The Linley Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

SYNOPSYS BUILDS A BETTER ARC

ARC EM Cores Improve Power, Area, and Code Density

By J. Scott Gardner (October 1, 2012)

After acquiring Virage Logic in 2011, Synopsys has been quietly building its portfolio of ARC-based CPUs to offer SoC designers more options for embedding 32-bit controllers. With little fanfare, the company just released a broad range of enhancements to its ARC EM family, improving CPU performance and code density while giving designers more flexibility to tune the CPU architecture for better efficiency in embedded applications. Already tiny and power efficient, the newest ARC EM CPUs add performance-enhancing features that raise the bar for what SoC designers expect from CPUs in the low-end category defined by ARM's Cortex-M family.

Synopsys is well known as an EDA company, with its DesignWare library of licensable intellectual property (IP) that includes interface IP, analog IP, embedded memories, and logic libraries. Whereas most chip designers know that DesignWare also includes licensable CPU cores and subsystems, many are surprised to learn that Synopsys is second only to ARM in the number of chips that use its CPU cores.

Because Synopsys works closely with other CPU-IP vendors such as ARM, MIPS, and Tensilica, it doesn't strongly promote its own products as direct replacements for CPUs from these other vendors. Instead, it embeds its ARC CPUs into IP subsystems (see *MPR 5/7/12,* "Catching a SoundWave With ARC") while also supporting its well-established customer base that uses deeply embedded CPU cores for high-volume, low-power applications such as memory cards, sensors, and SSD controllers.

As our recent report covering the CPU-IP market notes, The Linley Group estimates that 993 million chips containing Synopsys CPUs shipped during 2011, surpassing both MIPS and Tensilica by more than 50%. Many of these chips incorporate multiple ARC cores, and we expect Synopsys to maintain this market position as it continues to broaden its customer base. The ARC EM family, first announced in October 2011, will drive most of the future CPU volume for the company.

Embedded Siblings: ARC EM4 and EM6

The ARC EM family is based on a 32-bit RISC microarchitecture, shown in Figure 1, that uses the ARCv2 instruction set (see *MPR 1/9/10,* "ARC 601 Gets Small"). These CPUs have an efficient three-stage pipeline to minimize power

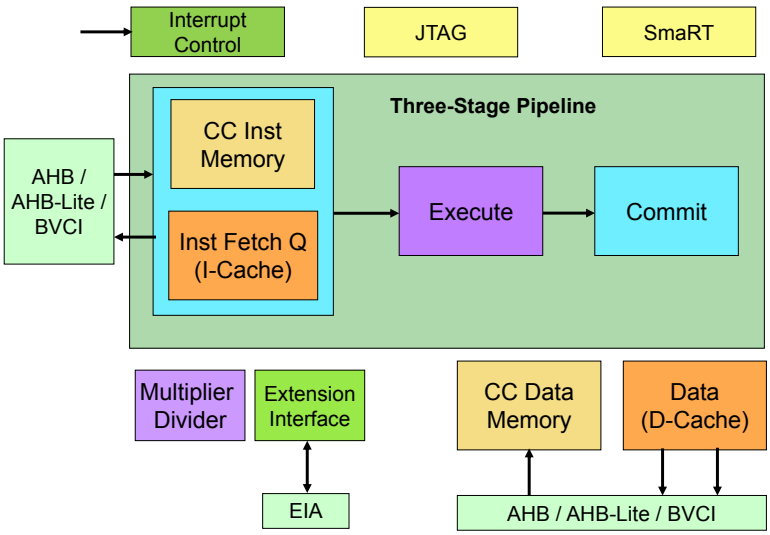

Figure 1. Block diagram of DesignWare ARC EM family. This CPU family uses a three-stage RISC pipeline with a 32-bit Harvard architecture and closely coupled memories. The ARC EM6 adds support for configurable level-one instruction and data caches.

consumption and die area. Consuming only 4.4 microwatts per megahertz and requiring just over 0.01mm^2 when power-optimized (synthesized for 100MHz) in a 40nm LP process, the ARC EM4 is the smallest CPU core that Synopsys offers. Even with a three-stage pipeline, chip designers can synthesize an EM4 CPU to reach 885MHz in 40nm GP while still fitting into a modest power and area budget (9.6 microwatts per megahertz and 0.04mm^2).

The tiny EM4 targets deeply embedded applications that perform an often-fixed function deep in the internals of an embedded system. As an example, solid-state memory devices require deeply embedded CPUs to control data blocks in order to implement algorithms such as wear leveling and error handling. These CPUs are essentially 32-bit microcontrollers that operate well below the maximum silicon speed while sipping power and consuming as little die area as possible. Deeply embedded applications do not benefit from a standard instruction set such as ARM or x86 because they typically do not run third-party software; instead, they run a small amount of internally developed code that can be compiled for any instruction set. Code size and power efficiency are the most important metrics in these designs.

The ARC EM6 CPU core uses the same three-stage microarchitecture as the EM4 and can be similarly configured with a simple CPU interface to closely coupled instruction and data memories (ranging in size from 512 bytes to 1MB). The EM6 adds support for level-one instruction and data caches, which the system designer can independently size from 2KB to 32KB. Designers can also configure the cache line size and set associativity to tune the hit rate for specific software workloads. Cache lines can be locked to ensure deterministic performance for real-time tasks. Both the EM4 and EM6 natively support AHB, AHB-Lite, and Basic Virtual Channel Interface (BVCI) bus protocols.

Building an Efficient CPU

The EM4 and EM6 benefit from their ARC heritage as configurable and extendable processors. In addition to configurability of the memory subsystem and SoC bus interface, designers can configure the CPU core using the Synopsys ARChitect tool to optimize performance for the expected software workload.

For example, the 32-bit register file (which can be built with either flip-flops or memory cells) is configurable for 16 to 32 registers. Using extension core registers, the CPU supports up to 60 total registers. Designers can also configure the register file with a second write port (the EM family uses two read ports).

Other configurable features include the program-counter and address-bus widths as well as timers, interrupts, multipliers, divide hardware, a loop counter, and big- or little-endian memory addressing. These CPU features can be configured (or excluded, in some cases) to minimize the number of transistors needed to efficiently perform an embedded software task.

In addition to configurability, the ARC EM family supports instruction-set architecture (ISA) extensions. Designers can add the ARC FPX unit, which complies with the IEEE 754 standard and supports both single- and double-precision data types. The ARC FPX allows the EM family to deliver approximately one megaflops per megahertz of floating-point performance. The ARC EM instruction set can also be extended to support new customer-defined instructions, which are decoded and executed in much the same way that floating-point instructions extend the ISA.

Instructions added using the Synopsys ARChitect design tool become part of the Synopsys co-design environment, which includes the nSIM instruction-accurate simulator (averaging 475 mips in turbo mode) as well as a cycle-accurate simulation using the xCAM tool (40–60KHz). For software development, the company supports its own ARC MetaWare Development tool kit.

Better Code Density and Performance

Synopsys updated the ARC EM4 and EM6 CPUs to improve code density, performance, and power consumption while also adding more options for embedded-CPU designers. Code density has emerged as a critical design parameter for deeply embedded CPUs, which can be dwarfed in size by their memory blocks.

The company has improved code density by an average of 8% over the previous generation, primarily by adding more 16-bit instructions for more-frequent operations. For frequently occurring 32-bit instructions, Synopsys used dictionary-compression techniques to further reduce code size. For example, the EEMBC benchmark test "cacheb01" is a cache-buster algorithm that makes heavy use of data and function pointers. The new instructions allow the compiler to improve code density for cacheb01 by more than 22%. In addition to saving die area and power consumption for code memory, greater use of 16-bit instructions reduces CPU power by requiring fewer accesses to program memory.

A host of other new features improve the performance of ARC EM CPUs, though the overall effect will depend on the system architecture and software design. Whereas the previous ARC EM version used a bit-serial approach for its hardware divide operation (requiring 34 cycles for a 32-bit divide), the newest version adds a radix-4 divide algorithm that preshifts the operands. The new hardware divider can generate two result bits per cycle while maintaining the same clock rate, cutting the divide time in half.

The ARC EM family now allows two-cycle memory access for closely coupled memory, enabling the CPU to run faster while supporting slower memory blocks such as nonvolatile memory and lower-cost RAM. Designers can partition memory into two different banks to support both

fast and slow memory blocks. This partitioning is transparent to software, requiring no extra tools. The ARC EM CPUs hide much of the memory latency by prefetching instructions into a configurable buffer.

Synopsys created a new interrupt architecture for the ARC EM family, adding more interrupts (up to 240) and supporting up to 16 interrupt-preemption levels. The interrupt priority is programmable, and a vector table enables software interrupt handlers. For faster context switches, the company added hardware support to automatically save and restore register state for either kernel or user-mode stack space. For even faster switch times, designers can configure the new EM family to switch between multiple register banks instead of accessing stack memory. Synopsys added several new instructions to support this new interrupt architecture.

New Memory-Protection Features

Protecting and partitioning memory has become much easier and more comprehensive. The ARC EM design adds a brand-new code-protection feature, preventing programs and the debugger from reading the contents of protected instruction memory. The CPU can execute any instructions in protected code memory, but external hardware signals can disable loading of data from this area. This feature can safeguard a company's IP while still allowing third-party software programming. Programmers can designate any of the 16 equally sized memory partitions as protected. This code-locking feature is often required by software vendors to protect their IP in an SoC design.

The ARC EM CPUs now support a full memory-protection unit (MPU). The (optional) ARC EM MPU implementation is based on the ARC 600 specification, which allows programmable read, write, and execute permissions for both user and kernel-mode code in up to 16 memory regions. The ARC EM version differs slightly from the ARC 600 MPU. For example, the ARC EM allows memory regions as small as 2KB. To augment the MPU, Synopsys also added an optional stack-checking mechanism that is similar to the high-performance ARC 700 specification and prevents access to undefined memory regions beyond the stack pointer.

Although the ARC EM family already supported deterministic processing for real-time kernels, the new versions provide an optional 64-bit real-time clock (RTC) as part of the CPU core. When enabled, the RTC counter runs freely and is controlled by several new ARC EM instructions. Unlike the ARC 700 RTC, the new design doesn't need an extension register or a specialized data path from the counter, simplifying the implementation.

Synopsys has restructured the module hierarchy for the CPU core and memory subsystem to increase the amount of architectural clock gating. The original ARC EM CPUs already provided a hardware-controlled halt mode, as well as a software-controlled sleep mode. The new versions give the designer more control over the mechanisms that bring the CPU and memory out of sleep or halt mode (re-enabling the clocks). The clocks are re-enabled with a wake-up signal from the IRQ unit during sleep mode (assuming the CPU isn't also halted). In systems that support a direct memory interface (DMI), the clocks can be enabled during a DMI transaction. The clocks are also re-enabled during a debug operation that uses JTAG.

To support multiple power modes, the new ARC EM adds eight user-specific states as an operand to the sleep instruction. The designer has complete control over these new sleep modes to implement "deeper sleep" states that control hardware using the three data bits from the sleep instruction. For example, some sleep states can reduce static power by dropping the voltage for logic or memory blocks. The designer could use this control logic to determine which blocks retain data and which power down completely. Using the extra sleep states gives SoC designers full software control over a power-management system optimized for each embedded design.

Synopsys Reaches for ARM's Cortex-M

Table 1 compares the ARC EM CPUs and ARM's newest Cortex-M CPUs. Note that many of these features are optional, and both Synopsys and ARM offer a number of different configurations for each CPU. The Cortex-M0+ is ARM's smallest core (see *MPR 3/19/2012,* "Cortex-M0+ Simplifies 32-Bit MCUs"), using the shortest possible pipeline (two stages). Cortex-M4 has the highest performance in the Cortex-M family and includes DSP features that rely on its 32-bit SIMD instructions (see *MPR 4/12/2010,* "ARM's Digital Signal Controller").

On the basis of published benchmark data, the ARC CPUs can be configured to deliver higher performance with better power efficiency than ARM's best Cortex-M cores. The EEMBC CoreMark scores showcase the tradeoffs for SoC designers that require CPU efficiency. For example, the ARC EM4 CPU requires approximately the same silicon area as the diminutive Cortex-M0+, yet the EM4's CoreMark performance is 30% better while consuming even less power.

Although ARM's Cortex-M0+ sacrifices some performance to be the smallest core, Cortex-M4 currently ranks as the fastest CPU in the Cortex-M family. The M4's Dhrystone mips per megahertz, however, is more than 20% less than the ARC EM's. The ARM CPU narrows the gap to 5% in CoreMarks per megahertz. Although the benchmarks put the ARC EM ahead of Cortex-M4 in CPU performance, the former fits into a power and area budget similar to that of the Cortex-M0+.

ARM's Cortex-M4 supports up to four-way SIMD on 8-bit data or two-way SIMD on 16-bit operands. For integer DSP operations that don't require much precision, its SIMD architecture should deliver better performance on this

portion of the code. The CPU also supports single-cycle multiply-accumulate instructions (using either 32-bit or dual 16-bit operands).

The ARC EM is intended for low-power embedded control applications. For applications that require extensive DSP calculations, Synopsys offers its own high-performance DSP extensions for the ARC 600 and ARC 700 cores. ARM does not allow designers to configure Cortex-M4 without the DSP unit, but Cortex-M3 leaves out the DSP and uses a three-stage pipeline like the M4. For a comparison that omits the DSP, ARM's Cortex-M3 (using the same design parameters from Table 1) consumes 32 microwatts per megahertz and fits in 0.12mm^2 —requiring almost 3x more power and area than the ARC EM4.

On the basis of a CoreMark compilation, the newest ARC EM CPUs report better code density than ARM's Cortex-M CPUs. ARM reports that its Cortex-M0+ requires 4,896 bytes for the CoreMark code, which is almost 28% more than the MetaWare compiler generates for CoreMark execution on the ARC EM. ARM's compiler, however, includes a significant amount of data in its object files, putting the company at a code-size disadvantage. To improve performance, ARM's compiler mixes instructions with literal pools (in-line data constants and branch target tables). The company reports that its CoreMark code size includes 970 bytes of data for Cortex-M4 and 1,036 bytes of data for Cortex-M0+. The MetaWare compiler would likely create similar data structures that are omitted from the .text section but still require memory storage.

Even excluding data bytes from ARM's execution code, we estimate the new ARC EM instructions provide slightly better code density than ARM's famously efficient Thumb-2 instruction set. Since ARM cores target broad markets that include discrete microcontrollers from vendors such as Freescale and TI, the ARM instruction set seldom changes. Synopsys benefits from its narrow focus on SoC designers, since it can tweak its instruction set for better efficiency in high-volume applications. The company seems to have made the right design tradeoffs to shrink the memory footprint of CPUs that are optimized for these embedded SoCs.

Table 1 also highlights a few other features to compare the ARC EM and Cortex-M CPUs. Depending on the software application and SoC architecture, the Synopsys products can be configured to offer several technical advantages. For example, the ARC EM's memory system is closely coupled with the CPU core while supporting an optional cache in the EM6 version. Although both Cortex-M and ARC EM can access memory or peripherals in one cycle, Cortex-M connects through the AHB-Lite interface, which is limited to a single-master protocol. The ARC EM can add a full AHB interface and offers support for a BVCI interconnect to memory and peripherals.

Synopsys has also substantially upgraded its ARC EM interrupt handling and now offers high-performance features similar to those of ARM's top-end Cortex-M products, including support for more interrupts, vectored interrupts, and hardware acceleration for context save/restore. Synopsys delivers more configurability than ARM for most of these options while also adding unique capabilities such as direct support for multiple register files to enable single-cycle context swaps.

	Synopsys ARC EM4	Synopsys ARC EM6	ARM Cortex-M0+	ARM Cortex-M4
CPU Architecture	ARCv2	ARCv2	ARMv6-M	ARMv7E-M
Pipeline Stages	3 stages	3 stages	2 stages	3 stages
CoreMarks/MHz	2.29CM/MHz	2.29CM/MHz	1.77CM/MHz	2.19CM/MHz
Dmips/MHz	1.52DM/MHz	1.52DM/MHz	0.93DM/MHz	1.25DM/MHz
Max CPU Speed*	390MHz	420MHz	200MHz§	250MHz§
Minimum Die Size†	0.04mm^2	0.05mm^2	0.04mm^2	0.17mm^2
Minimum Power (microwatts/MHz)†	8.4 µW/MHz	9.4 µW/MHz	11.2 µW/MHz	33.0 µW/MHz
CoreMark Code Size‡	3,798 bytes	3,798 bytes	4,896 bytes	4,766 bytes
Closely Coupled Memory	Up to 1MB instr/data	Up to 1MB instr/data	Memory on SoC bus	Memory on SoC bus
I/D Cache	None	Up to 32KB	None	None
Floating-Point Unit	SP/DP	SP/DP	None	SP only
MPU Regions	1, 2, 4, 8, or 16	1, 2, 4, 8, or 16	8 regions	8 regions
Max Interrupts	240 interrupts	240 interrupts	32 interrupts	240 interrupts
Sleep Modes	8 states	8 states	2 states	2 states
Extra Features	Code protect, stack check, hardware context switch, ISA extensions		None	32-bit SIMD DSP
SoC Interface	AHB, AHB-Lite, BVCI		AHB-Lite	AHB-Lite

Table 1. Comparison of DesignWare ARC EM and ARM CPUs. All data for TSMC 90nm LP, seven-track library, 1.2V, and 25°C. Base usable configuration excludes ETM, MPU, FPU and debug. ARM includes DSP and 1 IRQ + NMI. *Synthesized for speed; †synthesized for power/area; ‡ARM code size using ARM Compiler 4.1 (includes data for literal pools and excludes UART/timer); ARC EM code size using MetaWare 8.9. (Source: vendors, except §The Linley Group estimate)

Well Suited to Embedded CPUs

Synopsys has avoided the processor industry's bruising battles over CPU instruction sets and software ecosystems by focusing on deeply embedded applications. In these designs, high-level issues give way to the technical realities that affect power and cost in embedded systems. As more CPU cores are sprinkled throughout every embedded device, designers will select efficient CPUs that deliver the required performance while consuming the minimum amount of silicon area and power. Designers of deeply embedded systems care less about the instruction set and more about whether the vendor provides robust development and debugging tools and whether it has long-term viability and commitment to customer support.

The ARC CPU architecture has more than a 20-year history and is now backed by EDA powerhouse Synopsys. The company's recent improvements to the EM4 and EM6 CPU cores show that Synopsys is investing in the growth of its ARC products as a major component of the DesignWare IP library. Although the EDA company doesn't brag much about its own CPUs, the new ARC cores are looking very attractive and should start getting much more attention. ♦

Price and Availability

The Synopsys ARC EM cores are available now in the company's DesignWare IP library. The company does not publicly disclose licensing fees or royalties. For more information about the ARC EM family, access *www.synopsys.com/IP/ProcessorIP/ARCProcessors/ARCEM*.