

# On-Line JOURNAL

## AVR32 System-on-Chip for Hand-held Compute-intensive Embedded Systems

**By:** Oyvind Strom, PhD., Atmel Corporation

### Useful Links in this Article:

#### AVR32 EEMBC 32-bit MCU/DSP Benchmarks

<http://www.atmel.com/products/avr32/overview/overview3.asp>

#### Software, Datasheets, and Tools, including:

AVR32 UC3 Software Library

AVR32 GNU Toolchain - Windows

AVR32 GNU Toolchain - Linux Fedora 5

AVR32 GNU Toolchain - Linux Fedora 6

AVR32 GNU Toolchain - Linux Ubuntu Software

STK1000: a complete AT32AP7000 IDE

ATNGW100 Network Gateway Kit

EVK1100 evaluation kit

AVR32 Studio: free IDE

AVR32 GNU Toolchain

[http://www.atmel.com/dyn/general/advanced\\_search\\_results.asp?device=1&tools=1&faqs=1&datasheets=1&appNotes=1&userGuides=1&software=1&press=1&articles=1&flyers=1&checkAll=1&checkAllReference=1&target=AVR32](http://www.atmel.com/dyn/general/advanced_search_results.asp?device=1&tools=1&faqs=1&datasheets=1&appNotes=1&userGuides=1&software=1&press=1&articles=1&flyers=1&checkAll=1&checkAllReference=1&target=AVR32)

#### Search, Compare and Buy Processors:

<http://www.embedded-developer.com/corp/191/Atmel.htm>

### INTRODUCTION TO SYSTEM CHALLENGES

By 2009, there could be over 950 million MP3 players, media players such as iPod®s and other multimedia devices shipped yearly. Shipments of these hand-held infotainment systems are expected to grow by 211% between 2004 and 2009<sup>1</sup>. As the number of features in portable devices increases, the amount of

processing increases putting additional strain on battery life. Feature sets are growing like wildfire with consumers demanding the addition of cameras, video players, MP3 players, GPS functionality, Bluetooth® and even 802.11 to their phones, PDAs and infotainment in automotive. However, the capacity of batteries are not keeping up, leaving users with reduced playback or active time.

### THE PERFORMANCE/POWER CONSUMPTION DILEMMA

For example, the processors, LCD, RF and earphones in a typical hand-held media player consume close to 4 watts (Table 1) while the batteries typically used in these applications are 3.7 V and rated at 900-2000 mAh. This situation leaves the end-user with no choice but to constantly recharge the battery to be able to playback several hours of multimedia content.

Hand-held media player components	Power Consumption (Watts)
CPU	0.41
LCD panel	1.06
Bluetooth	N/A
CDMA	N/A
802.11b	0.25
SDRAM	0.56
Flash memory	0.03
Off-chip buses	0.62
DSP	0.86
<b>Total</b>	<b>3.79</b>

Table 1: Typical power consumption figures (Watts)<sup>2</sup>



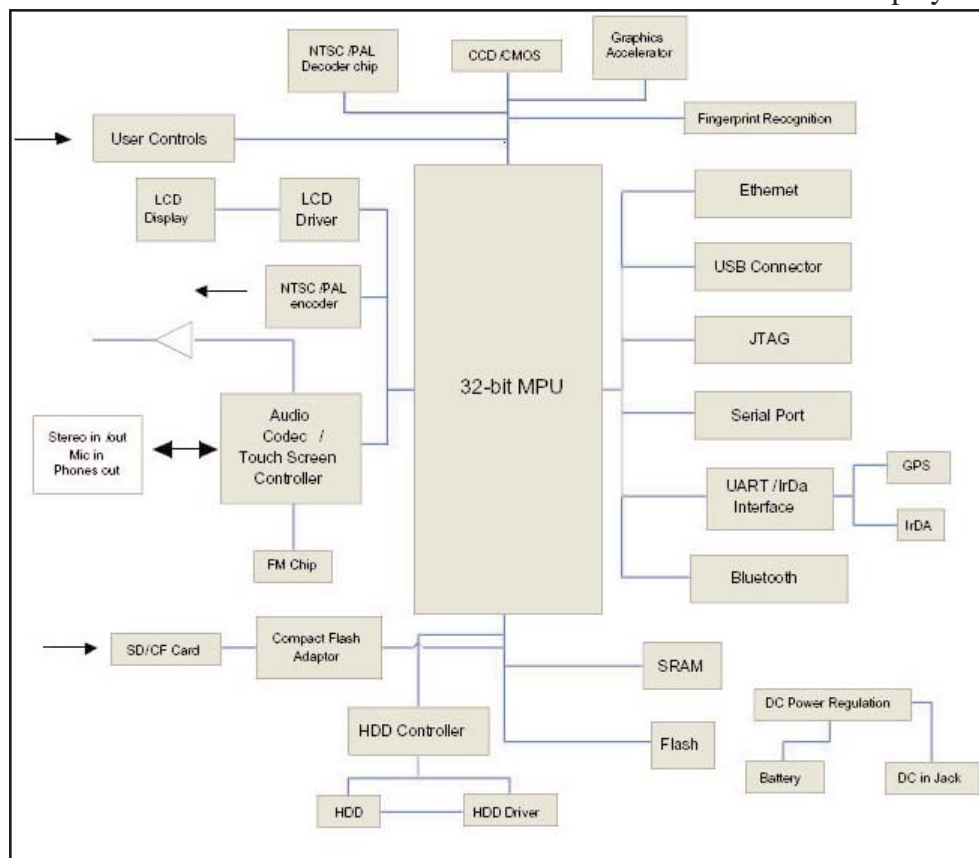
1. <http://www.redherring.com/Article.aspx?a=13988&hed=MP3+Player+Boom+Forecast&sector=Industries&subsector=Entertainment+And+Media>

2. Source: Research paper by Shim, Cho, Chang at the university of Korea (ESTIMedia2004) and Atmel.

The inevitable problem is that increasing the amount of processing usually requires a faster clock rate which will also increase the power consumption. If the power drain increases faster than the capacity of the power source, the battery life will become so unacceptably low or the batteries will become so unacceptably large that products will not succeed in the market. Increase computing efficiency and system integration.

The bottom line is that IC vendors and system integrators are going to have to find ways to add features while reducing power consumption. CPU cores that process more per clock cycle will help because they will allow designers to use a slower clock. DMA schemes that off-load data transfer tasks from the CPU help to keep the clock frequency low. In addition, smart processor architectures that dynamically minimize the bus clock frequency will also reduce power consumption. Finally, higher levels of integration into single-chip solutions will help keep the power consumption low by reducing system capacitance.

Hand-held multimedia systems typically require three ASSPs/MCUs and a DSP to perform board control tasks, decode the multimedia content and interface with the end-user. Since the DSPs cannot handle control tasks and the microcontroller is unable to cope with the ever increasing computational requirements, many products use both a TI C55x-based DSP and an ARM9™/ARM11™-based MCU on the same printed circuit board. Additional ICs are needed for baseband decoding, hard drive interface and control, MPEG- 2/MPEG-4 video decoding, Ethernet interface, Bluetooth protocol handling, LCD driver, high-speed USB interface, CCD camera chips and removable memory cards (CF, SD/MMC cards). In addition to consuming over 25% of the power, these multiple processors significantly increase the design complexity because each processor will require its own compiler, debugger and development environment. The designer must select one or more operating systems for each individual MCU as well as select communication protocols between the MCUs and DSPs that will result in the accurate decoding and output of video and audio to the speakers and displays.



With the expanding feature set, the number of chips are expected to increase putting added complexity on the software stacks for the inter-IC communication protocols, highspeed buses (USB, Ethernet) and numerous of other communication protocol stacks that are required in the application. One solution to improve battery life and lower system complexity is to integrate as many of these ICs as possible on a single piece of silicon. This solution should include a high throughput CPU that can execute both system control functions and compute-intensive DSP algorithms without increasing the clock frequency to keep chip count, design complexity and power consumption as low as possible.

Fig 1. Multimedia Player Typical System (Source: Semico Research Corp.)

## A NEW GENERATION OF PROCESSORS

One example of such a processor is Atmel's AP7000 which integrates the AVR<sup>®</sup>32 MCU/ DSP core with a vectored multiplier co-processor, on-chip SRAM, CPU caches, memory management unit, and multiple on-chip DMA controllers. Peripherals include on-chip audio DAC, TFT/STN LCD and HDD controllers, and 480 Mbps USB 2.0 with on-chip transceivers (PHY). Two 10/100 Ethernet MACs adds to the connectivity of the application while serial interfaces like RS232, USART, I2S, AC97, TWI/I2C, SPI, PS/2 and several synchronous serial modules (SSC), supporting most serial communication protocols, help replace the majority of external components in these systems.

The bus structure allows clocks to be set to individual frequencies in four domains (CPU, bus matrix, peripheral bridge A and peripheral bridge B) to minimize power consumption at all times. This high level of integration allows the deployment of software libraries and application code on a single platform and provides better control of system integration, testing and time-to-market. Moving all of the external functionality of a media player design into the AVR32 AP7000 chip enables both software and hardware engineers to focus their power saving strategies and techniques on one chip.

### AVR32 CPU

Most 32-bit architectures were developed in the 1970s and 1980s, long before USB, Ethernet, iPods or MP3 players. These CPUs are excellent in high MIPS applications. However, their instruction set architectures may not be optimal for applications that have evolved over the last decade that rely on a battery or signal line (USB) for their power supply. Atmel's AVR32 MCU/DSP core has been built, from the ground up, specifically to address the need for high computational throughput and low power consumption. The processor puts special emphasis on 1) maximizing the use of computational resources with a seven-stage pipeline and three parallel sub-pipelines that supports automatic data forwarding and out-of-order execution; 2) using pointer arithmetic to minimize the cycle count for load store operations, 3) accurate branch prediction with zero-penalty branches, and 4) maximizing code density to reduce cache misses.

## SEVEN-STAGE PIPELINE WITH OUT-OF-ORDER EXECUTION & AUTOMATIC DATA FORWARDING

CPUs with linear pipelines allow faster clock frequencies. However, in power-constrained applications it may be preferable to keep the clock low to reduce power consumption and increase the throughput per cycle with parallel sub-pipelines. For example, Atmel's AVR32 has a seven-stage pipeline, with three parallel ALUs, and multiply and load/store sub-pipelines that support out-of-order execution and automatic data forwarding. By keeping data close to the core, automatic data forwarding eliminates cycles wasted on reading and writing intermediate results to and from power-hungry register files. Parallel multiply, ALU and load/store sub-pipes, with hazard detection circuitry, allow out-of-order execution of instructions, thereby maximizing the utilization of computational resources. The AVR32 multiplier has a patented accumulator cache that allows single-cycle multiply/accumulate (MAC) operations with a minimum number of read/write ports. Less silicon and less power consumption.

## POINTER ARITHMETIC MINIMIZES LOAD/STORE CYCLES

On average, 30% of a processor's cycles are spent, not on operations, but on load/store instructions. The AVR32 core reduces the required number of load/store instructions with byte (8-bit), half-word (16-bit), word (32-bit) and double word (64-bit) load/store instructions that are combined with various pointer arithmetic to efficiently access tables, data structures and random data in the fewest number of cycles. The AVR32 core has 28 instructions that increase the efficiency of load/store operations.

## BRANCH PREDICTION AND FOLDING CAN ACHIEVE ZERO-CYCLE PENALTY IN LOOPS

Although deep pipelines enable higher clock frequencies, they introduce significant cycle penalties whenever there are jumps in the program flow. Branch prediction logic in the AVR32 pipeline can accurately predict all change-of-flow instructions and "fold" the branches with the target instruction, resulting in a zero-cycle branch penalty.

## ALGORITHM & COMPILER BENCHMARKING FOR ISA DEVELOPMENT

Many people think that code density is irrelevant because memory prices have become so low. However, code density affects performance because a cache miss can cost thousands of cycles and caches have not increased in size. They are still 8- 16- or 32-Kbytes.

The AVR32 instruction set was developed with an iterative process, by benchmarking algorithms and real-life applications and iteratively increasing compiler performance. The instruction set includes single-cycle SAD, FFT, iDCT, vector multiplication, Viterbi butterfly, block cipher algorithms, parallel and array arithmetic operations, as well as Java® and RISC instructions. Single-instruction-multiple-data (SIMD) instructions include single-cycle vectored 8- and 16-bit parallel addition and subtraction with optional saturation or halving, averaging, max/min, absolute value, shift and add/subtract operations.

Successive compiler benchmarking and refinement have resulted in an instruction set architecture (ISA) and compilers that generate binary code with 50% less memory requirements than that of comparable 32-bit cores. Smaller code means more instructions can be stored in the processor's instruction cache, thereby substantially reducing the number of cache misses and increasing overall processor throughput per cycle. The higher code density also helps to save power because fetching code from SRAM or external memory consumes much more power than fetching it from the processor cache. Denser code also improves power consumption.

## VECTOR MULTIPLICATION CO-PROCESSOR

Multimedia applications commonly require arithmetic operations on 3x8-bit matrices for image filtering (FIR filtering), image color-space conversion (RBG<->YUV), image scaling, and MPEG-4/H.264 quarter-pixel motion estimation. In order to keep CPU throughput as high as possible, the AP7000 has integrated the vector multiplication unit (VMU) with the AVR32 AP co-processor interface, tightly coupled to the CPU.

For example, MPEG-4 video is compressed using the YUV color space, while most video DACs and LCD controllers require RGB input. The vector multiplier co-processor executes this conversion in real-time without CPU intervention. When used for image scaling operations, the AP7000's vector multiplication co-processor increases the performance by as much as ten times.

## DIRECT MEMORY ACCESS (DMA)

Multimedia applications are data-intensive, with streams of data passing between the peripherals and the memories. Managing these data transfers can seriously compromise the CPU. Without DMA, the 100 Mbps Ethernet MACs and the 480 Mbps USB slave controller would saturate the CPU with data movement operations, rendering it useless for processing the data. The AP7000 provides several flexible direct memory access (DMA) mechanisms that off-load, from the CPU, data transfers between peripherals and memory and between two memory locations in the chip. Two simultaneous memory-to-memory data transfers can be performed between on-chip SRAM, or off-chip memories connected to the chip's external bus interface (EBI), as they are all addressable to the DMA controller. An 18-channel peripheral DMA controller (PDC) provides a tight integration of peripherals and memory and is directly mapped into the programming memory of each peripheral. The PDC is used to setup virtual channels between data registers in the peripherals and the chip's on-chip or off-chip memories, completely off-loading these tasks from the CPU. In total, the AP7000 has 18 peripheral-to-memory channels and two simultaneous memory-to-memory interfaces on-chip.

## WHEN SLOWER IS REALLY FASTER: REDUCED CLOCK FREQUENCY

Multimedia algorithms must be executed on the specified number of pixels at the required frame rate. The processors must execute every operation on every pixel each second, but there is no value in doing any more processing per second than required. As a result, a processor with a higher throughput per cycle can actually run slower than other processors with equal performance.

Since the AVR32 core can execute two to three times as many operations per clock cycle, it can execute the application using a much slower clock frequency. At just 100 MHz, it can execute quarter-VGA MPEG-4 decoding. Competing processors need clocks of 150 MHz to 266 MHz for the same function. This means that the processor will consume between 1/3 and 2/3 less power. With processing taking up as much as 45% of system current consumption, system battery life could be extended by 15% to 27%. New features could be added without sacrificing battery life.

### DYNAMIC FREQUENCY SCALING

The AP7000 architecture has a multi-layer, high-speed bus architecture that increases performance by allowing multiple operations to take place in parallel. Peripherals require different bus clocks. In a conventional bus structure, a peripheral that requires a fast clock (e.g., the bus matrix where all peripherals communicate across) will introduce a power consumption penalty by requiring all busses to operate at the same high clock rate. The AP7000 solves this problem by providing two peripheral bus bridges, one for high-speed peripherals and one for low-speed peripherals. The AP7000 can dynamically configure the individual clock frequencies of these two bridges, as well as the frequencies of the CPU's internal clock and that of the bus matrix, based on dynamic frequency scaling algorithms.

The clock frequency of the four domains is not required to be the same and can be set independently of each other in a single CPU cycle. For example, the CPU clock frequency can be decreased when the application is inactive, while maintaining a

constant (and low) clock frequency for the peripheral bridges during a data transfer (such as via Bluetooth, IrDA<sup>®</sup>, UART or AC97). To change the clock frequencies, a 32-bit value is written to a configuration register and the change is instantly reflected for all clock domains. Internal low-power oscillators and PLLs are running and locked and do not require any time for calibration between the frequency changes, thus making all changes instantaneously.

### POWER SAVING MODES

The AP7000 has a System Manager that provides a number of power saving modes. These include normal, idle, frozen, standby, stop, and static modes. The software programmer may define states in the application that employ these power modes to further reduce power consumption.

### HIGH-LEVEL OF SYSTEM INTEGRATION

The capacitive load created by interconnecting wires between chips on a printed circuit board can contribute significantly to power consumption. The best way to avoid this is to integrate as many chips as possible on the same piece of silicon. Atmel has taken 2/3 of the chips required in a multimedia system (Figure 1) and integrated them on the AP7000 to create a true system-on-chip (SoC) with virtually all the peripherals and interfaces required for target applications.

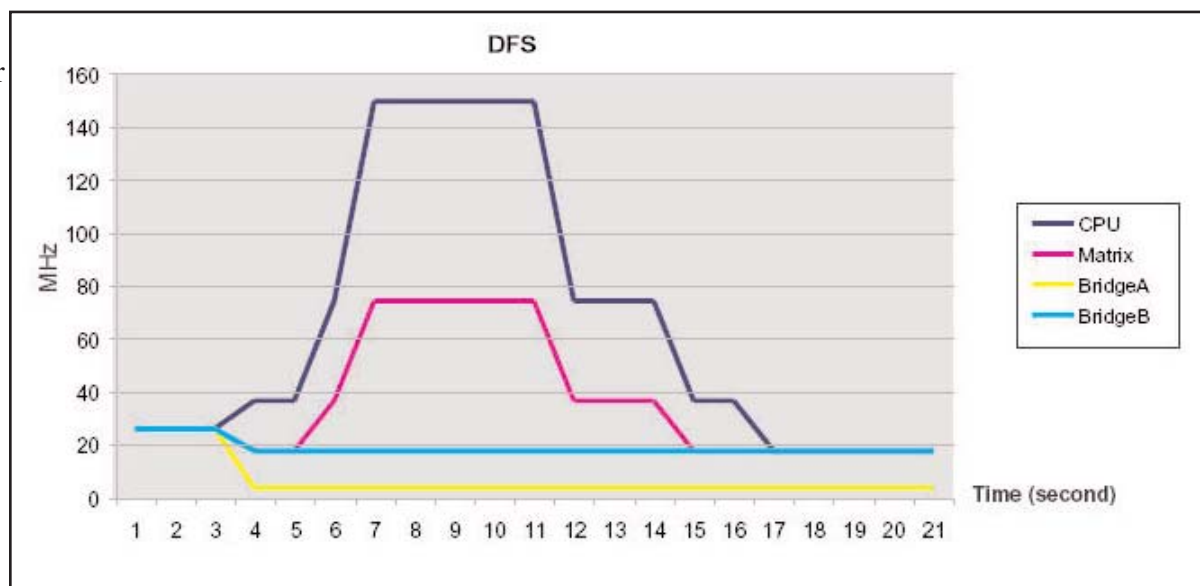


Figure 2: Clock Domains Dynamic Frequency Scaling

## SINGLE-PROCESSOR DEVELOPMENT ENVIRONMENT

The AVR32 core is a RISC MCU with DSP extensions which allows the software to be developed and debugged in a single, MCU-style development environment. The AVR32 Instruction Set Architecture (ISA) is specifically designed for high-level programming languages like C, C++ and Java. Compilers are available from Atmel (GNU GCC) and IAR Systems (IAR Embedded Workbench®). IAR Systems provide SIMD- and DSP-aware functionality in their compiler, which automatically compiles the appropriate SIMD DSP instructions based on C-patterns in the code. Both compilers support in-line assembly for tight-loop/inner-loop algorithmic optimizations from within the C/C++ development environment.

Atmel's existing JTAG-based debugger, the JTAG-ICE mkII, in addition to the Ashling Vitra high-end debugger, supports runtime control in addition to on-chip trace functionality through the On-Chip Debug (OCD) System of the AP7000. GCC and GNU Debugger (GDB) are available directly from Atmel and integrates directly into many integrated development environments (IDE), including the Eclipse™ debug environment.

The AP7000 has a fully-supported Linux® 2.6 kernel to further ease the transition of existing code or the adoption of the many hundreds of thousands of open source and freely-available applications available for use in embedded systems. Linux is also available from Atmel.

## CONCLUSION

The AP7000 eliminates the need for a multi-chip DSP/MCU development environment, thus unifying the software engineer's development environment and reducing complexity and board space for the hardware engineer.

By using the AP7000 MCU/DSP, designers can remove the DSP for application processing and achieve, in total, a 30% reduction of power consumption in the application. The AP7000 runs at a lower clock frequency as well as decoding an MPEG-4 video-stream and running a full Linux operating system on the same chip.

The power consumption is reduced further by removing off-chip buses from the application since the number of chips on the board is reduced. The 30% reduced power consumption increases battery life and allows the end-user longer operating time. This also allows system designers to add more functionality without reducing the battery life compared to existing solutions.

Hand-held media player components	Regular Power Consumption (Watts)	AP7000 Power Consumption (Watts)
CPU (MPEG-4 decoding and Linux)	0.41	0.25
LCD panel	1.06	1.06
Bluetooth	N/A	N/A
CDMA	N/A	N/A
802.11b	0.25	0.25
SDRAM	0.56	0.56
Flash memory	0.03	0.03
Off-chip buses	0.62	0.50
DSP	0.86	N/A
<b>Total</b>	<b>3.79</b>	<b>2.65</b>

Table 2: Typical Power Consumption Figures of AP7000 and a Traditional Player.