from data space	Rd	←	(k)	None	2(1)(2)
LD	Rd, X	Load Indirect	Rd	←	(X)	None	1(1)(2)
LD	Rd, X+	Load Indirect and Post-Increment	Rd X	← ←	(X) X + 1	None	1(1)(2)
LD	Rd, -X	Load Indirect and Pre-Decrement	X ← X - 1, Rd ← (X)	← ←	X - 1 (X)	None	2(1)(2)
LD	Rd, Y	Load Indirect	Rd ← (Y)	←	(Y)	None	1(1)(2)
LD	Rd, Y+	Load Indirect and Post-Increment	Rd Y	← ←	(Y) Y + 1	None	1(1)(2)
LD	Rd, -Y	Load Indirect and Pre-Decrement	Y Rd	← ←	Y - 1 (Y)	None	2(1)(2)
LDD	Rd, Y+q	Load Indirect with Displacement	Rd	←	(Y + q)	None	2(1)(2)
LD	Rd, Z	Load Indirect	Rd	←	(Z)	None	1(1)(2)
LD	Rd, Z+	Load Indirect and Post-Increment	Rd Z	← ←	(Z), Z+1	None	1(1)(2)
LD	Rd, -Z	Load Indirect and Pre-Decrement	Z Rd	← ←	Z - 1, (Z)	None	2(1)(2)
LDD	Rd, Z+q	Load Indirect with Displacement	Rd	←	(Z + q)	None	2(1)(2)
STS	k, Rr	Store Direct to Data Space	(k)	←	Rd	None	2(1)
ST	X, Rr	Store Indirect	(X)	←	Rr	None	1(1)
ST	X+, Rr	Store Indirect and Post-Increment	(X) X	← ←	Rr, X + 1	None	1(1)
ST	-X, Rr	Store Indirect and Pre-Decrement	X (X)	← ←	X - 1, Rr	None	2(1)
ST	Y, Rr	Store Indirect	(Y)	←	Rr	None	1(1)
ST	Y+, Rr	Store Indirect and Post-Increment	(Y) Y	← ←	Rr, Y + 1	None	1(1)
ST	-Y, Rr	Store Indirect and Pre-Decrement	Y (Y)	← ←	Y - 1, Rr	None	2(1)
STD	Y+q, Rr	Store Indirect with Displacement	(Y + q)	←	Rr	None	2(1)
ST	Z, Rr	Store Indirect	(Z)	←	Rr	None	1(1)
ST	Z+, Rr	Store Indirect and Post-Increment	(Z) Z	← ←	Rr Z + 1	None	1(1)
ST	-Z, Rr	Store Indirect and Pre-Decrement	Z	←	Z - 1	None	2(1)
STD	Z+q,Rr	Store Indirect with Displacement	(Z + q)	←	Rr	None	2(1)
LPM		Load Program Memory	R0	←	(Z)	None	3
LPM	Rd, Z	Load Program Memory	Rd	←	(Z)	None	3
LPM	Rd, Z+	Load Program Memory and Post-Increment	Rd Z	← ←	(Z), Z + 1	None	3
ELPM		Extended Load Program Memory	R0	←	(RAMPZ:Z)	None	3
ELPM	Rd, Z	Extended Load Program Memory	Rd	←	(RAMPZ:Z)	None	3
ELPM	Rd, Z+	Extended Load Program Memory and Post-Increment	Rd Z	← ←	(RAMPZ:Z), Z + 1	None	3
SPM		Store Program Memory	(RAMPZ:Z)	←	R1:R0	None	-

Mnemonics	Operands	Description	Operation			Flags	#Clocks
SPM	Z+	Store Program Memory and Post-Increment by 2	(RAMPZ:Z) Z	← ←	R1:R0, Z + 2	None	-
IN	Rd, A	In From I/O Location	Rd	←	I/O(A)	None	1
OUT	A, Rr	Out To I/O Location	I/O(A)	←	Rr	None	1
PUSH	Rr	Push Register on Stack	STACK	←	Rr	None	1(1)
POP	Rd	Pop Register from Stack	Rd	←	STACK	None	2(1)
Bit and Bit-test Instructions							

LSL	Rd	Logical Shift Left	Rd(n+1)
			Rd(0)
			C
				←
				←
				←
					Rd(n),
					0,
					Rd(7)
						Z,C,N,V,H	1

LSR	Rd	Logical Shift Right	Rd(n)
			Rd(7)
			C
				←
				←
				←
					Rd(n+1),
					0,
					Rd(0)
						Z,C,N,V	1

ROL	Rd	Rotate Left Through Carry	Rd(0)
			Rd(n+1)
			C
				←
				←
				←
					C,
					Rd(n),
					Rd(7)
						Z,C,N,V,H	1

ROR	Rd	Rotate Right Through Carry	Rd(7)
			Rd(n)
			C
				←
				←
				←
					C,
					Rd(n+1),
					Rd(0)
						Z,C,N,V	1

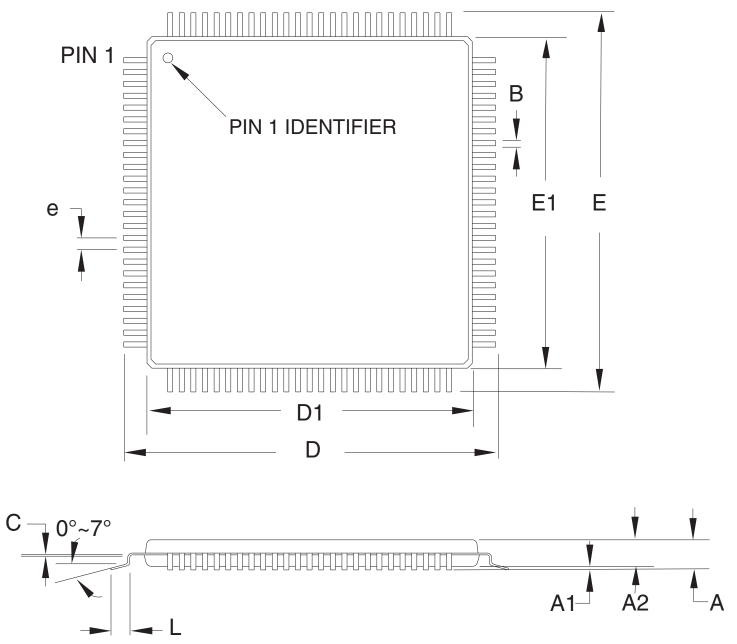
ASR	Rd	Arithmetic Shift Right	Rd(n)	←	Rd(n+1), n=0..6	Z,C,N,V	1
SWAP	Rd	Swap Nibbles	Rd(3..0)	↔	Rd(7..4)	None	1
BSET	s	Flag Set	SREG(s)	←	1	SREG(s)	1
BCLR	s	Flag Clear	SREG(s)	←	0	SREG(s)	1
SBI	A, b	Set Bit in I/O Register	I/O(A, b)	←	1	None	1
CBI	A, b	Clear Bit in I/O Register	I/O(A, b)	←	0	None	1
BST	Rr, b	Bit Store from Register to T	T	←	Rr(b)	T	1
BLD	Rd, b	Bit load from T to Register	Rd(b)	←	T	None	1
SEC		Set Carry	C	←	1	C	1
CLC		Clear Carry	C	←	0	C	1
SEN		Set Negative Flag	N	←	1	N	1
CLN		Clear Negative Flag	N	←	0	N	1
SEZ		Set Zero Flag	Z	←	1	Z	1
CLZ		Clear Zero Flag	Z	←	0	Z	1
SEI		Global Interrupt Enable	I	←	1	I	1
CLI		Global Interrupt Disable	I	←	0	I	1
SES		Set Signed Test Flag	S	←	1	S	1
CLS		Clear Signed Test Flag	S	←	0	S	1
SEV		Set Two's Complement Overflow	V	←	1	V	1
CLV		Clear Two's Complement Overflow	V	←	0	V	1
SET		Set T in SREG	T	←	1	T	1
CLT		Clear T in SREG	T	←	0	T	1
SEH		Set Half Carry Flag in SREG	H	←	1	H	1
CLH		Clear Half Carry Flag in SREG	H	←	0	H	1

Mnemonics	Operands	Description	Operation	Flags	#Clocks
MCU Control Instructions					
BREAK		Break	(See specific descr. for BREAK)	None	1
NOP		No Operation		None	1
SLEEP		Sleep	(see specific descr. for Sleep)	None	1
WDR		Watchdog Reset	(see specific descr. for WDR)	None	1

Notes:
1. Cycle times for Data memory accesses assume internal memory accesses, and are not valid for accesses via the external RAM interface.
2. One extra cycle must be added when accessing Internal SRAM.

33. Packaging information

33.1 100A



COMMON DIMENSIONS
(Unit of Measure = mm)

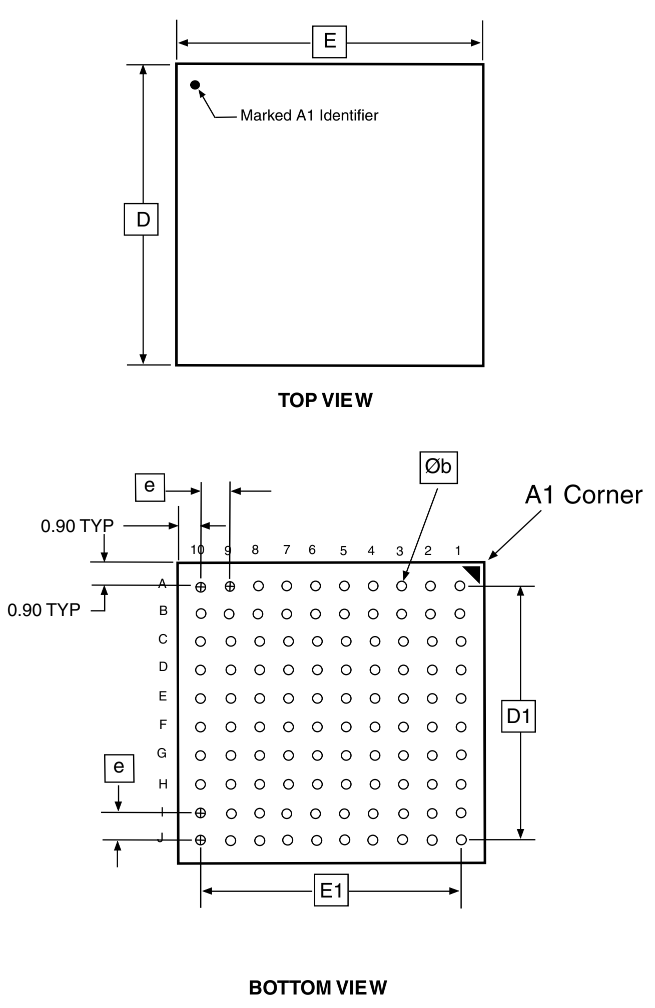
SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.20	
A1	0.05	–	0.15	
A2	0.95	1.00	1.05	
D	15.75	16.00	16.25	
D1	13.90	14.00	14.10	Note 2
E	15.75	16.00	16.25	
E1	13.90	14.00	14.10	Note 2
B	0.17	–	0.27	
C	0.09	–	0.20	
L	0.45	–	0.75	
e	0.50 TYP			

Notes:
1. This package conforms to JEDEC reference MS-026, Variation AED.
2. Dimensions D1 and E1 do not include mold protrusion. Allowable protrusion is 0.25 mm per side. Dimensions D1 and E1 are maximum plastic body size dimensions including mold mismatch.
3. Lead coplanarity is 0.08 mm maximum.

2010-10-20

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**100A,** 100-lead, 14 x 14 mm Body Size, 1.0 mm Body Thickness, 0.5 mm Lead Pitch, Thin Profile Plastic Quad Flat Package (TQFP)	100A	D

33.2 100C1



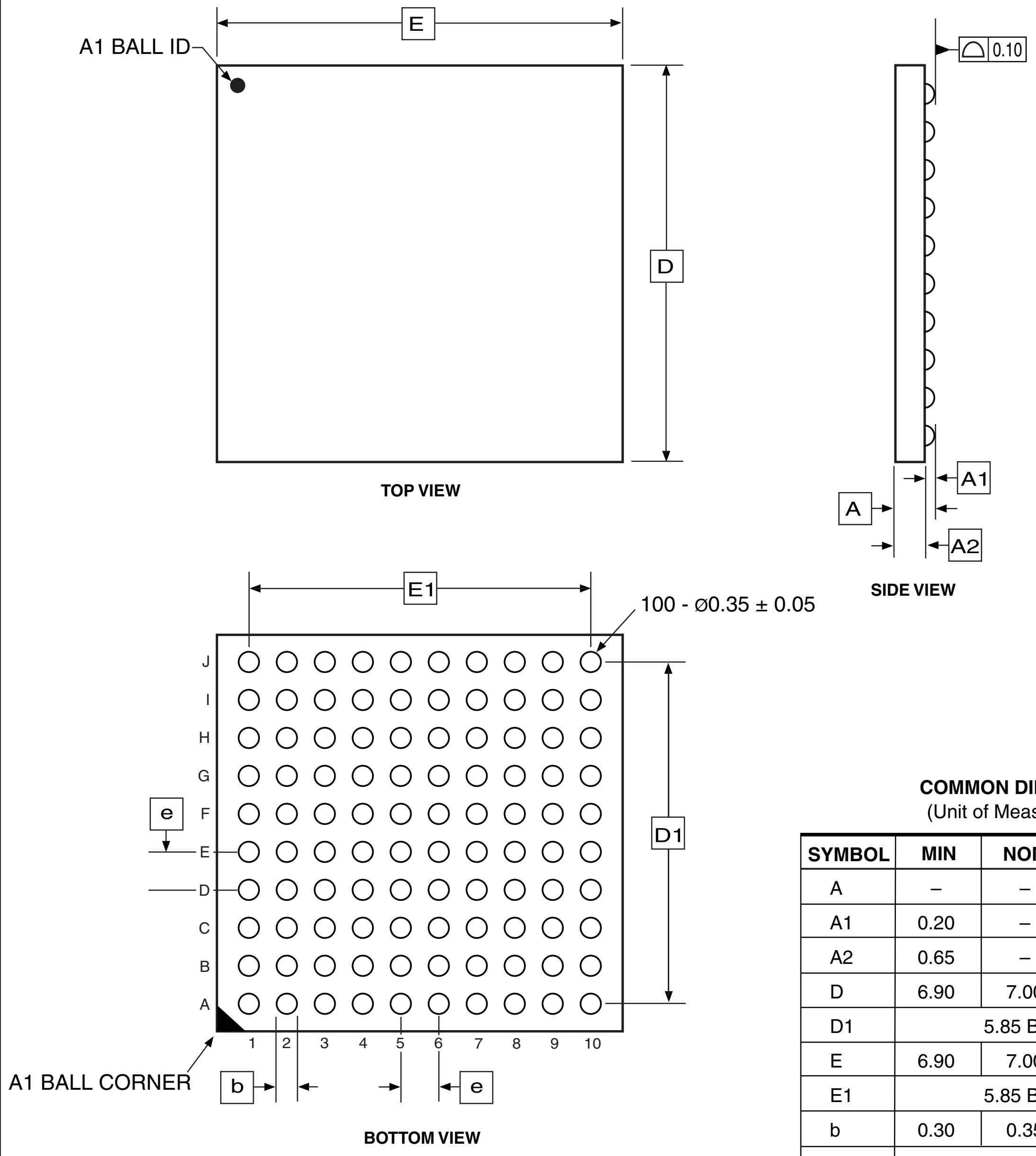
COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	1.10	–	1.20	
A1	0.30	0.35	0.40	
D	8.90	9.00	9.10	
E	8.90	9.00	9.10	
D1	7.10	7.20	7.30	
E1	7.10	7.20	7.30	
Øb	0.35	0.40	0.45	
e	0.80 TYP			

5/25/06

	TITLE	DRAWING NO.	REV.
2325 Orchard Parkway San Jose, CA 95131	**100C1**, 100-ball, 9 x 9 x 1.2 mm Body, Ball Pitch 0.80 mm Chip Array BGA Package (CBGA)	100C1	A

33.3 100C2



COMMON DIMENSIONS
(Unit of Measure = mm)

SYMBOL	MIN	NOM	MAX	NOTE
A	–	–	1.00	
A1	0.20	–	–	
A2	0.65	–	–	
D	6.90	7.00	7.10	
D1		5.85 BSC		
E	6.90	7.00	7.10	
E1		5.85 BSC		
b	0.30	0.35	0.40	
e		0.65 BSC		

12/23/08

	TITLE	GPC	DRAWING NO.	REV.
Package Drawing Contact: packagedrawings@atmel.com	**100C2,** 100-ball (10 x 10 Array), 0.65 mm Pitch, 7.0 x 7.0 x 1.0 mm, Very Thin, Fine-Pitch Ball Grid Array Package (VFBGA)	CIF	100C2	A

34. Electrical Characteristics

34.1 Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin with respect to Ground	-0.5V to V_{CC}+0.5V
Maximum Operating Voltage	3.6V
DC Current per I/O Pin	20.0 mA
DC Current V_{CC} and GND Pins	200.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

34.2 DC Characteristics

Table 34-1. Current consumption.

Symbol	Parameter	Condition		Min	Typ	Max	Units
I_{CC}	Active mode(1)	1 MHz, Ext. Clk	V_{CC} = 1.8V		365		µA
			V_{CC} = 3.0V		790		
		2 MHz, Ext. Clk	V_{CC} = 1.8V		690	800	
			V_{CC} = 3.0V		1400	1600	
		32 MHz, Ext. Clk	V_{CC} = 3.0V		18.35	20	mA
	Idle mode(1)	1 MHz, Ext. Clk	V_{CC} = 1.8V		135		µA
			V_{CC} = 3.0V		255		
		2 MHz, Ext. Clk	V_{CC} = 1.8V		270	380	
			V_{CC} = 3.0V		510	650	
		32 MHz, Ext. Clk	V_{CC} = 3.0V		8.15	9.2	mA
	Power-down mode	All Functions Disabled	V_{CC} = 3.0V		0.1		µA
		All Functions Disabled, T = 85°C	V_{CC} = 3.0V		2	5	
		ULP, WDT, Sampled BOD	V_{CC} = 1.8V		0.5		
			V_{CC} = 3.0V		0.6		
		ULP, WDT, Sampled BOD, T=85°C	V_{CC} = 3.0V		3	10	
	Power-save mode	RTC 1 kHz from Low Power 32 kHz	V_{CC} = 1.8V		0.52		µA
			V_{CC} = 3.0V		0.55		
		RTC from Low Power 32 kHz	V_{CC} = 3.0V		1.16		

Symbol	Parameter	Condition	Min	Typ	Max	Units
Module current consumption[(2)]						
I_{CC}	RC32M			395		µA
	RC32M w/DFLL	Internal 32.768 kHz oscillator as DFLL source		TBD		
	RC2M			120		
	RC2M w/DFLL	Internal 32.768 kHz oscillator as DFLL source		155		
	RC32K			30		
	PLL	Multiplication factor = 10x		195		
	Watchdog normal mode			TBD		
	BOD Continuous mode			120		
	BOD Sampled mode			1		
	Internal 1.00 V ref			85		
	Temperature reference			80		
	RTC with int. 32 kHz RC as source	No prescaling		30		
	RTC with ULP as source	No prescaling		1		
	ADC	250 kS/s - Int. 1V Ref		3.6		mA
	DAC Normal Mode	1000 kS/s, Single channel, Int. 1V Ref		1.8		
	DAC Low-Power Mode	1000 KS/s, Single channel, Int. 1V Ref		1		
	AC High-speed			220		µA
	AC Low-power			110		
	USART	Rx and Tx enabled, 9600 BAUD		7.5		
	DMA			180		
	Timer/Counter	Prescaler DIV1		18		
	AES			195		
	Flash/EEPROM Programming	Vcc = 2V		20		mA
		Vcc = 3V		30		

Note: 1. All Power Reduction Registers set. Typical numbers measured at T = 25°C if nothing else is specified.
2. with no prescaling

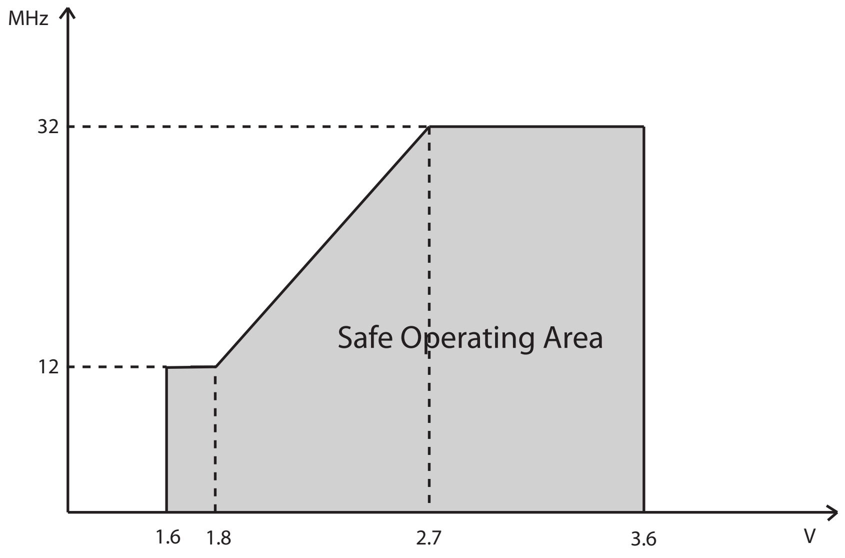
34.3 Speed

Table 34-2. Operating voltage and frequency.

Symbol	Parameter	Condition	Min	Typ	Max	Units
Clk_{CPU}	CPU clock frequency	V_{CC} = 1.6V	0		12	MHz
		V_{CC} = 1.8V	0		12	
		V_{CC} = 2.7V	0		32	
		V_{CC} = 3.6V	0		32	

The maximum CPU clock frequency of the XMEGA A1 devices is depending on V_{CC}. As shown in Figure 34-1 on page 75 the Frequency vs. V_{CC} curve is linear between 1.8V < V_{CC} < 2.7V.

Figure 34-1. Maximum Frequency vs. Vcc



34.4 Flash and EEPROM Memory Characteristics

Table 34-3. Endurance and data retention.

Symbol	Parameter	Condition		Min	Typ	Max	Units
	Flash	Write/Erase cycles	25°C	10K			Cycle
			85°C	10K			
		Data retention	25°C	100			Year
			55°C	25			
	EEPROM	Write/Erase cycles	25°C	80K			Cycle
			85°C	30K			
		Data retention	25°C	100			Year
			55°C	25			

Table 34-4. Programming time.

Symbol	Parameter	Condition	Min	Typ[1]	Max	Units
	Chip Erase	Flash, EEPROM[2] and SRAM Erase		40		ms
	Flash	Page Erase		4		
		Page Write		6		
		Page WriteAutomatic Page Erase and Write		12		
	EEPROM	Page Erase		4		
		Page Write		6		
		Page Write Automatic Page Erase and Write		12		

Notes: 1. Programming is timed from the internal 2 MHz oscillator.
2. EEPROM is not erased if the EESAVE fuse is programmed.

34.5 ADC Characteristics

Table 34-5. ADC characteristics

Symbol	Parameter	Condition		Min	Typ	Max	Units
RES	Resolution	Programmable: 8/12		8	12	12	Bits
INL	Integral Non-Linearity	500 kS/s		-5	<±1	5	LSB
DNL	Differential Non-Linearity	500 kS/s			< ±0.75		LSB
	Gain Error				±10		mV
	Offset Error				±2		mV
ADC_{clk}	ADC Clock frequency	Max is 1/4 of Peripheral Clock	V_{CC}≥2.0V			2000	kHz
			V_{CC}<2.0V			500	

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Conversion rate	V_{CC}≥2.0V			2000	ksps
		V_{CC}<2.0V			500	
	Conversion time (propagation delay)	(RES+2)/2+GAIN RES = 8 or 12, GAIN = 0 or 1	5	7	8	ADC_{clk} cycles
	Sampling Time	1/2 ADC_{clk} cycle	0.25			µS
	Conversion range		0		VREF	V
AVCC	Analog Supply Voltage		V_{cc}-0.3		V_{cc}+0.3	V
VREF	Reference voltage		1.0		V_{cc}-0.6	V
	Input bandwidth	V_{CC}≥2.0V			2000	kHz
		V_{CC}<2.0V			500	
INT1V	Internal 1.00V reference			1.00		V
INTVCC	Internal V_{CC}/1.6			V_{CC}/1.6		V
SCALEDVCC	Scaled internal V_{CC}/10 input			V_{CC}/10		V
R_{AREF}	Reference input resistance			>10		MΩ
	Start-up time			12	24	ADC_{clk} cycles
	Internal input sampling speed	Temp. sensor, V_{CC}/10, Bandgap			100	ksps

Table 34-6. ADC gain stage characteristics.

Symbol	Parameter	Condition		Min	Typ	Max	Units
	Gain error	1 to 64 gain			< ±1		%
	Offset error				< ±1		mV
Vrms	Noise level at input	64x gain	VREF = Int. 1V		0.12		mV
			VREF = Ext. 2V		0.06		
	Clock rate	Same as ADC				1000	kHz

34.6 DAC Characteristics

Table 34-7. DAC characteristics.

Symbol	Parameter	Condition		Min	Typ	Max	Units
INL	Integral Non-Linearity	V_{CC} = 1.6-3.6V	VREF = Ext. ref		5		LSB
DNL	Differential Non-Linearity	V_{CC} = 1.6-3.6V	VREF = Ext. ref		0.6	<±1	LSB
			VREF= AV_{CC}		0.6		
F_{clk}	Conversion rate					1000	ksps
AREF	External reference voltage			1.1		AV_{CC}-0.6	V
	Reference input impedance				>10		MΩ
	Max output voltage	R_{load}=100kΩ			AV_{CC}*0.98		V
	Min output voltage	R_{load}=100kΩ			0.01		V

34.7 Analog Comparator Characteristics

Table 34-8. Analog Comparator characteristics.

Symbol	Parameter	Condition		Min	Typ	Max	Units
V_{off}	Input Offset Voltage	V_{CC} = 1.6 - 3.6V			<±5		mV
I_{lk}	Input Leakage Current	V_{CC} = 1.6 - 3.6V			< 1000		pA
V_{hys1}	Hysteresis, No	V_{CC} = 1.6 - 3.6V			0		mV
V_{hys2}	Hysteresis, Small	V_{CC} = 1.6 - 3.6V	mode = HS		25		mV
V_{hys3}	Hysteresis, Large	V_{CC} = 1.6 - 3.6V	mode = HS		50		mV
t_{delay}	Propagation delay	V_{CC} = 3.0V, T= 85°C	mode = HS			100	ns
		V_{CC} = 1.6 - 3.6V	mode = HS		70		
		V_{CC} = 1.6 - 3.6V	mode = LP		140		

34.8 Bandgap Characteristics

Table 34-9. Bandgap voltage characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Bandgap startup time	As reference for ADC or DAC	1 Clk_PER + 2.5µs			µs
	Bandgap voltage			1.1		V

Symbol	Parameter	Condition	Min	Typ	Max	Units
	ADC/DAC ref	T= 85°C, After calibration	0.99		1.01	V
				1		
	Variation over voltage and temperature	V_{CC} = 1.6 - 3.6V, T = -40°C to 85°C		±5		%

34.9 Brownout Detection Characteristics

Table 34-10. Brownout Detection characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
	BOD level 0 falling Vcc			1.6		V
	BOD level 1 falling Vcc			1.9		
	BOD level 2 falling Vcc			2.1		
	BOD level 3 falling Vcc			2.4		
	BOD level 4 falling Vcc			2.6		
	BOD level 5 falling Vcc			2.9		
	BOD level 6 falling Vcc			3.2		
	BOD level 7 falling Vcc			3.4		
	Hysteresis	BOD level 0-5		2		%

Note: 1. BOD is calibrated to BOD level 0 at 85°C, and BOD level 0 is the default level.

34.10 PAD Characteristics

Table 34-11. PAD characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{IH}	Input High Voltage	V_{CC} = 2.4 - 3.6V	0.7*V_{CC}		V_{CC}+0.5	V
		V_{CC} = 1.6 - 2.4V	0.8*V_{CC}		V_{CC}+0.5	
V_{IL}	Input Low Voltage	V_{CC} = 2.4 - 3.6V	-0.5		0.3*V_{CC}	V
		V_{CC} = 1.6 - 2.4V	-0.5		0.2*V_{CC}	
V_{OL}	Output Low Voltage GPIO	I_{OL} = 15 mA, V_{CC} = 3.3V		0.45	0.76	V
		I_{OL} = 10 mA, V_{CC} = 3.0V		0.3	0.64	
		I_{OL}= 5 mA, V_{CC} = 1.8V		0.2	0.46	

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{OH}	Output High Voltage GPIO	I_{OH} = -8 mA, V_{CC} = 3.3V	2.6	3		V
		I_{OH} = -6 mA, V_{CC} = 3.0V	2.1	2.2		
		I_{OH} = -2 mA, V_{CC} = 1.8V	1.4	1.6		
I_{IL}	Input Leakage Current I/O pin			<0.001	1	µA
I_{IH}	Input Leakage Current I/O pin			<0.001	1	µA
R_P	I/O pin Pull/Buss keeper Resistor			20		kΩ
R_{RST}	Reset pin Pull-up Resistor			20		kΩ
	Input hysteresis			0.5		V

34.11 POR Characteristics

Table 34-12. Power-on Reset characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT-}	POR threshold voltage falling Vcc			1		V
V_{POT+}	POR threshold voltage rising Vcc			1.4		V

34.12 Reset Characteristics

Table 34-13. Reset characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Minimum reset pulse width			90		ns
	Reset threshold voltage	V_{CC} = 2.7 - 3.6V		0.45*V_{CC}		V
		V_{CC} = 1.6 - 2.7V		0.42*V_{CC}		

34.13 Oscillator Characteristics

Table 34-14. Internal 32.768kHz oscillator characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Accuracy	T = 85°C, V_{CC} = 3V, After production calibration	-0.5		0.5	%

Table 34-15. Internal 2MHz oscillator characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Accuracy	T = 85°C, V_{CC} = 3V, After production calibration	-1.5		1.5	%
	DFLL Calibration step size	T = 25°C, V_{CC} = 3V		0.175		%

Table 34-16. Internal 32MHz oscillator characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Accuracy	T = 85°C, V_{CC} = 3V, After production calibration	-1.5		1.5	%
	DFLL Calibration stepsize	T = 25°C, V_{CC} = 3V		0.2		%

Table 34-17. Internal 32kHz, ULP oscillator characteristics.

Symbol	Parameter	Condition	Min	Typ	Max	Units
	Output frequency 32 kHz ULP OSC	T = 85°C, V_{CC} = 3.0V		26		kHz

Table 34-18. Maximum load capacitance (CL) and ESR recommendation for 32.768kHz crystal.

Crystal CL [pF]	Max ESR [kΩ]
6.5	60
9	35

Table 34-19. Device wake-up time from sleep.

Symbol	Parameter	Condition[1]	Min	Typ[2]	Max	Units
	Idle Sleep, Standby and Extended Standby sleep mode	Int. 32.768 kHz RC		130		µS
		Int. 2 MHz RC		2		
		Ext. 2 MHz Clock		2		
		Int. 32 MHz RC		0.17		
	Power-save and Power-down Sleep mode	Int. 32.768 kHz RC		320		
		Int. 2 MHz RC		10.3		
		Ext. 2 MHz Clock		4.5		
		Int. 32 MHz RC		5.8		

Notes: 1. Non-prescaled System Clock source.
2. Time from pin change on external interrupt pin to first available clock cycle. Additional interrupt response time is minimum 5 system clock source cycles.

35. Typical Characteristics

35.1 Active Supply Current

Figure 35-1. Active Supply Current vs. Frequency
f_{SYS} = 1 - 32 MHz, T = 25°C

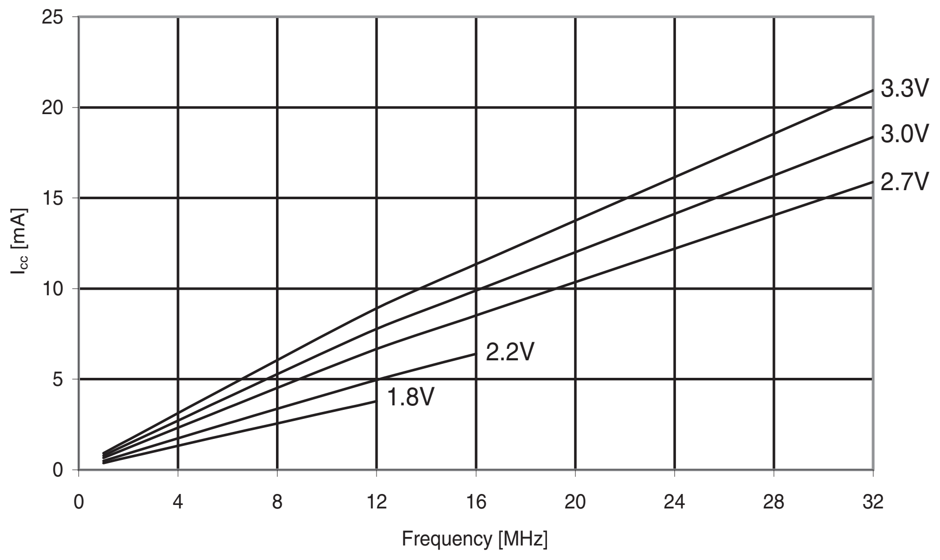
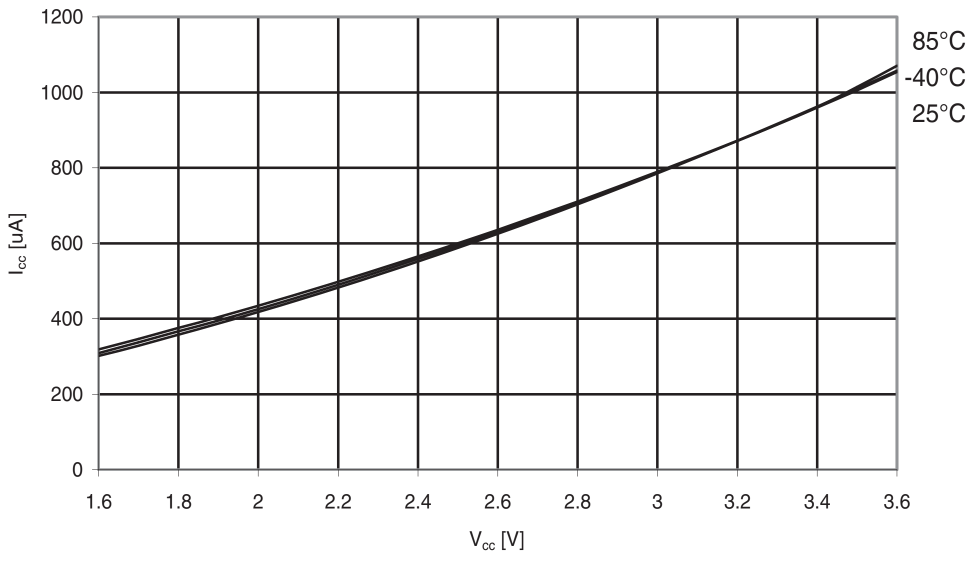


Figure 35-2. Active Supply Current vs. V_{CC}
f_{SYS} = 1.0 MHz



35.2 Idle Supply Current

Figure 35-3. Idle Supply Current vs. Frequency

f_{SYS} = 1 - 32 MHz, T = 25°C

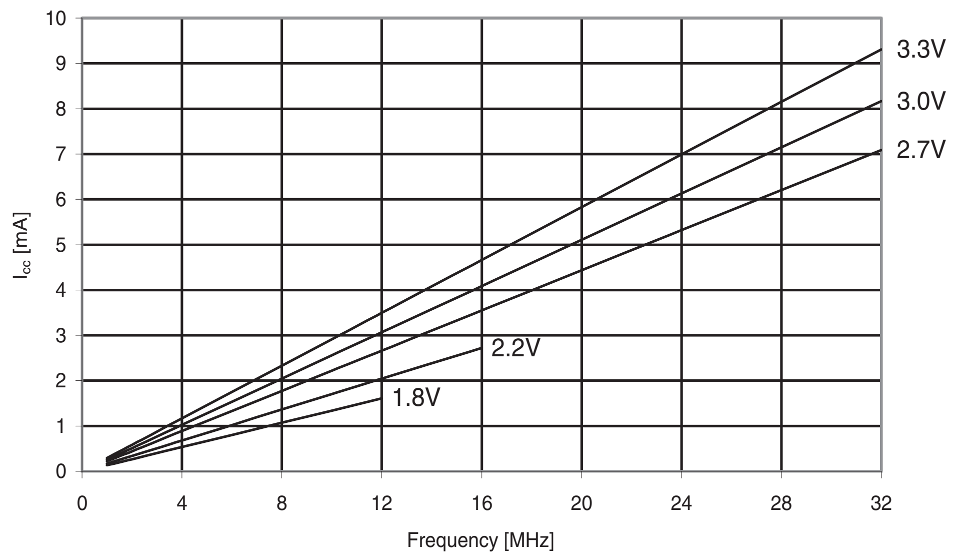
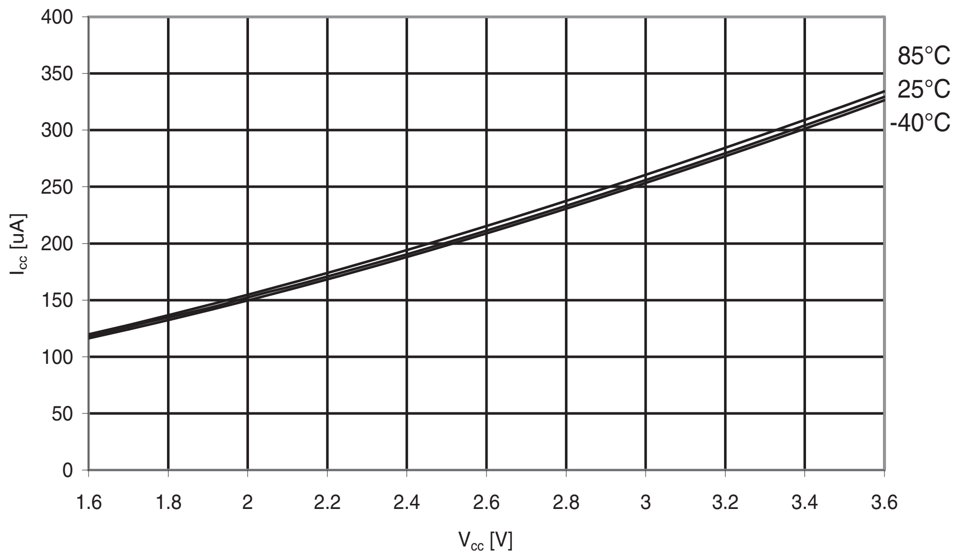


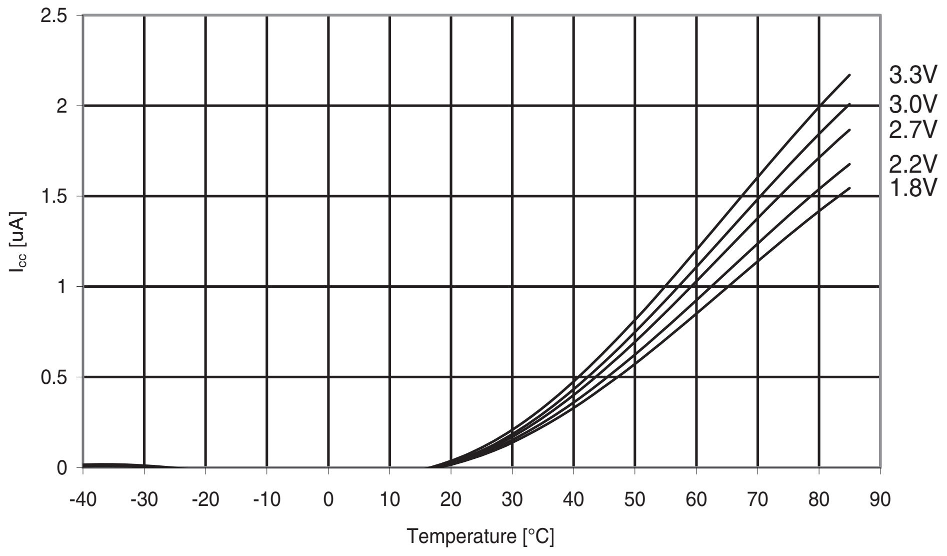
Figure 35-4. Active Supply Current vs. V_{CC}

f_{SYS} = 1.0 MHz



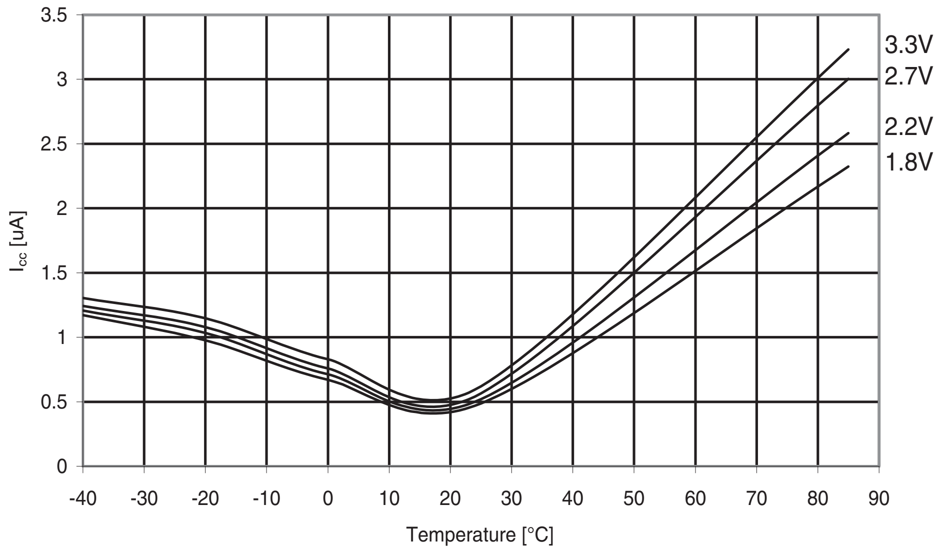
35.3 Power-down Supply Current

Figure 35-5. Power-down Supply Current vs. Temperature



35.4 Power-save Supply Current

Figure 35-6. Power-save Supply Current vs. Temperature
Sampled BOD, WDT, RTC from ULP enabled



35.5 Pin Pull-up

Figure 35-7. I/O Reset Pull-up Resistor Current vs. Reset Pin Voltage

V_{CC} = 1.8V

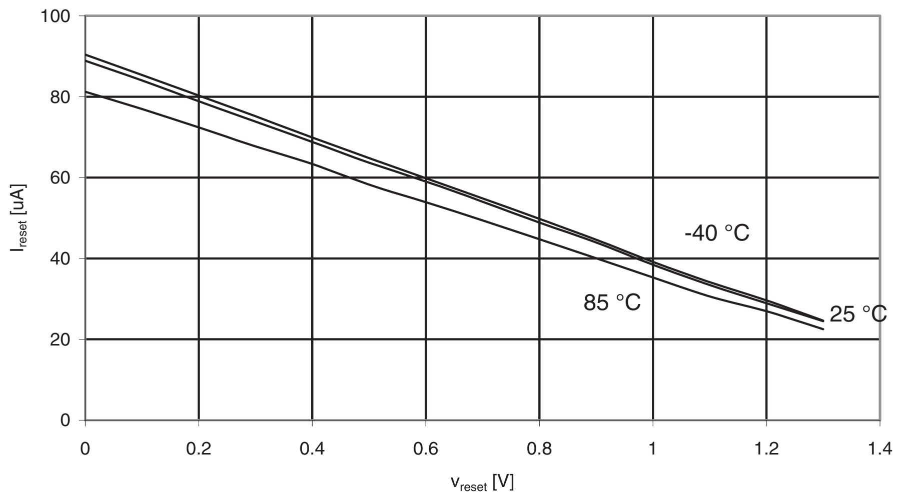


Figure 35-8. I/O Reset Pull-up Resistor Current vs. Reset Pin Voltage

V_{CC} = 3.0V

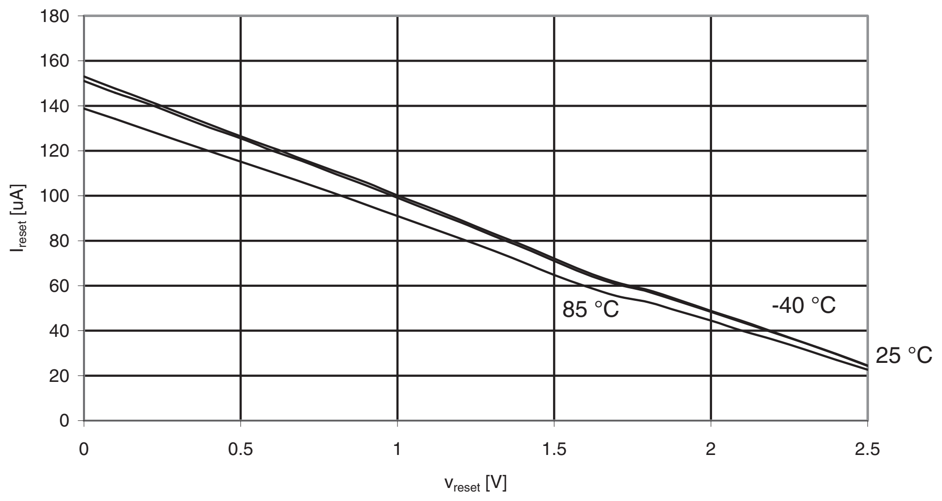
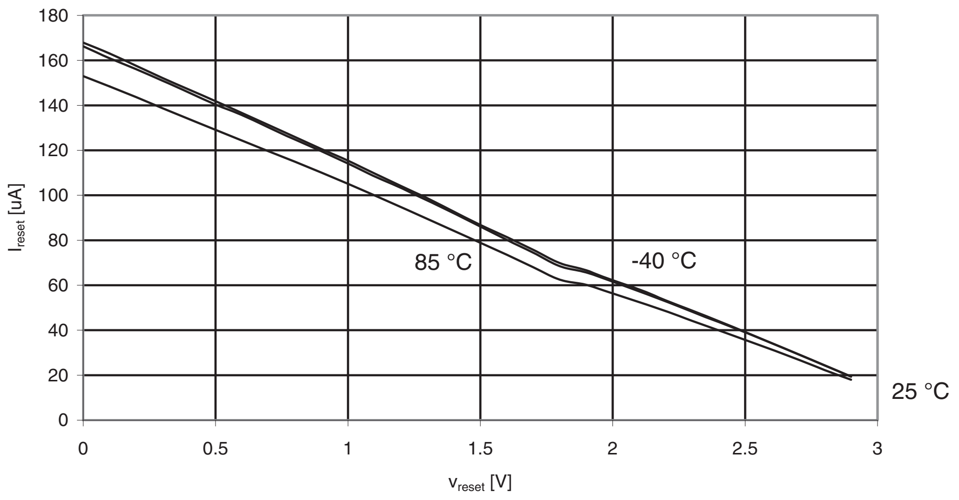


Figure 35-9. I/O Reset Pull-up Resistor Current vs. Reset Pin Voltage

V_{CC} = 3.3V



35.6 Pin Thresholds and Hysteresis

Figure 35-10.I/O Pin Input Threshold Voltage vs. V_{CC}

V_{IH} - I/O Pin Read as "1"

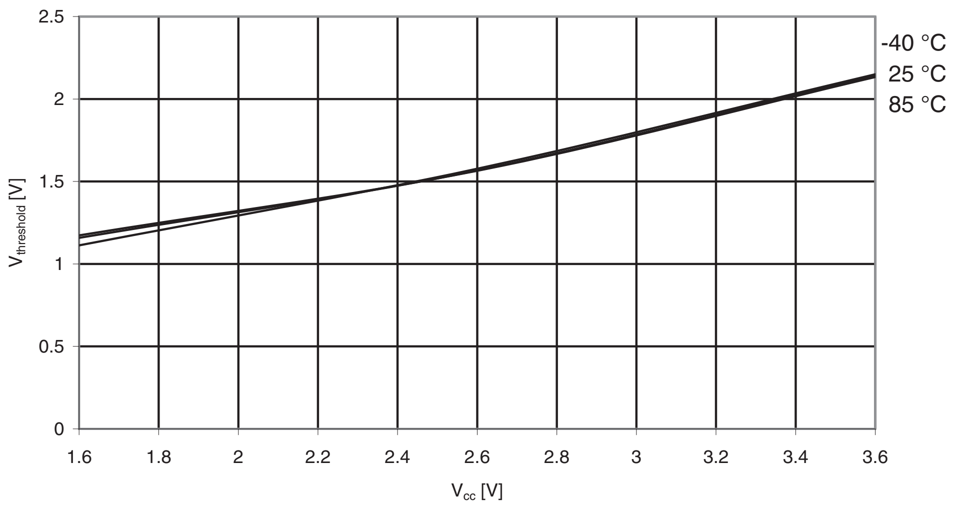


Figure 35-11.I/O Pin Input Threshold Voltage vs. V_{CC}

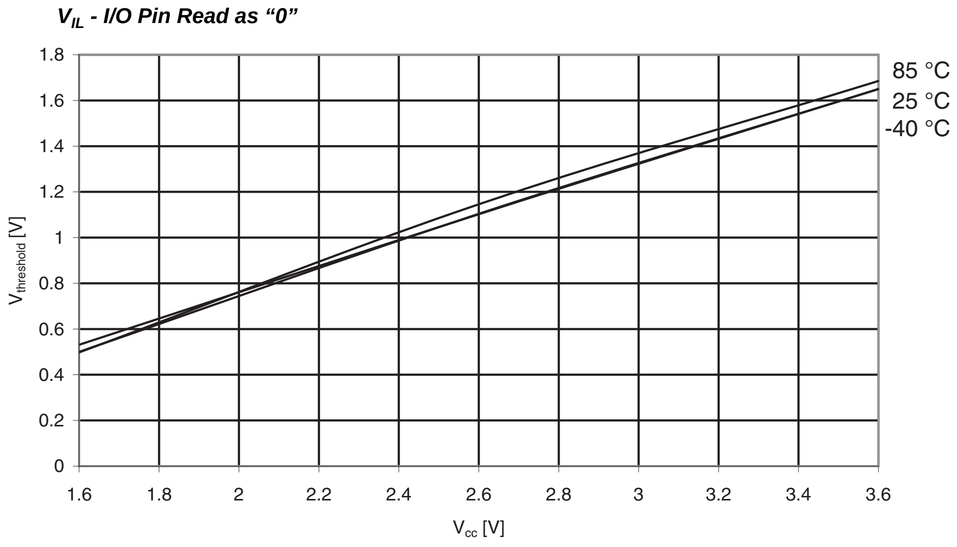


Figure 35-12.I/O Pin Input Hysteresis vs. $V_{CC.}$

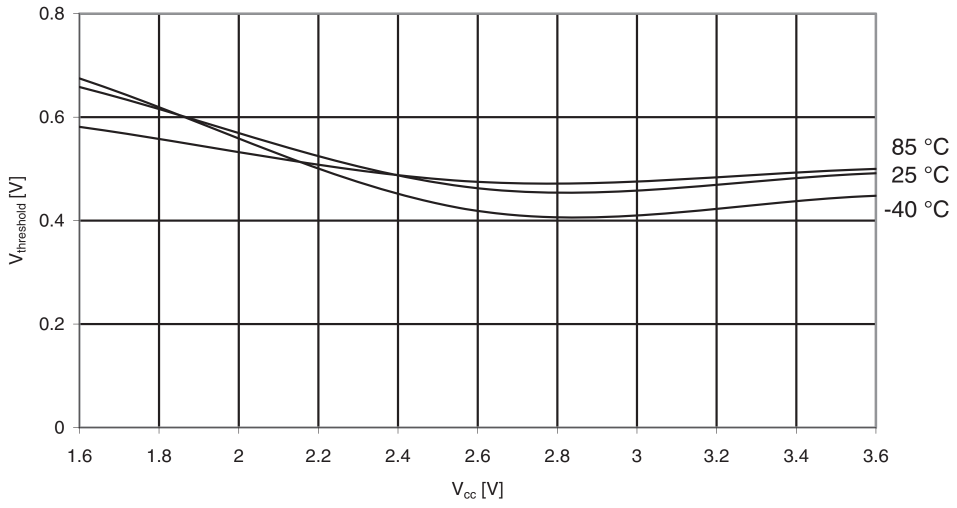


Figure 35-13.Reset Input Threshold Voltage vs. V_{CC}

V_{IH} - I/O Pin Read as “1”

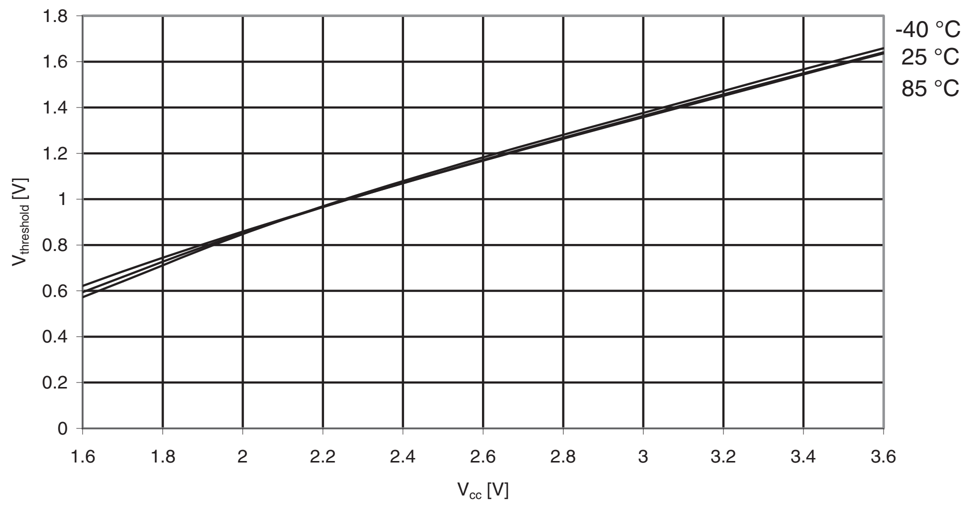
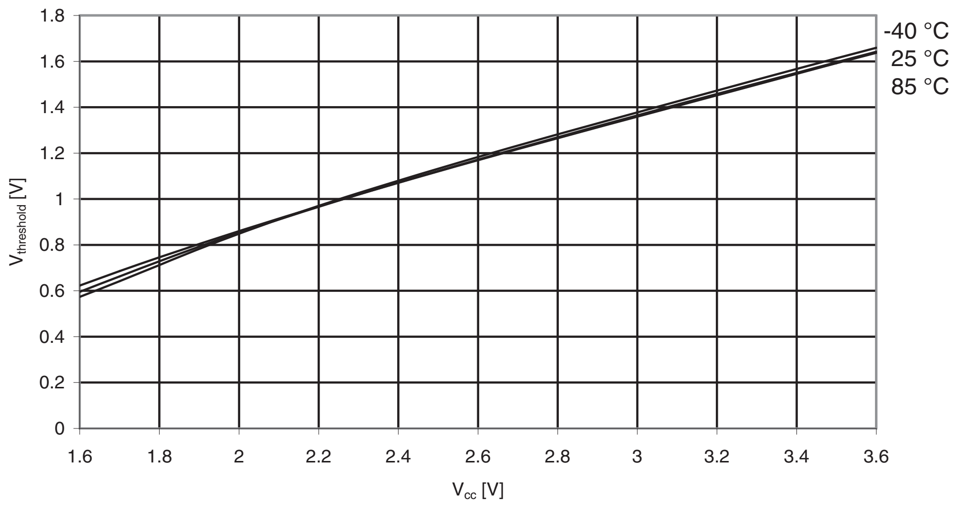


Figure 35-14.Reset Input Threshold Voltage vs. V_{CC}

V_{IL} - I/O Pin Read as “0”



35.7 Bod Thresholds

Figure 35-15.BOD Thresholds vs. Temperature

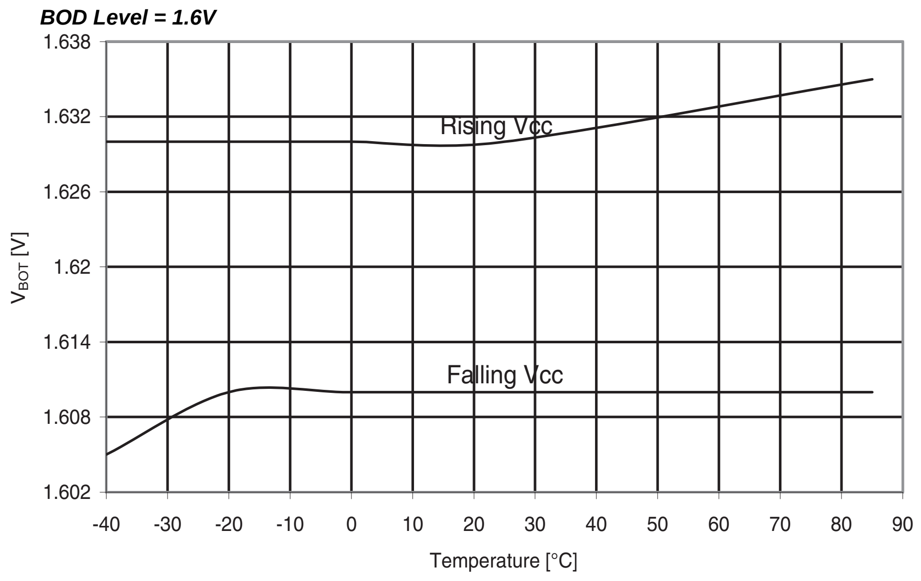
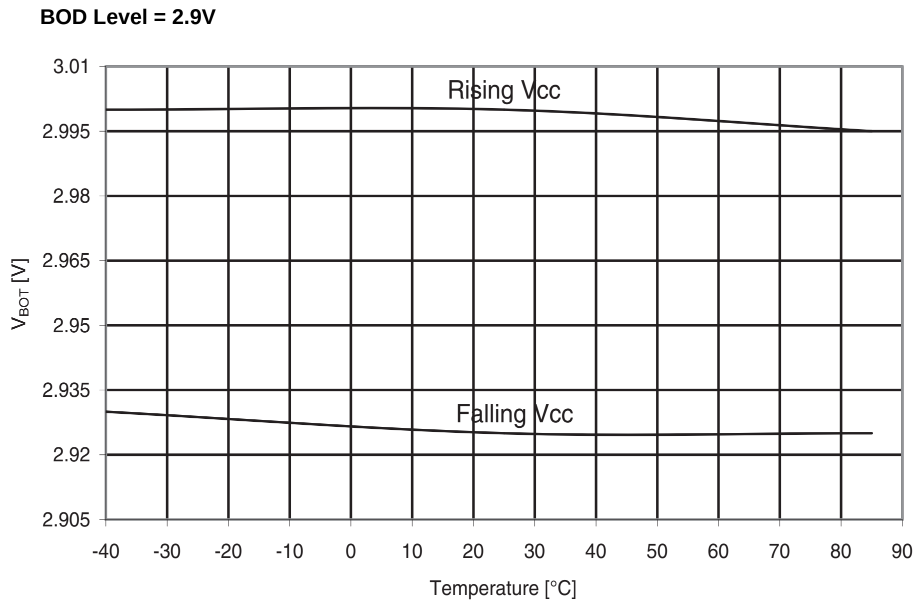
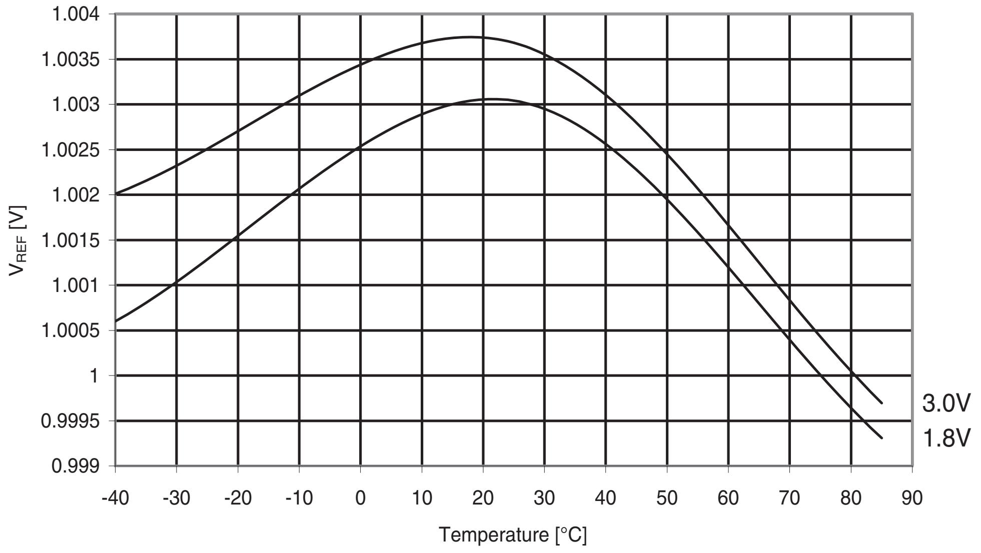


Figure 35-16.BOD Thresholds vs. Temperature



35.8 Bandgap

Figure 35-17.Internal 1.00V Reference vs. Temperature.



35.9 Analog Comparator

Figure 35-18.Analog Comparator Hysteresis vs. V_{CC}
High-speed, Small hysteresis

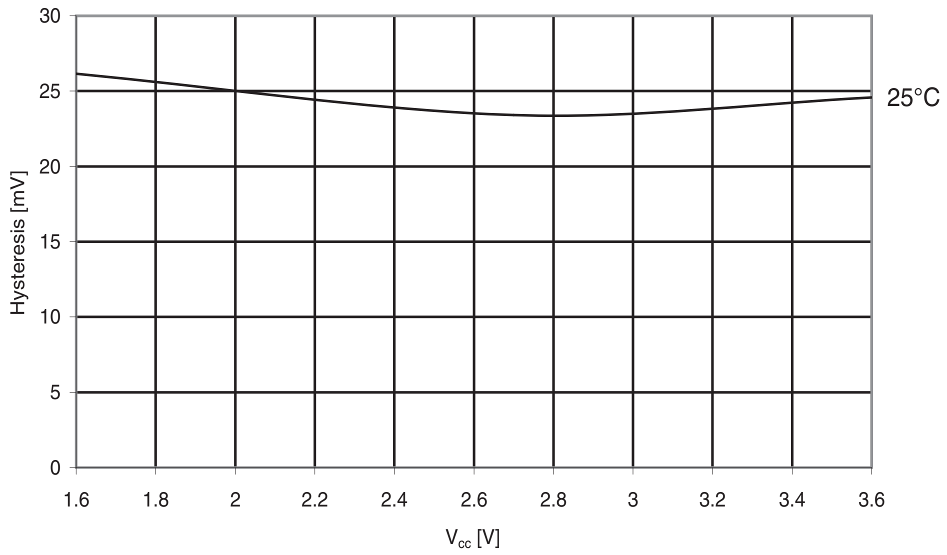


Figure 35-19.Analog Comparator Hysteresis vs. V_{CC}, High-speed

Large hysteresis

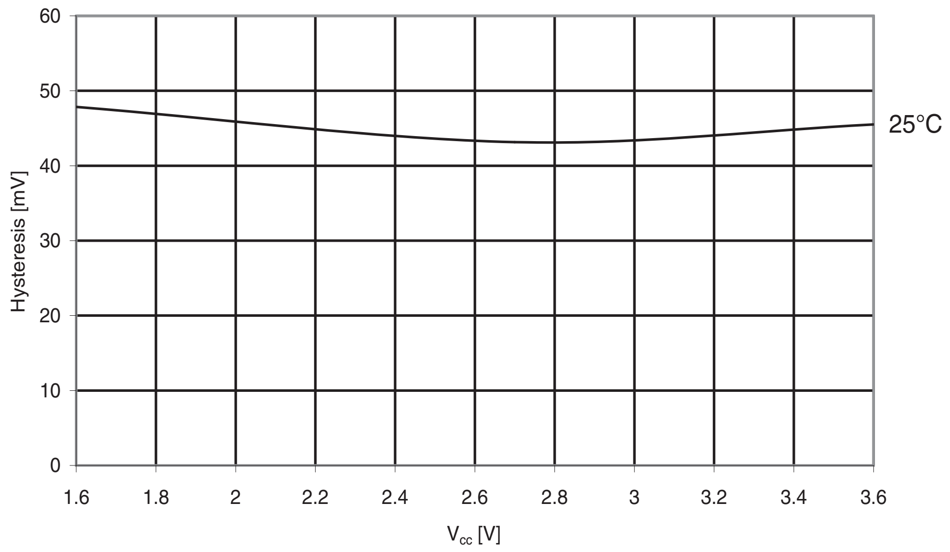
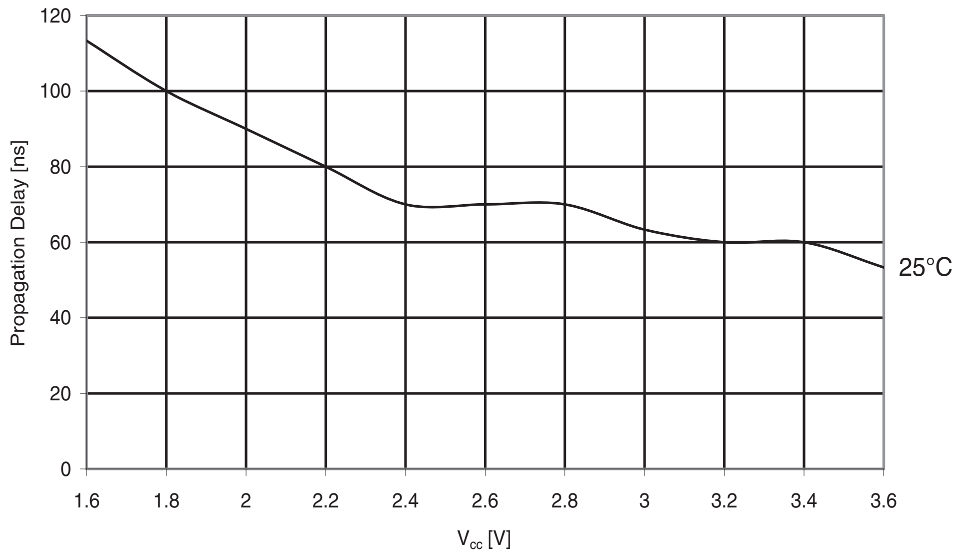


Figure 35-20.Analog Comparator Propagation Delay vs. V_{CC}

High-speed



35.10 Oscillators and Wake-up Time

Figure 35-21.Internal 32.768 kHz Oscillator Frequency vs. Temperature

1.024 kHz output

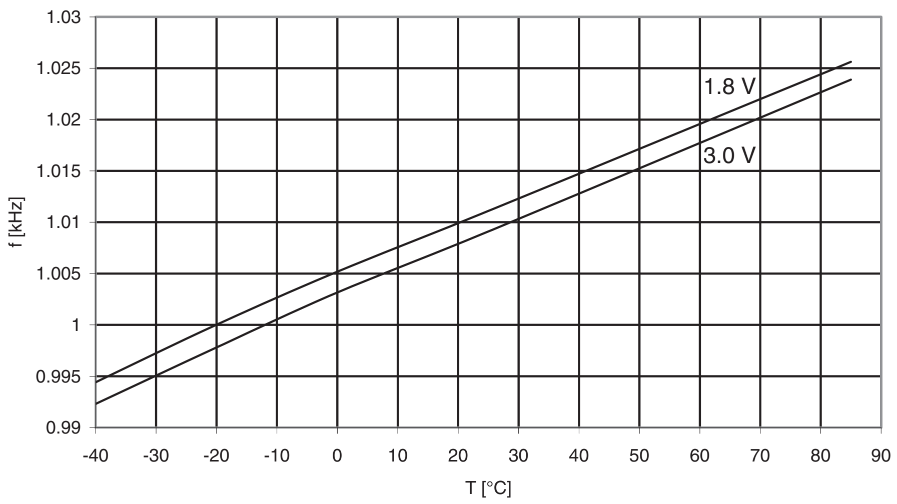


Figure 35-22.Ultra Low-Power (ULP) Oscillator Frequency vs. Temperature

1 kHz output

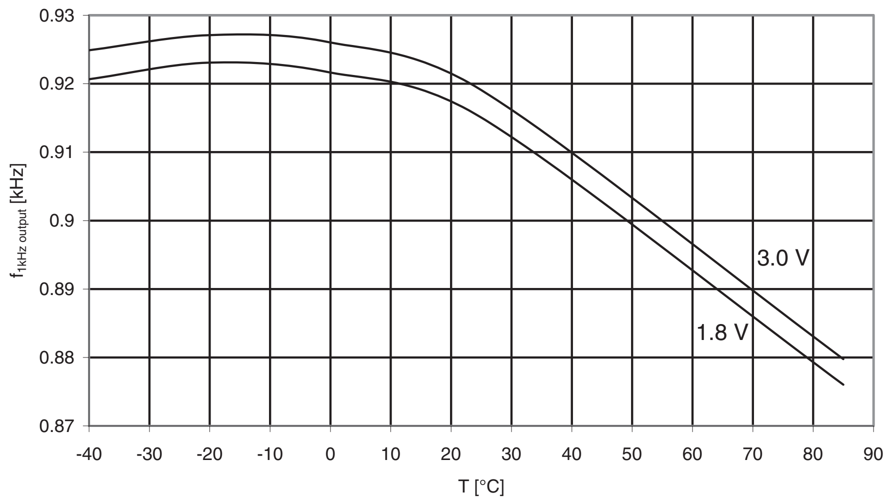


Figure 35-23.Internal 2 MHz Oscillator CalA Calibration Step Size

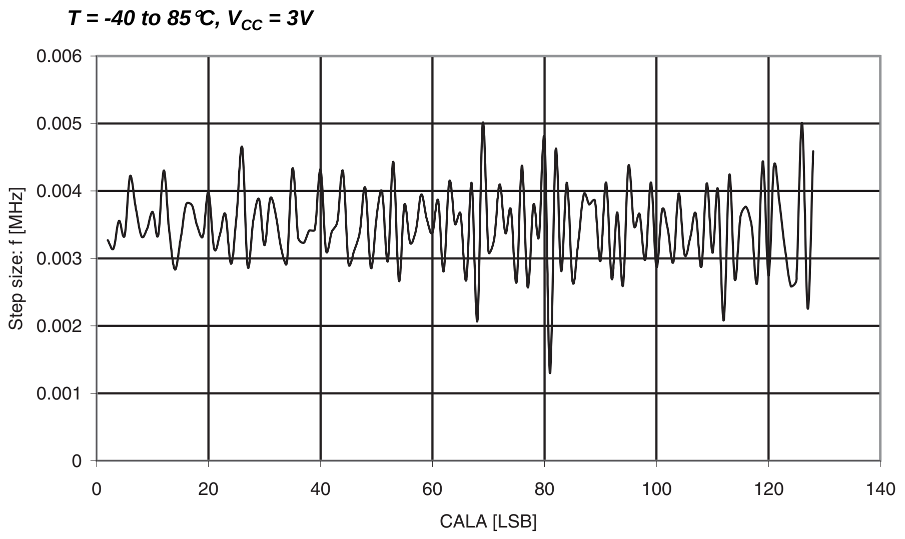


Figure 35-24.Internal 2 MHz Oscillator CalB Calibration Step Size

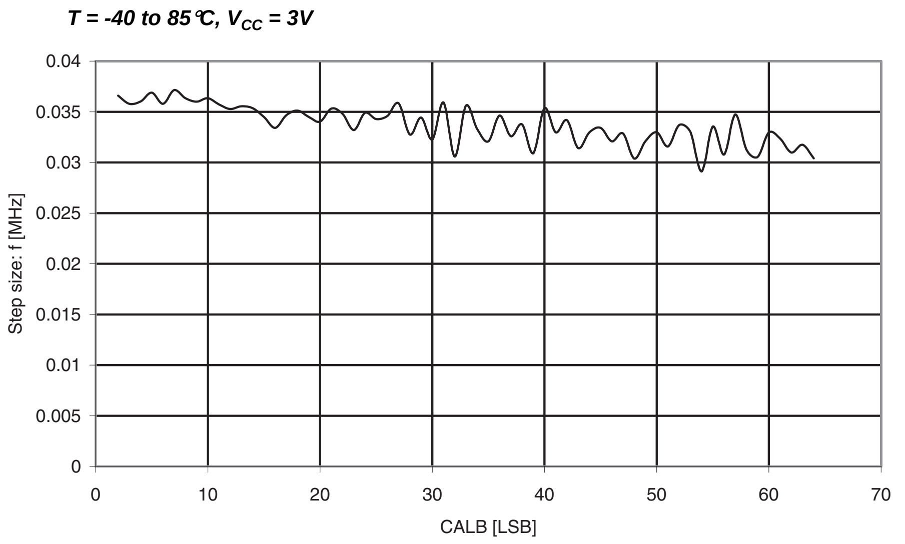


Figure 35-25.Internal 32 MHz Oscillator CalA Calibration Step Size

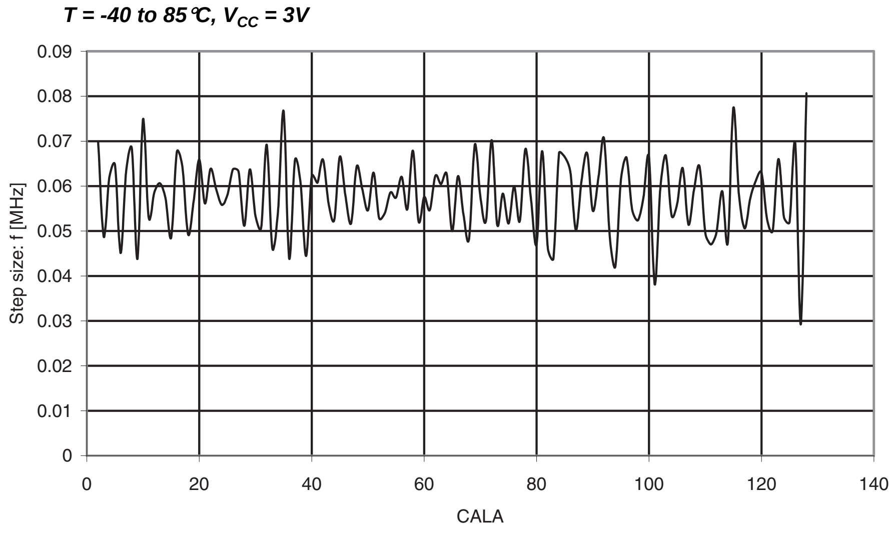
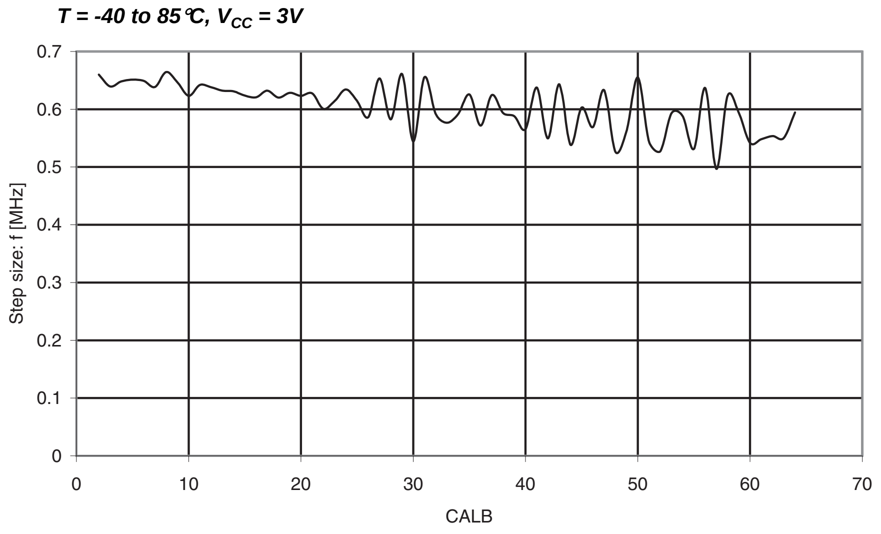
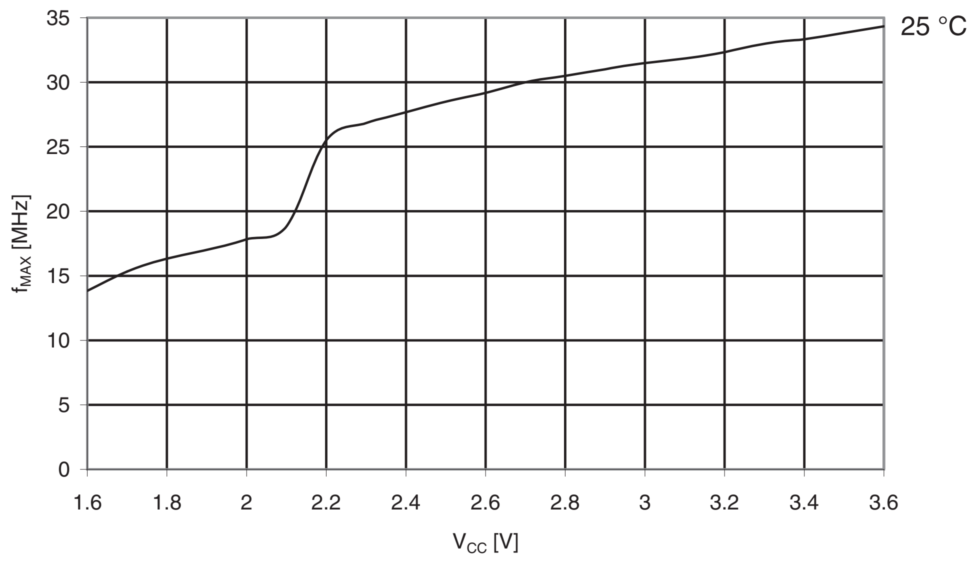


Figure 35-26.Internal 32 MHz Oscillator CalB Calibration Step Size



35.11 PDI Speed

Figure 35-27.PDI Speed vs. V_{CC}



36. Errata

36.1 ATxmega64A1and ATxmega128A1 rev. H

- Bandgap voltage input for the ACs can not be changed when used for both ACs simultaneously
- VCC voltage scaler for AC is non-linear
- The ADC has up to ±2 LSB inaccuracy
- ADC gain stage output range is limited to 2.4 V
- Sampling speed limited to 500 ksps for supply voltage below 2.0V
- ADC Event on compare match non-functional
- Bandgap measurement with the ADC is non-functional when VCC is below 2.7V
- Accuracy lost on first three samples after switching input to ADC gain stage
- The input difference between two succeeding ADC samples is limited by VREF
- Increased noise when using internal 1.0V reference at low temperature
- Configuration of PGM and CWCM not as described in XMEGA A Manual
- PWM is not restarted properly after a fault in cycle-by-cycle mode
- BOD will be enabled at any reset
- BODACT fuse location is not correct
- Sampled BOD in Active mode will cause noise when bandgap is used as reference
- DAC has up to ±10 LSB noise in Sampled Mode
- DAC is nonlinear and inaccurate when reference is above 2.4V or VCC - 0.6V
- DAC refresh may be blocked in S/H mode
- Conversion lost on DAC channel B in event triggered mode
- Both DFLLs and both oscillators have to be enabled for one to work
- Access error when multiple bus masters are accessing SDRAM
- EEPROM page buffer always written when NVM DATA0 is written
- Pending full asynchronous pin change interrupts will not wake the device
- Pin configuration does not affect Analog Comparator Output
- Low level interrupt triggered when pin input is disabled
- JTAG enable does not override Analog Comparator B output
- NMI Flag for Crystal Oscillator Failure automatically cleared
- Flash Power Reduction Mode can not be enabled when entering sleep
- Some NVM Commands are non-functional
- Crystal start-up time required after power-save even if crystal is source for RTC
- Setting PRHIRES bit makes PWM output unavailable
- Accessing EBI address space with PREBI set will lock Bus Master
- RTC Counter value not correctly read after sleep
- Pending asynchronous RTC-interrupts will not wake up device
- TWI, the minimum I2C SCL low time could be violated in Master Read mode
- TWI address-mask feature is non-functional
- TWI, a general address call will match independent of the R/W-bit value
- TWI Transmit collision flag not cleared on repeated start
- Clearing TWI Stop Interrupt Flag may lock the bus

- TWI START condition at bus timeout will cause transaction to be dropped
- TWI Data Interrupt Flag erroneously read as set
- WDR instruction inside closed window will not issue reset

1. Bandgap voltage input for the ACs cannot be changed when used for both ACs simultaneously

If the Bandgap voltage is selected as input for one Analog Comparator (AC) and then selected/deselected as input for another AC, the first comparator will be affected for up to 1 µs and could potentially give a wrong comparison result.

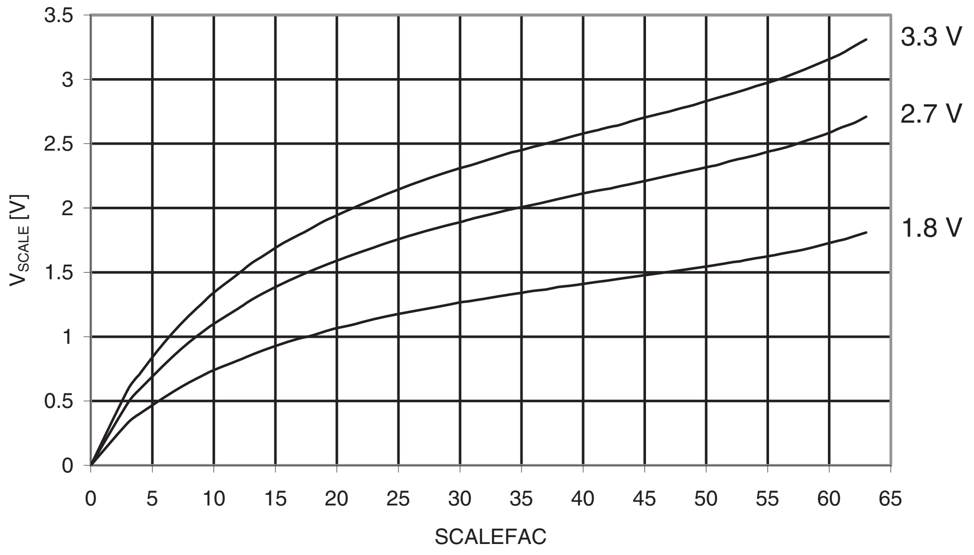
Problem fix/Workaround

If the Bandgap is required for both ACs simultaneously, configure the input selection for both ACs before enabling any of them.

2. VCC voltage scaler for AC is non-linear

The 6-bit VCC voltage scaler in the Analog Comparators is non-linear.

Figure 36-1. Analog Comparator Voltage Scaler vs. Scalefac
T = 25°C



Problem fix/Workaround

Use external voltage input for the analog comparator if accurate voltage levels are needed

3. The ADC has up to ±2 LSB inaccuracy

The ADC will have up to ±2 LSB inaccuracy, visible as a saw-tooth pattern on the input voltage/ output value transfer function of the ADC. The inaccuracy increases with increasing voltage reference reaching ±2 LSB with 3V reference.

Problem fix/Workaround

None, the actual ADC resolution will be reduced with up to ±2 LSB.

4. ADC gain stage output range is limited to 2.4 V

The amplified output of the ADC gain stage will never go above 2.4 V, hence the differential input will only give correct output when below 2.4 V/gain. For the available gain settings, this gives a differential input range of:

–	1x	gain:	2.4	V
–	2x	gain:	1.2	V
–	4x	gain:	0.6	V
–	8x	gain:	300	mV
–	16x	gain:	150	mV
–	32x	gain:	75	mV
–	64x	gain:	38	mV

Problem fix/Workaround

Keep the amplified voltage output from the ADC gain stage below 2.4 V in order to get a correct result, or keep ADC voltage reference below 2.4 V.

5. Sampling speed limited to 500 ksps for supply voltage below 2.0V

The sampling frequency is limited to 500 ksps for supply voltage below 2.0V. At higher sampling rate the INL error will be several hundred LSB.

Problem fix/Workaround

None.

6. ADC Event on compare match non-functional

ADC signalling event will be given at every conversion complete even if Interrupt mode (INTMODE) is set to BELOW or ABOVE.

Problem fix/Workaround

Enable and use interrupt on compare match when using the compare function.

7. Bandgap measurement with the ADC is non-functional when VCC is below 2.7V

The ADC can not be used to do bandgap measurements when VCC is below 2.7V.

Problem fix/Workaround

None.

8. Accuracy lost on first three samples after switching input to ADC gain stage

Due to memory effect in the ADC gain stage, the first three samples after changing input channel must be disregarded to achieve 12-bit accuracy.

Problem fix/Workaround

Run three ADC conversions and discard these results after changing input channels to ADC gain stage.

9. The input difference between two succeeding ADC samples is limited by VREF

If the difference in input between two samples changes more than the size of the reference, the ADC will not be able to convert the data correctly. Two conversions will be required before the conversion is correct.

Problem fix/Workaround

Discard the first conversion if input is changed more than VREF, or ensure that the input never changes more then VREF.

10. Increased noise when using internal 1.0V reference at low temperature

When operating at below 0°C and using internal 1.0V reference the RMS noise will be up 4 LSB, Peak-to-peak noise up to 25 LSB.

Problem fix/Workaround

Use averaging to remove noise.

11. Configuration of PGM and CWCM not as described in XMEGA A Manual

Enabling Common Waveform Channel Mode will enable Pattern generation mode (PGM), but not Common Waveform Channel Mode.

Enabling Pattern Generation Mode (PGM) and not Common Waveform Channel Mode (CWCM) will enable both Pattern Generation Mode and Common Waveform Channel Mode.

Problem fix/Workaround

PGM	CWCM	Description
0	0	PGM and CWCM disabled
0	1	PGM enabled
1	0	PGM and CWCM enabled
1	1	PGM enabled

12 PWM is not restarted properly after a fault in cycle-by-cycle mode

When the AWeX fault restore mode is set to cycle-by-cycle, the waveform output will not return to normal operation at first update after fault condition is no longer present.

Problem fix/Workaround

Do a write to any AWeX I/O register to re-enable the output.

13. BOD will be enabled after any reset

If any reset source goes active, the BOD will be enabled and keep the device in reset if the VCC voltage is below the programmed BOD level. During Power-On Reset, reset will not be released until VCC is above the programmed BOD level even if the BOD is disabled.

Problem fix/Workaround

Do not set the BOD level higher than VCC even if the BOD is not used.

14. BODACT fuse location is not correct

The fuses for enabling BOD in active mode (BODACT) are located at FUSEBYTE2, bit 2 and 3 and not in FUSEBYTE 5 as described in the XMEGA A Manual.

Problem fix/Workaround

Access the fuses in FUSEBYTE2.

15. Sampled BOD in Active mode will cause noise when bandgap is used as reference

Using the BOD in sampled mode when the device is running in Active or Idle mode will add noise on the bandgap reference for ADC, DAC and Analog Comparator.

Problem fix/Workaround

If the bandgap is used as reference for either the ADC, DAC or Analog Comparator, the BOD must not be set in sampled mode.

16. DAC has up to ±10 LSB noise in Sampled Mode

The DAC has noise of up to ±10 LSB in Sampled Mode for entire operation range.

Problem fix/Workaround

Use the DAC in continuous mode.

17. DAC is nonlinear and inaccurate when reference is above 2.4V or VCC - 0.6V

Using the DAC with a reference voltage above 2.4V or VCC - 0.6V will give inaccurate output when converting codes that give below 0.75V output:

- ±10 LSB for continuous mode
- ±200 LSB for Sample and Hold mode

Problem fix/Workaround

None.

18. DAC has up to ±10 LSB noise in Sampled Mode

If the DAC is running in Sample and Hold (S/H) mode and conversion for one channel is done at maximum rate (i.e. the DAC is always busy doing conversion for this channel), this will block refresh signals to the second channel.

Problem fix/Workaround

When using the DAC in S/H mode, ensure that none of the channels is running at maximum conversion rate, or ensure that the conversion rate of both channels is high enough to not require refresh.

19. Conversion lost on DAC channel B in event triggered mode

If during dual channel operation channel 1 is set in auto trigged conversion mode, channel 1 conversions are occasionally lost. This means that not all data-values written to the Channel 1 data register are converted.

Problem fix/Workaround

Keep the DAC conversion interval in the range 000-001 (1 and 3 CLK), and limit the Peripheral clock frequency so the conversion internal never is shorter than 1.5 µs.

20. Both DFLLs and both oscillators have to be enabled for one to work

In order to use the automatic runtime calibration for the 2 MHz or the 32 MHz internal oscillators, the DFLL for both oscillators and both oscillators have to be enabled for one to work.

Problem fix/Workaround

Enable both DFLLs and both oscillators when using automatic runtime calibration for either of the internal oscillators.

21. Access error when multiple bus masters are accessing SDRAM

If one bus master (CPU and DMA channels) is using the EBI to access an SDRAM in burst mode and another bus master is accessing the same row number in a different BANK of the SDRAM in the cycle directly after the burst access is complete, the access for the second bus master will fail.

Problem fix/Workaround

Do not put stack pointer in SDRAM and use DMA Controller in 1 byte burst mode if CPU and DMA Controller are required to access SDRAM at the same time.

22. EEPROM page buffer always written when NVM DATA0 is written

If the EEPROM is memory mapped, writing to NVM DATA0 will corrupt data in the EEPROM page buffer.

Problem fix/Workaround

Before writing to NVM DATA0, for example when doing software CRC or flash page buffer write, check if EEPROM page buffer active loading flag (EELOAD) is set. Do not write NVM DATA0 when EELOAD is set.

23. Pending full asynchronous pin change interrupts will not wake the device

Any full asynchronous pin-change Interrupt from pin 2, on any port, that is pending when the sleep instruction is executed, will be ignored until the device is woken from another source or the source triggers again. This applies when entering all sleep modes where the System Clock is stopped.

Problem fix/Workaround

None.

24. Pin configuration does not affect Analog Comparator Output

The Output/Pull and inverted pin configuration does not affect the Analog Comparator output function.

Problem fix/Workaround

None for Output/Pull configuration.

For inverted I/O, configure the Analog Comparator to give an inverted result (i.e. connect positive input to the negative AC input and vice versa), or use and external inverter to change polarity of Analog Comparator output.

25. Low level interrupt triggered when pin input is disabled

If a pin input is disabled, but pin is configured to trigger on low level, interrupt request will be sent.

Problem fix/Workaround

Ensure that Interrupt mask for the disabled pin is cleared.

26. JTAG enable does not override Analog Comparator B output

When JTAG is enabled this will not override the Analog Comparator B (ACB) output, AC0OUT on pin 7 if this is enabled.

Problem fix/Workaround

Use Analog Comparator output for ACA when JTAG is used, or use the PDI as debug interface.

27. NMI Flag for Crystal Oscillator Failure automatically cleared

NMI flag for Crystal Oscillator Failure (XOSCFDIF) will be automatically cleared when executing the NMI interrupt handler.

Problem fix/Workaround

This device revision has only one NMI interrupt source, so checking the interrupt source in software is not required.

28. Flash Power Reduction Mode can not be enabled when entering sleep

If Flash Power Reduction Mode is enabled when entering Power-save or Extended Standby sleep mode, the device will only wake up on every fourth wake-up request. If Flash Power Reduction Mode is enabled when entering Idle sleep mode, the wake-up time will vary with up to 16 CPU clock cycles.

Problem fix/Workaround

Disable Flash Power Reduction mode before entering sleep mode.

29. Some NVM Commands are non-functional

The following NVM commands are non-functional:

–	0x2B	Erase Flash Page
–	0x2E	Write Flash Page
–	0x2F	Erase & Write Flash Page
–	0x3A	Flash Range CRC

Problem fix/Workaround

None for Flash Range CRC

Use separate programming commands for accessing application and boot section.

–	0x22	Erase Application Section Page
–	0x24	Write Application Section Page
–	0x25	Erase & Write Application Section Page
–	0x2A	Erase Boot Loader Section Page
–	0x2C	Write Boot Loader Section Page
–	0x2D	Erase & Write Boot Loader Section Page

30. Crystal start-up time required after power-save even if crystal is source for RTC

Even if 32.768 kHz crystal is used for RTC during sleep, the clock from the crystal will not be ready for the system before the specified start-up time. See "XOSCSEL[3:0]: Crystal Oscillator Selection" in XMEGA A Manual. If BOD is used in active mode, the BOD will be on during this period (0.5s).

Problem fix/Workaround

If faster start-up is required, go to sleep with internal oscillator as system clock.

31. Setting PRHIRES bit makes PWM output unavailable

Setting the HIRES Power Reduction (PR) bit for PORTx will make any Frequency or PWM output for the corresponding Timer/Counters (TCx0 and TCx1) unavailable on the pin even if the Hi-Res is not used.

Problem fix/Workaround

Do not write the HIRES PR bit on PORTx when frequency or PWM output from TCx0/1 is used.

32. Accessing EBI address space with PREBI set will lock Bus Master

If EBI Power Reduction Bit is set while EBI is enabled, accessing external memory could result in bus hang-up, blocking all further access to all data memory.

Problem fix/Workaround

Ensure that EBI is disabled before setting EBI Power Reduction bit.

33. RTC Counter value not correctly read after sleep

If the RTC is set to wake up the device on RTC Overflow and bit 0 of RTC CNT is identical to bit 0 of RTC PER as the device is entering sleep, the value in the RTC count register can not be read correctly within the first prescaled RTC clock cycle after wakeup. The value read will be the same as the value in the register when entering sleep.

The same applies if RTC Compare Match is used as wake-up source.

Problem fix/Workaround

Wait at least one prescaled RTC clock cycle before reading the RTC CNT value.

34. Pending asynchronous RTC-interrupts will not wake up device

Asynchronous Interrupts from the Real-Time-Counter that is pending when the sleep instruction is executed, will be ignored until the device is woken from another source or the source triggers again.

Problem fix/Workaround

None.

35. TWI, the minimum I^{2}C SCL low time could be violated in Master Read mode

If the TWI is in Master Read mode and issues a Repeated Start on the bus, this will immediately release the SCL line even if one complete SCL low period has not passed. This means that the minimum SCL low time in the I2C specification could be violated.

Problem fix/Workaround

If this is a problem in the application, ensure in software that the Repeated Start is never issued before one SCL low time has passed.

36. TWI address-mask feature is non-functional

The address-mask feature is non-functional, so the TWI can not perform hardware address match on more than one address.

Problem fix/Workaround

If the TWI must respond to multiple addresses, enable Promiscuous Mode for the TWI to respond to all address and implement the address-mask function in software.

37. TWI, a general address call will match independent of the R/W-bit value

When the TWI is in Slave mode and a general address call is issued on the bus, the TWI Slave will get an address match regardless of the received R/W bit.

Problem fix/Workaround

Use software to check the R/W-bit on general call address match.

38. TWI Transmit collision flag not cleared on repeated start

The TWI transmit collision flag should be automatically cleared on start and repeated start, but is only cleared on start.

Problem fix/Workaround

Clear the flag in software after address interrupt.

39. Clearing TWI Stop Interrupt Flag may lock the bus

If software clears the STOP Interrupt Flag (APIF) on the same Peripheral Clock cycle as the hardware sets this flag due to a new address received, CLKHOLD is not cleared and the SCL line is not released. This will lock the bus.

Problem fix/Workaround

Check if the bus state is IDLE. If this is the case, it is safe to clear APIF. If the bus state is not IDLE, wait for the SCL pin to be low before clearing APIF.

Code:

```
/* Only clear the interrupt flag if within a "safe zone". */
while ( /* Bus not IDLE: */
       ((COMMS_TWI.MASTER.STATUS & TWI_MASTER_BUSSTATE_gm) !=
         TWI_MASTER_BUSSTATE_IDLE_gc)) &&
         /* SCL not held by slave: */
         !(COMMS_TWI.SLAVE.STATUS & TWI_SLAVE_CLKHOLD_bm)
      )
{
    /* Ensure that the SCL line is low */
    if ( !(COMMS_PORT.IN & PIN1_bm) )
        if ( !(COMMS_PORT.IN & PIN1_bm) )
            break;
}
/* Check for an pending address match interrupt */
if ( !(COMMS_TWI.SLAVE.STATUS & TWI_SLAVE_CLKHOLD_bm) )
{
    /* Safely clear interrupt flag */
    COMMS_TWI.SLAVE.STATUS |= (uint8_t)TWI_SLAVE_APIF_bm;
}
```

40. TWI START condition at bus timeout will cause transaction to be dropped

If Bus Timeout is enabled and a timeout occurs on the same Peripheral Clock cycle as a START is detected, the transaction will be dropped.

Problem fix/Workaround

None.

41. TWI Data Interrupt Flag erroneously read as set

When issuing the TWI slave response command CMD=0b11, it takes 1 Peripheral Clock cycle to clear the data interrupt flag (DIF). A read of DIF directly after issuing the command will show the DIF still set.

Problem fix/Workaround

Add one NOP instruction before checking DIF.

42. WDR instruction inside closed window will not issue reset

When a WDR instruction is execute within one ULP clock cycle after updating the window control register, the counter can be cleared without giving a system reset.

Problem fix/Workaround

Wait at least one ULP clock cycle before executing a WDR instruction.

36.2 ATxmega64A1 and ATxmega128A1 rev. G

- Bootloader Section in Flash is non-functional
- Bandgap voltage input for the ACs cannot be changed when used for both ACs simultaneously
- DAC is nonlinear and inaccurate when reference is above 2.4V
- ADC gain stage output range is limited to 2.4 V
- The ADC has up to ±2 LSB inaccuracy
- TWI, a general address call will match independent of the R/W-bit value
- TWI, the minimum I^{2}C SCL low time could be violated in Master Read mode
- Setting HIRES PR bit makes PWM output unavailable
- EEPROM erase and write does not work with all System Clock sources
- BOD will be enabled after any reset
- Propagation delay analog Comparator increasing to 2 ms at -40°C
- Sampled BOD in Active mode will cause noise when bandgap is used as reference
- Default setting for SDRAM refresh period too low
- Flash Power Reduction Mode can not be enabled when entering sleep mode
- Enabling Analog Comparator B output will cause JTAG failure
- JTAG enable does not override Analog Comparator B output
- Bandgap measurement with the ADC is non-functional when V_{CC} is below 2.7V
- DAC refresh may be blocked in S/H mode
- Inverted I/O enable does not affect Analog Comparator Output
- Both DFLLs and both oscillators has to be enabled for one to work

1. Bootloader Section in Flash is non-functional

The Bootloader Section is non-functional, and bootloader or application code cannot reside in this part of the Flash.

Problem fix/Workaround

None, do not use the Bootloader Section.

2. Bandgap voltage input for the ACs cannot be changed when used for both ACs simultaneously

If the Bandgap voltage is selected as input for one Analog Comparator (AC) and then selected/deselected as input for the another AC, the first comparator will be affected for up to 1 us and could potentially give a wrong comparison result.

Problem fix/Workaround

If the Bandgap is required for both ACs simultaneously, configure the input selection for both ACs before enabling any of them.

3. DAC is nonlinear and inaccurate when reference is above 2.4V

Using the DAC with a reference voltage above 2.4V give inaccurate output when converting codes that give below 0.75V output:

- ±20 LSB for continuous mode

- ±200 LSB for Sample and Hold mode

Problem fix/Workaround

None, avoid using a voltage reference above 2.4V.

4. ADC gain stage output range is limited to 2.4 V

The amplified output of the ADC gain stage will never go above 2.4 V, hence the differential input will only give correct output when below 2.4 V/gain. For the available gain settings, this gives a differential input range of:

–	1x	gain:	2.4	V
–	2x	gain:	1.2	V
–	4x	gain:	0.6	V
–	8x	gain:	300	mV
–	16x	gain:	150	mV
–	32x	gain:	75	mV
–	64x	gain:	38	mV

Problem fix/Workaround

Keep the amplified voltage output from the ADC gain stage below 2.4 V in order to get a correct result, or keep ADC voltage reference below 2.4 V.

5. The ADC has up to ±2 LSB inaccuracy

The ADC will have up to ±2 LSB inaccuracy, visible as a saw-tooth pattern on the input voltage/ output value transfer function of the ADC. The inaccuracy increases with increasing voltage reference reaching ±2 LSB with 3V reference.

Problem fix/Workaround

None, the actual ADC resolution will be reduced with up to ±2 LSB.

6. TWI, a general address call will match independent of the R/W-bit value

When the TWI is in Slave mode and a general address call is issued on the bus, the TWI Slave will get an address match regardless of the R/W-bit (ADDR[0] bit) value in the Slave Address Register.

Problem fix/Workaround

Use software to check the R/W-bit on general call address match.

7. TWI, the minimum I^{2}C SCL low time could be violated in Master Read mode

When the TWI is in Master Read mode and issuing a Repeated Start on the bus, this will immediately release the SCL line even if one complete SCL low period has not passed. This means that the minimum SCL low time in the I^{2}C specification could be violated.

Problem fix/Workaround

If this causes a potential problem in the application, software must ensure that the Repeated Start is never issued before one SCL low time has passed.

8. Setting HIRES PR bit makes PWM output unavailable

Setting the HIRES Power Reduction (PR) bit for PORTx will make any Frequency or PWM output for the corresponding Timer/Counters (TCx0 and TCx1) unavailable on the pin.

Problem fix/Workaround

Do not write the HIRES PR bit on PORTx when frequency or PWM output from TCx0/1 is used.

9. EEPROM erase and write does not work with all System Clock sources

When doing EEPROM erase or Write operations with other clock sources than the 2 MHz RCOSC, Flash will be read wrongly for one or two clock cycles at the end of the EEPROM operation.

Problem fix/Workaround

Alt 1: Use the internal 2 MHz RCOSC when doing erase or write operations on EEPROM.

Alt 2: Ensure to be in sleep mode while completing erase or write on EEPROM. After starting erase or write operations on EEPROM, other interrupts should be disabled and the device put to sleep.

10. BOD will be enabled after any reset

If any reset source goes active, the BOD will be enabled and keep the device in reset if the VCC voltage is below the programmed BOD level. During Power-On Reset, reset will not be released until VCC is above the programmed BOD level even if the BOD is disabled.

Problem fix/Workaround

Do not set the BOD level higher than VCC even if the BOD is not used.

11. Propagation delay analog Comparator increasing to 2 ms at -40 °C

When the analog comparator is used at temperatures reaching down to -40 °C, the propagation delay will increase to ~2 ms.

Problem fix/Workaround

None

12. **Sampled BOD in Active mode will cause noise when bandgap is used as reference**

Using the BOD in sampled mode when the device is running in Active or Idle mode will add noise on the bandgap reference for ADC and DAC.

Problem fix/Workaround

If the bandgap is used as reference for either the ADC or the DAC, the BOD must not be set in sampled mode.

13. **Default setting for SDRAM refresh period too low**

If the SDRAM refresh period is set to a value less then 0x20, the SDRAM content may be corrupted when accessing through On-Chip Debug sessions.

Problem fix/Workaround

The SDRAM refresh period (REFRESHH/L) should not be set to a value less then 0x20.

14. **Flash Power Reduction Mode can not be enabled when entering sleep mode**

If Flash Power Reduction Mode is enabled when entering Power-save or Extended Standby sleep mode, the device will only wake up on every fourth wake-up request.

If Flash Power Reduction Mode is enabled when entering Idle sleep mode, the wake-up time will vary with up to 16 CPU clock cycles.

Problem fix/Workaround

Disable Flash Power Reduction mode before entering sleep mode.

15. **JTAG enable does not override Analog Comparator B output**

When JTAG is enabled this will not override the Anlog Comparator B (ACB)ouput, AC0OUT on pin 7 if this is enabled.

Problem fix/Workaround

AC0OUT for ACB should not be enabled when JTAG is used. Use only analog comparator output for ACA when JTAG is used, or use the PDI as debug interface.

16. **Bandgap measurement with the ADC is non-functional when V_{CC} is below 2.7V**

The ADC cannot be used to do bandgap measurements when V_{CC} is below 2.7V.

Problem fix/Workaround

If internal voltages must be measured when V_{CC} is below 2.7V, measure the internal 1.00V reference instead of the bandgap.

17. DAC refresh may be blocked in S/H mode

If the DAC is running in Sample and Hold (S/H) mode and conversion for one channel is done at maximum rate (i.e. the DAC is always busy doing conversion for this channel), this will block refresh signals to the second channel.

Problem fix/Workarund

When using the DAC in S/H mode, ensure that none of the channels is running at maximum conversion rate, or ensure that the conversion rate of both channels is high enough to not require refresh.

18. Inverted I/O enable does not affect Analog Comparator Output

The inverted I/O pin function does not affect the Analog Comparator output function.

Problem fix/Workarund

Configure the analog comparator setup to give a inverted result (i.e. connect positive input to the negative AC input and vice versa), or use and externel inverter to change polarity of Analog Comparator Output.

19. Both DFLLs and both oscillators has to be enabled for one to work

In order to use the automatic runtime calibration for the 2 MHz or the 32 MHz internal oscilla-tors, the DFLL for both oscillators and both oscillators has to be enabled for one to work.

Problem fix/Workarund

Enabled both DFLLs and oscillators when using automatic runtime calibration for one of the internal oscillators.

37. Datasheet Revision History

Please note that the referring page numbers in this section are referred to this document. The referring revision in this section are referring to the document revision.

37.1 8067O – 06/2013

1.	Not recommended for new designs - Use XMEGA A1U series.

37.2 8067N – 03/2013

1.	Removed all references to ATxmega192A1, ATxmega256A1 and ATxmega384A1.
2.	Updated module description. Based on the XMEGA A1U device datasheet.
3.	Updated analog comparator (AC) overview, Figure 28-1 on page 53.
4.	Updated “ADC Characteristics” on page 76.
5	Updated page erase time in “Flash and EEPROM Memory Characteristics” on page 76.
6	Updated Output low voltage conditions from I_{OH} to I_{OL} in “PAD Characteristics” on page 79.
7.	Removed TBDs from: “DC Characteristics” on page 73. “DAC Characteristics” on page 78. “Bandgap Characteristics” on page 78.
8.	Updated “Errata” on page 96 to be valid for both ATxmega64A1 and ATxmega128A1.
9.	Removed Boundary Scan Order table.

37.3 8067M – 09/2010

1.	Updated Errata “ATxmega64A1and ATxmega128A1 rev. H” on page 96

37.4 8067L – 08/2010

1.	Removed Footnote 3 of Figure 2-1 on page 3
2.	Updated “Features” on page 32. Event Channel 0 output on port pin 7
3.	Updated “DC Characteristics” on page 73, by adding I_{CC} for Flash/EEPROM Programming.
4.	Added AVCC in “ADC Characteristics” on page 76.
5.	Updated Start up time in “ADC Characteristics” on page 76.

6.	Updated “DAC Characteristics” on page 78. Removed DC output impedance.
7.	Fixed typo in “Packaging information” section.
8.	Fixed typo in “Errata” section.

37.5 8067K – 02/2010

1.	Added “PDI Speed vs. VCC” on page 95.

37.6 8067J – 02/2010

1.	Removed JTAG Reset from the datasheet.
2.	Updated “Timer/Counter and AWEX functions” on page 56.
3.	Updated “Alternate Pin Functions” on page 58.
3.	Updated all “Electrical Characteristics” on page 73.
4.	Updated “PAD Characteristics” on page 79.
5.	Changed Internal Oscillator Speed to “Oscillators and Wake-up Time” on page 92.
6.	Updated “Errata” on page 96

37.7 8067I – 04/2009

1.	Updated “Ordering Information” on page 2.
2.	Updated “PAD Characteristics” on page 79.

37.8 8067H – 04/2009

1.	Editorial updates.
2.	Updated “Overview” on page 54.
3.	Updated Table 29-9 on page 54.
4.	Updated “Peripheral Module Address Map” on page 62. IRCOM has address map: 0x08F8.
5.	Updated “Electrical Characteristics” on page 73.
6.	Updated “PAD Characteristics” on page 79.
7.	Updated “Typical Characteristics” on page 82.

37.9 8067G – 11/2008

1.	Updated “Block Diagram” on page 6.
2.	Updated feature list in “Memories” on page 12.
3.	Updated “Programming and Debugging” on page 54.
4.	Updated “Peripheral Module Address Map” on page 62. IRCOM has address 0x8F0.
5.	Added “Electrical Characteristics” on page 73.
6.	Added “Typical Characteristics” on page 82.
7.	Added “ATxmega64A1and ATxmega128A1 rev. H” on page 96.
8.	Updated “ATxmega64A1 and ATxmega128A1 rev. G” on page 107.

37.10 8067F – 09/2008

1.	Updated “Features” on page 1
2.	Updated “Ordering Information” on page 2
3.	Updated Figure 7-1 on page 11 and Figure 7-2 on page 11.
4.	Updated Table 7-2 on page 15.
5.	Updated “Features” on page 48 and “Overview” on page 48.
6.	Removed “Interrupt Vector Summary” section from datasheet.

37.11 8067E – 08/2008

1.	Changed Figure 2-1’s title to “Block diagram and pinout” on page 3.
2.	Updated Figure 2-2 on page 4.
3.	Updated Table 29-2 on page 51 and Table 29-3 on page 52.

37.12 8067D – 07/2008

1.	Updated “Ordering Information” on page 2.
2.	Updated “Peripheral Module Address Map” on page 62.
3.	Inserted “Interrupt Vector Summary” on page 56.

37.13 8067C – 06/2008

1.	Updated the Front page and "Features" on page 1.
2.	Updated the "DC Characteristics" on page 73.
3.	Updated Figure 3-1 on page 6.
4.	Added "Flash and EEPROM Page Size" on page 15.
5.	Updated Table 33-6 on page 72 with new data: Gain Error, Offset Error and Signal -to-Noise Ratio (SNR).
6.	Updated Errata "ATxmega64A1 and ATxmega128A1 rev. G" on page 107.

37.14 8067B – 05/2008

1.	Updated "Pinout/Block Diagram" on page 3 and "Pinout and Pin Functions" on page 55.
2.	Added XMEGA A1 Block Diagram, Figure 3-1 on page 6.
3.	Updated "Overview" on page 5 included the XMEGA A1 explanation text on page 6.
4.	Updated AVR CPU "Features" on page 8.
5.	Updated Event System block diagram, Figure 10-1 on page 20.
6.	Updated "Interrupts and Programmable Multilevel Interrupt Controller" on page 29.
7.	Updated "AC - Analog Comparator" on page 52.
8.	Updated "Alternate Pin Function Description" on page 55.
9.	Updated "Alternate Pin Functions" on page 58.
10.	Updated "Typical Characteristics" on page 82.
11.	Updated "Ordering Information" on page 2.
12.	Updated "Overview" on page 5.
13.	Updated Figure 6-1 on page 8.
14.	Inserted a new Figure 16-1 on page 37.
15.	Updated Speed grades in "Speed" on page 75.
16.	Added a new ATxmega384A1 device in "Features" on page 1, updated "Ordering Information" on page 2 and "Memories" on page 12.
17.	Replaced the Figure 3-1 on page 6 by a new XMEGA A1 detailed block diagram.
18.	Inserted Errata "ATxmega64A1 and ATxmega128A1 rev. G" on page 107.

37.15 8067A – 02/2008

1.	Initial revision.

Table of Contents

Atmel® | Enabling Unlimited Possibilities®

Atmel Corporation
1600 Technology Drive
San Jose, CA 95110
USA
Tel: (+1) (408) 441-0311
Fax: (+1) (408) 487-2600
www.atmel.com

Atmel Asia Limited
Unit 01-5 & 16, 19F
BEA Tower, Millennium City 5
418 Kwun Tong Roa
Kwun Tong, Kowloon
HONG KONG
Tel: (+852) 2245-6100
Fax: (+852) 2722-1369

Atmel Munich GmbH
Business Campus
Parkring 4
D-85748 Garching b. Munich
GERMANY
Tel: (+49) 89-31970-0
Fax: (+49) 89-3194621

Atmel Japan G.K.
16F Shin-Osaki Kangyo Bldg
1-6-4 Osaki, Shinagawa-ku
Tokyo 141-0032
JAPAN
Tel: (+81) (3) 6417-0300
Fax: (+81) (3) 6417-0370