ATMEL WHITE PAPER

Introducing a New Breed of Microcontrollers for 8/16-bit Applications

By Kristian Saether, AVR Product Marketing Manager
&
Ingar Fredriksen, AVR Product Marketing Director

Summary

System performance is not the same as computing power. Low power consumption, good peripherals and real-time signal handling is more important for many embedded application. Traditionally 8/16-bit microcontrollers have handled all these challenges well, but with a continuous increase in functionality and feature requirements in embedded designs, many microcontroller companies now start to struggle with their traditional 8/16-bit MCU offering. Instead some companies offer a 32-bit microcontroller not optimum for the task. This whitepaper discusses the key challenges that 8/16-bit embedded developers meet and how the new XMEGA™ AVR® family from Atmel® brings 8/16-bit microcontrollers up to a new level of system performance.

Atmel Corporation • 2325 Orchard Parkway • San Jose, CA 95131
TEL (408) 441-0311 • FAX (408) 487-2600 • Web Site: *http://www.atmel.com*

Table of Contents

Understanding System Performance

Is the million dollar answer to "embedded system performance" only four letters? The obvious candidate, MIPS - million instructions per second, gives an indication of a microcontroller's computing ability. However, all embedded applications require more than computing power. The reason why there is no finite answer to system performance is because there are as many requirements as there are applications. Most of the requirements are equally important and potentially hard to combine. One application might be cost driven and requires a highly integrated microcontroller with multiple timers and communication interfaces, whereas another product needs high accuracy, fast analog features. Common to both may be the power source, for instance a battery. In real-time applications consequences may be catastrophic if communication fails. This requires an agile microcontroller with the ability to process tasks in the correct order and without varying response times. A shared challenge for all these applications may be the need for periodic field upgrades.

Independent of the specific product challenges, system performance is also about easy-to-use development tools, application examples and good documentation, and efficient support network.

System Performance Challenges

Reaching the boundaries for older 8-bit CPUs

Focus on system performance is important because more and more 8/16-bit microcontroller families do not meet today's requirements. Old and inefficient architectures limit processing power, memory size, peripheral handling and low power capabilities. 8-bit architecture such as 8051, PIC14, PIC16, PIC18, 78K0, and HC08 were developed before the emergence of high-level languages such as C. The instruction set was made for assembler development only, and the CPUs lack key functionality such as 16-bit arithmetic support, conditional jumps and memory pointers.

Many CPU architectures require several clock cycles per instruction. Microchip's 8-bit PIC families use 4 clock cycles for the simplest instructions. This results in 5 MIPS out of a 20 MHz clock. Other CPU architectures like the 8051 core use minimum 6 clock cycles per instruction, resulting in even less MIPS at a given clock speed.

```
int max(int *array)
{
  char a;
  int maximum=-32768;

  for (a=0;a<16;a++)
     if (array[a]>maximum)
        maximum=array[a];
  return (maximum);
}
```

Above is an example of a simple C code function. Table 1 shows the result of the same code example compiled on three different CPU architectures. The 8051 CPU has almost 4 times longer execution time than a PIC16 CPU, and 28 times longer execution time than an AVR CPU.

Table 1. Code size and execution time for different CPU architectures.

CPU Architectures	Code size (bytes)	Execution time (cycles)
8051	112	9384
PIC16	87	2492
AVR	46	335

Some companies have solved the clock division issue and offers microcontrollers with one-cycle instructions. With clock speeds up to 100 MHz, Silicon Labs claims up to 100 MIPS peak in their 8-bit 8051 based microcontrollers. There are issues, though. Since most instructions use two clock cycles and more, the real output is closer to 50 MIPS. Since the 8051 is an accumulator based CPU, all data that requires computing need to be copied to the accumulator. Inspecting assembly code made for 8051 CPUs shows that 65-70% of the instructions are used to move data. So the 50 “8051” MIPS are equivalent to 15 MIPS of a modern 8/16-bit CPU architecture using a file of registers that are all connected to the Arithmetic Logic Unit (ALU).

```
MOV     A,0x82
ADD     A,R1
MOV     0x82,A
MOV     A,0x83
ADDC    A,R2
MOV     0x83,A
MOVX    A,@DPTR
MOV     OxF0,A
INC     DPTR
MOVX    A,@DPTR
RET
```

A third problem for older CPU architectures is the lack of support for larger memory spaces. In the seventies it was hard to imagine a need for more than 64KB of memory in embedded applications. This led many CPU designers to choose a 16-bit address bus. The CPU, registers, instructions, and data buses for program and data are still limited by this. In 2006, 9% of 8-bit embedded applications used 64KB or more program memory, representing 26% of the 8-bit MCU revenue. Projections for 2009 say 14% of the 8-bit embedded applications representing 36% of the revenue will use 64KB or more program memory. [Source: MCU memory trends. Semico 2007]

All this makes older 8-bit MCUs less competitive from a system performance perspective.

Using 32-bit microcontrollers for the wrong reasons

To solve older CPU architectures inability to meet market needs, many vendors promote 32-bit solutions. This is great for applications that need 32-bit processing power, but many designers switch to 32-bit for the wrong reasons. Solving the 8/16-bit microcontroller limitations with a 32-bit MCU comes at a high price.

Most 32-bit microcontrollers do not offer high speed, high resolution Analog to Digital Converters. EMC performance is normally lower and ESD protection weaker. High I/O drive strength, many internal and external oscillator options and no need for external components to accommodate on-chip voltage regulators are other benefits of 8/16-bit microcontrollers.

A 32-bit CPU contains significantly more digital logic than any 8/16-bit CPU. This increases manufacturing cost. Migrating to a smaller manufacturing process geometry reduces the cost penalty. The drawback is a much higher leakage and static current consumption. Applications like water meters, gas meters, thermostats, key fobs, toll road tags, alarm systems etc. spend a majority of their time in sleep mode where the CPU is stopped. Since battery life of such applications must be 5-10 years, a transition from 8/16-bit to 32-bit microcontrollers is not possible. The requirement for increased system performance needs to be solved in different ways.

Keeping it within the family

Companies making embedded products periodically have to upgrade their portfolio to be competitive. These new products normally arise from the need for refining, updating, or reducing the product cost. Other factors include process updates, competitors, and market trends. Independent of the initial causes, the new product is always based on some core ideas. The new or updated version can thus rely on already existing platforms and source code.

Studies show that the possibility to re-use existing hardware and software reduce the development time with more than 50%. The reasons for this reduction can be reuse of software or hardware. Engineer's knowledge of a specific MCU product family, its documentation and development tools is also a major factor to minimize development time and cost.

Protecting Intellectual Property and handling confidential information

Some embedded applications handle personal information. Others grant access to restricted areas or monetary funds. Almost all microcontrollers run intellectual property software. The owners of this IP may lose future revenues if the software is lost to hackers and copy products are introduced in the market. Due to this most MCUs have implemented protection mechanisms that prevent hackers or third parties from using programming, debugging or test interfaces to read out the program memory.

A much more difficult and relevant problem today is the increased use of communication and wireless signaling between applications or modules. To restrict third party access encryption must be implemented. An example of this is remote keyless car entry or wireless home appliances. Without encrypted data transfer anyone could access your car or use your wireless internet connection. A traditional way to implement this is to invent

closed or secret algorithms so that only those who know the algorithm can use it. This is known as security by obscurity, but it is a very dangerous solution. To come around this problem, open and public algorithms such as AES and DES allow for public inspection and approval of their security. If you use secret or obscure encryption algorithms, there is no way to review the security level or discover critical design flaws.

The problem with open encryption standards are that they put heavy requirements on computing power and secure algorithm design. DES encryption or decryption of an 8 byte block typically uses around 100,000 cycles on a modern 8-bit MCU. This equals 15 MIPS of pure processing power to support 9600 kbps communication speed. 32-bit CPUs will do typically do this 50-60% faster. Using 45,000 cycles or 1.4 ms at 32 MHz only allow for 45 kbps secure data communication until all the CPU time is used for encryption and decryption. The obvious problem is that little time is left for any actual application processing. Secondly, most wireless applications are battery driven, and if most of processing power must be used on crypto it will shorten battery life.

High integration versus interrupt latency and safety

Microcontrollers include more and more peripherals, and often the peripheral is of key importance to how good a microcontroller solves its task. They are used for sensor input, system control, data communication, fault control, timing and much more. The traditional approach is to have interrupts to control peripherals. The advantage is that software does not have to poll each peripheral for status. One disadvantage is that processing cycles are used for context switching between interrupt routines. In modern CPU architectures 20-100 cycles are used to switch context. Imagine a simple task as receiving incoming SPI data. At 1 Mbps, the SPI receive interrupt will trigger at a rate of 125 kHz. If the SPI interrupt takes 25 cycles including context switching (1.25 ms), more than 15% of 20 MIPS CPU time is required just to handle the SPI interrupt. As more interrupts are used this soon adds up to an overwhelming number of task for the CPU.

Another disadvantage with interrupts is the response time to critical system events. Some interrupt sources may need immediate response when triggered. Examples are airbags for car, emergency shutdown of power tools or other critical applications. All require immediate response and shut down of the control system to prevent permanent and fatal failure. If the MCU needs to finish other interrupt service routines or uses too long switching context, failure detection will be too slow and unpredictable.

The AVR XMEGA Solves the System Performance Challenge

The increased need for higher embedded performance combined with reduced cost, size and power consumption mandate new microcontrollers that are designed especially to meet all these requirements in the best possible way. In 2008 Atmel introduces the AVR XMEGA microcontroller family. XMEGA offers a significant technology upgrade for AVR and extends the application boundaries for 8/16-bit microcontrollers to meet the requirements for the next generation of embedded design.

CPU performance

High Level Languages like C have become the standard methodology for embedded microcontrollers due to improved time to market and simplified maintenance support.

XMEGA uses the same high-performance core as all existing AVR microcontrollers. The AVR core was designed in the 1990's to take advantage of semiconductor integration and software capabilities unknown when earlier 8 bit cores was designed. It uses a Harvard architecture, where the program memory space is separated from the data memory space. Program memory is accessed with single level pipelining. While one instruction is being executed, the next instruction is pre-fetched from the program memory. Central to the AVR performance is the fast-access RISC register file, which consists of 32 x 8-bit general purpose working registers. Within one single clock cycle, AVR can feed two arbitrary registers from the register file to the ALU, do a requested operation, and write back the result to an arbitrary register. The AVR core also incorporates support for atomic 16-bit register access, 16-and 32-bit arithmetic, and three 16/24-bit memory pointers. The result of this is a single cycle core with up to 1MIPS/MHz, high code density, linear address map up to 16 Mbytes, and unmatched performance in the 8/16 bit market.

In XMEGA devices processing performance is increased with higher processor speeds over the entire voltage range. AVR XMEGA devices can operate up to 12 MHz at 1.6V and 32 MHz from 2.7V. In addition to improved CPU performance, XMEGA has a clever set of peripherals that offload the CPU to further improve system performance,

The AVR core is already established as the industry leader in the 8-bit market; with the new XMEGA family of microcontrollers the entire 8- and 16-bit market is covered by one microcontroller family.

Flexible 4-channel DMA controller

The XMEGA features a 4 channel Direct Memory Access (DMA) controller that can move data between any locations in the data memory space. Data can be moved from a peripheral register to internal or external SRAM, between SRAM locations, and even between peripheral registers directly. The four DMA channels have individual priority, source, destination, triggers, addressing modes, and transfer block sizes. The DMA can transmit from 1 byte to 16 Mbytes in a single transfer. The large data transfer size is possible due to the simple linear data memory address space in the AVR, and auto increment/decrement and reload features in the DMA controller.

Table 1. XMEGA UART communication with and without DMA

Transfer Speed [kbps]	CPU usage with DMA controller [%]	CPU usage without DMA controller [%]
9.6	0.01	0.26
19.2	0.01	0.52
38.4	0.03	1.04
57.6	0.04	1.57
115.2	0.08	3.14
1200	0.85	34.15
3500	5.17	99.59

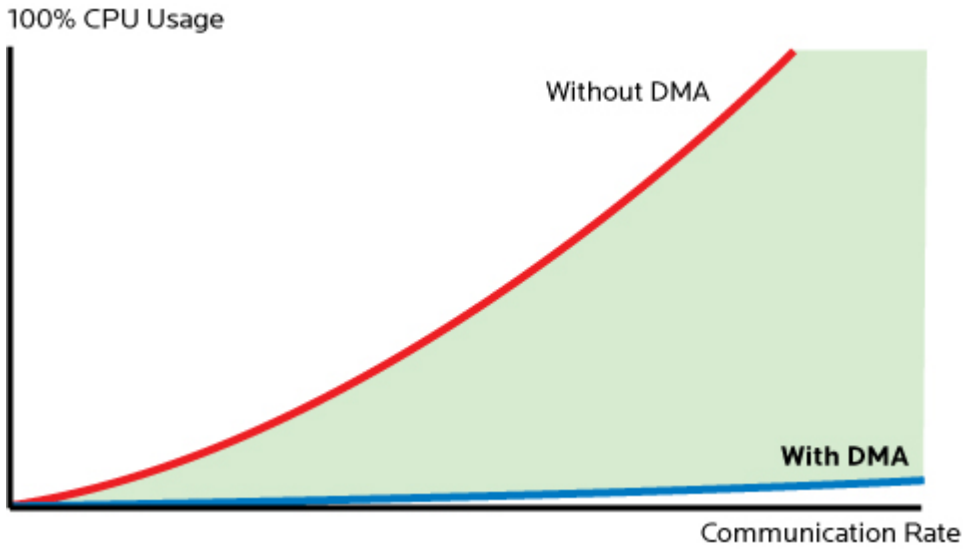


Figure 1. XMEGA UART communication with and without DMA.

The DMA controller uses the same data bus as the CPU to transfer data. As a consequence of the Harvard architecture and the 32 general purpose registers of the AVR, the data bus is only used by 15 of the 132 instructions available in XMEGA. When the bus is free the DMA Controller can move data between memories and peripherals without using CPU resources. In addition to using the data bus when it is free, the internal buses to the peripheral registers including IO pins, memory mapped EEPROM, the internal SRAM, and the External Bus Interface are split to enable simultaneous bus access from the DMA Controller and the CPU. Hence there is always a communication channel available for the DMA.

The DMA controller is able to handle data transfer with minimal CPU intervention. It enables both higher performance and lower power consumption than solutions without DMA controller for embedded applications that include data transfer.

Innovative 8-channel Event System

Like a reflex in the human body the innovative XMEGA Event System enables inter-peripheral communication without CPU or DMA usage. Events are routed between peripherals through a dedicated network outside the CPU, data bus and DMA controller. The benefit of this is predictable and latency-free inter-peripheral signal communication; reducing CPU time and freeing interrupt resources. The Event System enables the possibility for a change of state in one peripheral to automatically trigger actions in other peripherals. This is extremely powerful as it allows for autonomous and predictable control of peripherals without using any interrupts or CPU resources.

Which event should trigger what event action is fully configurable and up to the designer to decide. The configuration can be kept static and locked, or be dynamic and change during various stages of the application execution. Up to 8 pairs of peripherals can be interconnected at any time and all the Event Channels operate in parallel.

Independent of the CPU and DMA, the response time for the Event System will never be more than 2 clock cycles of the IO clock and it does not generate software overhead for the CPU during operation.

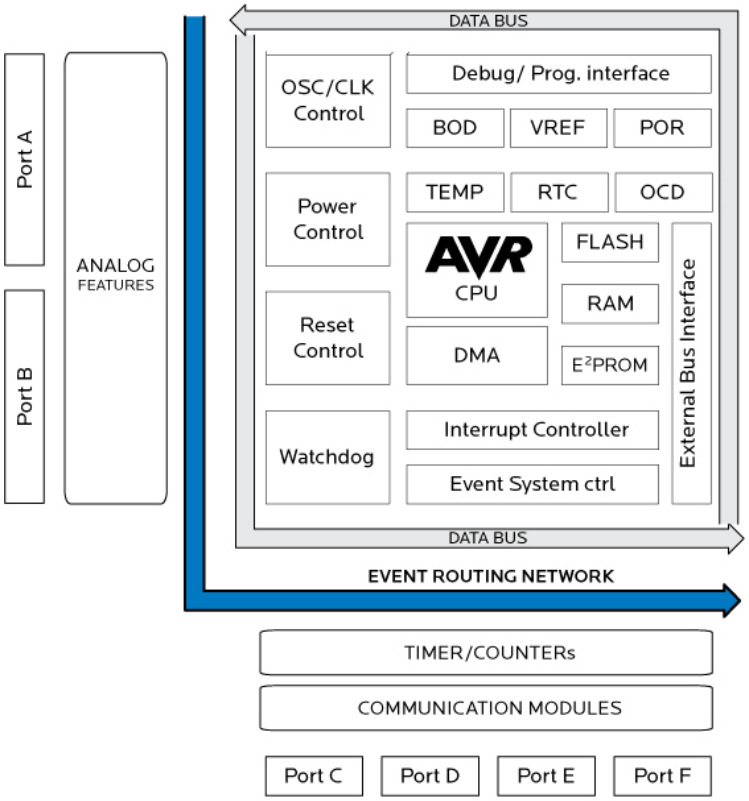


Figure 2. Event Routing Network

The Event System removes the bottlenecks associated with multiple and/or frequently triggering interrupts. There is no software overhead and critical tasks can be performed independent of the CPU, and with a guaranteed latency shorter than any interrupt response time in the industry today.

Low power with second generation picoPower

The Power consumption of an embedded application can be expressed as

$$P = P_{static} \cdot t_{sleep} + P_{dynamic} \cdot t_{active}$$

The entire power consumption of a microcontroller in sleep mode is leakage current. The leakage current is solely determined by the manufacturing process. So for microcontroller companies, reducing static power consumption is all about choosing the right process. Atmel uses its proprietary embedded Flash processes to manufacture microcontrollers with picoPower technology. With XMEGA Atmel introduces the 2nd generation picoPower technology, further extending their lead in the low power microcontroller industry.

For dynamic power consumption there are a few more factors the microcontroller vendor can adjust to reduce power consumption.

$$P_{dynamic} = K \cdot V_{CC}^{2} \cdot f$$

Most important is lowering VCC. All Atmel picoPower products including AVR XMEGA offer true 1.6V operation. True 1.6V operation means that all analog modules, Flash, EEPROM and RAM operates at 1.6V. Other 8/16-bit MCU products either has a higher minimum VCC or does not support all functions down to 1.6V. With picoPower you can even use the ADC, reprogram Flash, write to EEPROM, and use internal oscillators the way down to 1.6V.

Lower the operating frequency is important to reduce dynamic power consumption, but not recommended for a picoPower application. If the frequency is lowered, the application spends more time in active mode and less in the battery saving static mode. This increases average power consumption and decreases battery life.

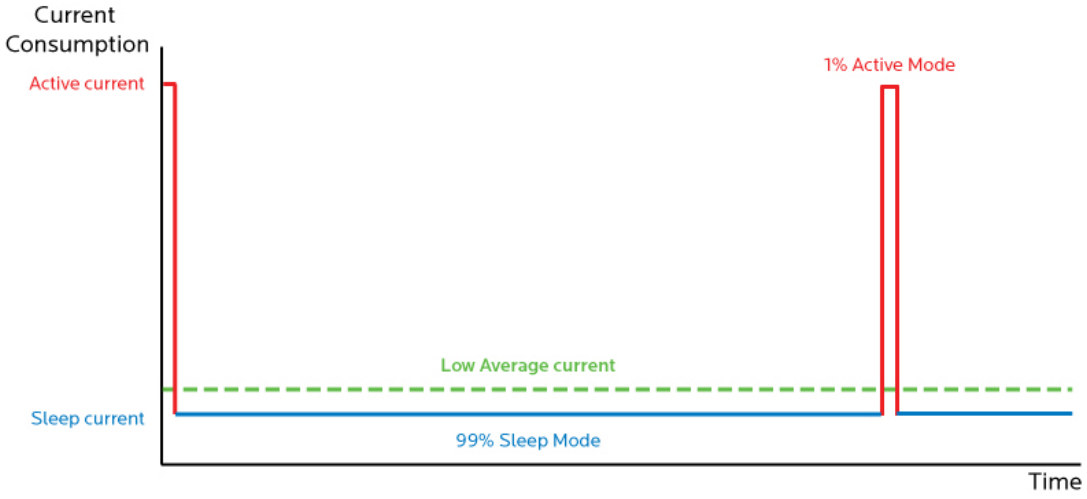


Figure 3. Run-time for picoPower devices.

In most applications, the processor does not need to run continuously and peripherals may be idle for longer periods. When active it is often crucial to have a very high throughput. The XMEGA family takes advantage of this by combining 32 MIPS with very flexible sleep modes and a new powerful DMA and event system. This allows the XMEGA devices to spend more time in sleep mode than competitive microcontrollers, significantly reducing overall power consumption in an application.

XMEGA delivers industry leading power consumption numbers. In RAM retention mode, current consumption is typically 100 nA. Any pin change or I2C communication can bring the picoPower device into Active mode within a few microseconds. If a Real Time Clock function is needed, the 32 kHz crystal oscillator connected to a timer can be enabled with a power consumption of 650 nA. Texas Instruments has similar current numbers for their MSP430 devices, but require 2.2V operation for all functions in the microcontroller to operate. Operating at 2.2V instead of 1.6V gives an instant 37.5% penalty in energy consumption.

XMEGA is able to deliver up to 12 MIPS at 1.6V, with an active mode power consumption of only 3.6 mA. Texas Instruments MSP430 devices have similar current consumption, but require VCC at 2.7V to deliver 12 MIPS. At this voltage the 8/16-bit XMEGA can run at 32 MHz and deliver up to 32 MIPS.

Table 2. *Typical current consumption multiplied with required VCC*

	Flash [KB]	Active 1 MIPS [mW]	Active 12 MIPS [mW]	RTC mode [uW]	RAM retention [nW]
ATmega48P	4	0.5	17.1	0.90	180
ATmega644P	64	0.7	18.6	1.08	540
MSP430F2121	4	0.4	8.6	1.54	220
MSP430F2491	60	0.6	10.3	1.98	220
MSP430F2419	120	0.8	14.9	2.20	440
ATxmega128A1	128	0.6	6.3	1.17	180

To ensure safe operation it is always recommended to have a brown-out detector (BOD) enabled. This has to be either internal in the microcontroller or an external device. The brown-out detector will ensure that the microcontroller is held in reset if VCC is lower than minimum. The XMEGA BOD has a new and innovative sampling mode. In sampling mode, the BOD is turned on only once per ms to check status of VCC. If everything is OK, the BOD is turned off and remains off until next checkpoint. This gives XMEGA a max power consumption of 2 µA in Power Save mode with a 32 kHz crystal running, Real Time Counter operating, Watchdog Timer enabled and BOD operative with max 1 ms reaction time. To achieve the same performance, TI MSP430 devices need to enable the System Voltage Supervisor that consumes up to 15 µA.

High-end analog modules

XMEGA microcontrollers have leading analog integration and offers high analog accuracy features with high-speed analog peripherals. The XMEGA ADC has 12-bit resolution and is capable of up to 2 million samples per second. It delivers the fastest sample rate and most accurate result for conventional microcontrollers. Increasing ADC resolution from 10 to 12 bits gives four times better resolution. For medical applications such as glucose meter this means that glucose levels can be measured four times more accurate and inserted insulin can be dozed four times more accurate.

A high analog sample rate is important for many embedded applications. This enables high frequency signals to be measured more accurately. For applications such as scanners and motor control high sample rate and accurate timing is needed in order to measure signals at the right time and at a high frequency. A second advantage of high sample rate is the ability to over sample to increase resolution at no cost. In general n bit of extra resolution is achieved with 2^n extra samples of the same signal. Thus, the 2 Msps on the XMEGA ADC makes it possible to measure a 125 kHz signal with 16-bit resolution. Compared to competition this enables superior resolution at a speed and cost that 16-bit ADCs cannot offer.

Table 3. ADC Resolution Comparison

Part	Sample Rate	Resolution
XMEGA A	2000 ksps	12-bit
STM32F101	1000 ksps	12-bit
PIC24	500 ksps	12-bit
MSP430	300 ksps	12-bit
NEC78K0R	165 ksps	10-bit

XMEGA also features digital-to-analog converters (DAC) with up to 1 Msps conversion rate and four 12-bit channels. For all applications that require analog output for instrument calibration, actuator input, sound and speak interface a DAC offers both higher resolution and lower cost than using a PWM with external filter for analog signaling. Removing external components is important for applications with strict size constraints such as optical transceivers.

Table 4. DAC Resolution Comparison

Part	Sample Rate	Resolution
XMEGA A	1000 ksps	12-bit
STM32F101	N/A	
PIC24	N/A	
MSP430	35 ksps	12-bit
NEC78K0R	N/A	

Up to four analog comparators are available, and both a wide selection of I/O pins, DAC output and scaled VCC voltage can be used as input for the comparators. A comparator pair can be used in window mode to check if an input signal is inside, above or below a voltage window. The state of the comparator can be polled by software, used to trigger events or interrupts, or output to a pin directly,

Huge communication bandwidth

In embedded applications like board controllers, vending machines, alarm systems, terminals and so on the need of several communication interfaces on one microcontroller is

huge. To solve this XMEGA is equipped with up to eight USARTs, four Serial Peripheral Interfaces (SPIs) and four I2C and SMBus compatible Two Wire Interfaces (TWIs). This gives both flexibility for layout and the right communication support for embedded applications that need a mix of the following:

- SPI for display data
- TWI for capacitive touch sensors
- TWI for board bus
- SPI for serial EEPROM or Flash memories
- SPI for camera interface
- USART for debug port
- USART for user interface
- USART, I2C and SPI for sensor and actuator interface
- SMBus for smart battery charger or host
- SPI for RF interfaces
- SPI for Bluetooth interface
- USART for smart card reader interface

With the help from the DMA Controller, several communication interfaces can be implemented with minimal CPU usage.

Flexible Timer/Counters

XMEGA has up to eight 16-bit Timer/Counters with time keeping, frequency waveform, Pulse Width Modulation (PWM) and input capture capabilities. Compared to competitor solutions with fewer Timer/Counters, XMEGA makes it easier to have several time bases in the application. Timer/Counter are the most used and most flexible peripherals in a microcontroller. Independent time bases are needed to support the following tasks in embedded applications:

- Time keeping
- PWM generation
- Frequency generation
- Input capture
- Frequency measurements
- Pulse-width measurements
- Duty cycle measurements
- Operating System clock tick
- ADC Time Triggers
- DAC Time Triggers
- Bit-banged communication
- DMA Triggers
- Clock and calendar

To provide additional functionality the Timer/Counters in XMEGA has extensions that enable more specific functionality. A The High-Resolution Extension enables PWM with up to 128 MHz timer tick. High resolution gives smaller step size, and thus improved accuracy for PWM outputs. The Timer/Counters in XMEGA can offer 31.25 kHz PWM with 12-bit resolution and 250 kHz PWM with 9-bit resolution. In a buck converter, high speed PWM reduces cost and size of external filter.

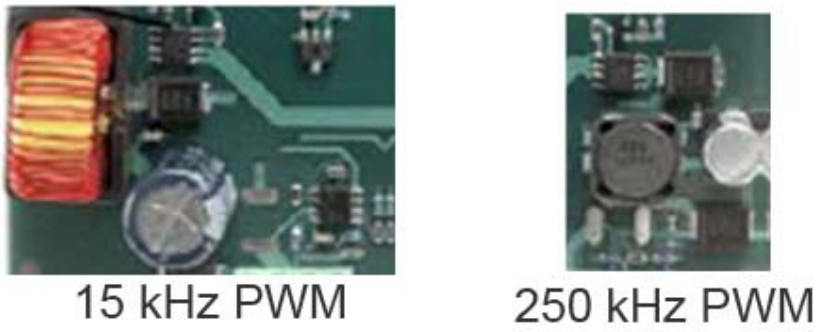


Figure 4. Comparison of capacitor and coil size for a buck converter with low and high speed PWM.

Fast AES and DES Crypto engine

XMEGA has a hardware crypto engine with support for Advanced Encryption Standard (AES) and Data Encryption Standard (DES). These supports the growing need for encrypted communication and secure boot loaders. Compared to software solutions, the crypto engine can speed up AES encryption/decryption from 4000 to 375 clock cycles for a 16-byte block. For DES encryption/decryption the improvement is from 100,000 to only 16 clock cycles for a block of 8 bytes. Where almost all other available microcontrollers can only support very slow encrypted communication, XMEGA can support high speed encrypted communication, and solve the problem with adding secure encryption for wireless low power applications. Using AES maximum bandwidth on encrypted UART communication is more than 10 Mbps.

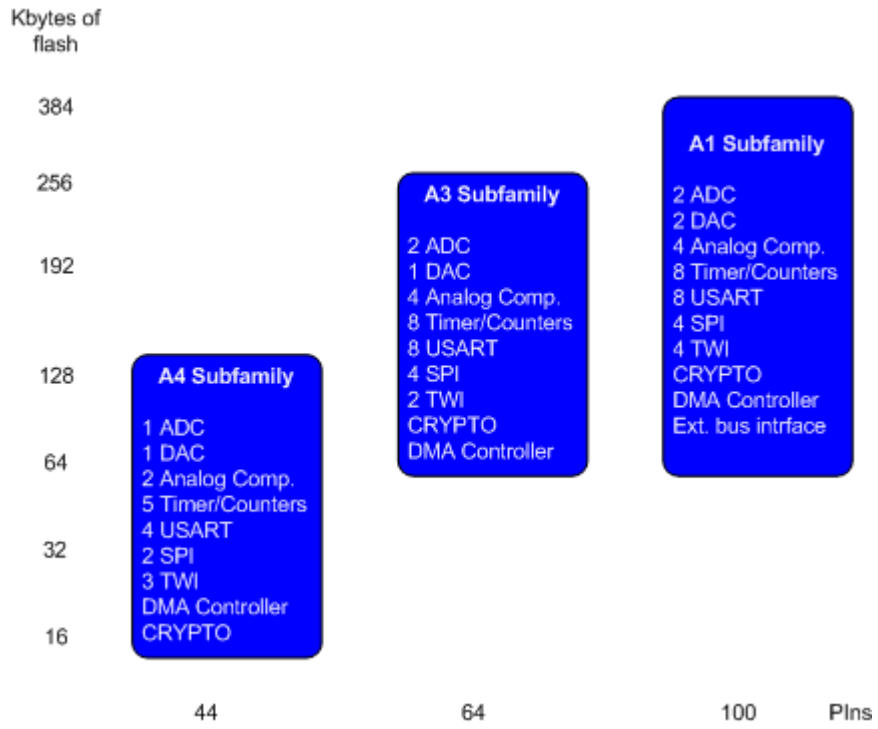
Development tools

Using AVR Flash microcontrollers reduce time to market. All AVR devices have an on-chip debug system and are in practice its own emulator. This ensures 100% real behavior and electrical characteristics during development and debugging. The on-chip debug system is fully non-intrusive, leaving all resources untouched and available for application code while debugging. Atmel offers its on-chip debug system for all AVR devices ranging from 8-pin 1K tinyAVR® devices to the 256K 100-pin megaAVR® and XMEGA devices.

Atmel offers the free AVR Studio® to start writing and debugging code instantly. One tool chain controlled by AVR Studio supports all AVR devices with Starter Kits, Evaluation Kits and Reference Designs. Atmel provides the GNU gcc C compiler and GNU gdb debugger free of charge. Commercial licenses from IAR® (Embedded Workbench) are also available. Atmel's AVR Studio, STK600 and JTAGICE mkII provide the XMEGA with a multiplatform integrated development environment (IDE) already configured for the GNU tool chain. A comprehensive collection of application notes also kick-start troublesome tasks.

Conclusion

By introducing AVR XMEGA, Atmel pushes the boundaries of the 8/16-bit MCU world. It delivers an excellent combination of high-end peripherals, good performance and ultra low power consumption.



About Atmel Corporation

Atmel is a worldwide leader in the design and manufacture of microcontrollers, advanced logic, mixed-signal, nonvolatile memory and radio frequency (RF) components. Leveraging one of the industry's broadest intellectual property (IP) technology portfolios, Atmel is able to provide the electronics industry with complete system solutions focused on consumer, industrial, security, communications, computing and automotive markets.

Further information can be obtained from Atmel's Web site at www.atmel.com/avr.

Contact:

Kristian Saether, e-mail: kristian.saether@atmel.com
Ingar Fredriksen, e-mail: ingar.fredriksen@atmel.com