The Linley Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

Adapteva: More Flops, Less Watts

Epiphany Offers Floating-Point Accelerator for Mobile Processors

By Linley Gwennap {6/13/11-02}

Most mobile-processor vendors focus on improving MIPS per watt, a common measure of power efficiency. Adapteva, however, has taken on a different problem: increasing floating-point operations per second (flops) per watt. The tiny startup has developed and tested a unique architecture that delivers industry-leading flops per watt. Although some people think floating-point (FP) performance is needed only in supercomputers and specialized signal-processing applications, this type of powerful FP engine could soon be coming to a smartphone near you.

Adapteva offers its Epiphany multicore architecture as an intellectual-property (IP) core that scales to various performance levels. The company rates its basic 16-core design at 19Gflops while drawing just 270mW (typical) when implemented in a 28nm LP process. At 2mm^2, this design would only modestly increase the cost of a typical mobile application processor. Configured as a coprocessor, Epiphany could deliver impressive FP capability within the power budget of a typical mobile device.

Why would a mobile device need such capability? Voice recognition can ease text entry on handheld devices with tiny or nonexistent keyboards, but the voice capabilities of most mobile devices are adequate only for simple command-and-control functions. For general speech-to-text capability, many advanced voice-recognition algorithms depend on floating-point performance. Services such as Google Voice Search decode the voice signal on a remote server, which has plenty of FP performance, but this approach adds latency and doesn't work if the network is inaccessible. Performing high-quality voice recognition directly on a mobile processor will require a new approach.

In a recent article, we discussed the trend toward visual computing in mobile devices (see *MPR 5/30/11,* "Visual Computing Becomes Embedded"). Visual computing can enable gesture-based gaming, advanced user interfaces, augmented reality, and even improved health and safety. Visual processing, however, requires many more flops than voice processing. Adapteva's architecture can deliver the performance required for visual computing.

Custom Architecture With FP Focus

To optimize its CPU for power, Adapteva started with a clean slate instead of a standard instruction set. Epiphany uses a simple RISC instruction set, which focuses on floating-point operations and load/store operations, so it omits complex integer operations such as multiply and divide. Each 32-bit entry in the 64-entry register file can hold an integer or a single-precision floating-point value.

As Figure 1 shows, the CPU itself is a simple two-issue design capable of executing one integer operation

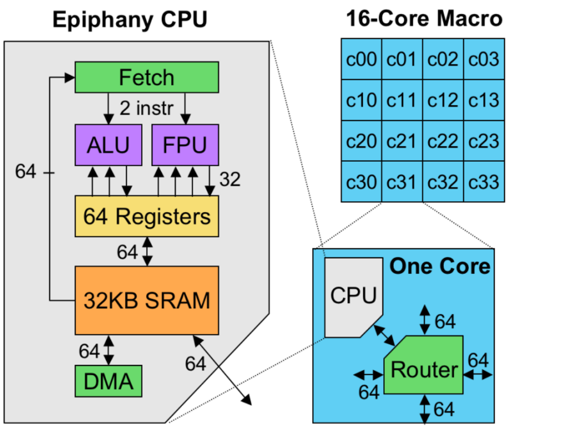

Figure 1. Epiphany CPU core and network design. Each core contains a simple CPU and a router for the mesh network. The network can be extended to a 64×64 array, although the initial implementation contains 16 cores.

and one FP operation per cycle. The CPU relies on Adapteva's compiler to optimally arrange the instructions rather than reordering instructions in hardware. To minimize power and area, the design has no dynamic branch prediction, although its short (six-stage) integer pipeline keeps the misprediction penalty small. As a scalar integer design, the CPU achieves an EEMBC CoreMark score of about 1.3/MHz—a little less than that of an ARM9 CPU. By comparison, a modern high-performance CPU such as Cortex-A9 can achieve 2.9/MHz.

Epiphany is optimized for floating-point programs, not an integer test like CoreMark. FP loads and FP stores count as integer operations, so the CPU can execute an FP calculation while loading data for the next calculation. The instruction set supports load and store double instructions that access two consecutive 32-bit registers, taking advantage of the 64-bit path from the SRAM to the register file. Using these instructions, the CPU can load two operands per cycle.

The single-precision FPU can execute one FP multiply-accumulate (FMAC) per cycle to achieve its peak rate of two FP ops per cycle. The FPU is not fully IEEE 754–compliant but uses standard data formats and rounding modes. It is optimized for FMAC operations and has a latency of four cycles. The instruction set also supports FP addition, subtraction, and multiplication but not complicated operations such as divide and square root. Lacking support for double precision, denorms, and division, the FPU is not suited to scientific or technical computing; it is tuned for signal processing.

The CPU core contains a single direct-mapped 32KB SRAM. Software is responsible for loading program instructions and data into this SRAM. The design also eschews memory management of any kind, implementing a flat 32-bit memory space without any protection. This approach eliminates both the die area and performance overhead of a traditional TLB.

As Figure 1 shows, Epiphany uses a tile approach to arrange the cores in a mesh network. This approach is similar to Tilera's (see *MPR 11/5/07,* "Tilera's Cores Communicate Better"). Each core can transfer 64 bits per cycle in each of five directions: north, south, east, west, and to/from the CPU. Using the mesh, each CPU can access the SRAM of any other CPU. With this approach, the 16-core design can be viewed as having 512KB of directly accessible SRAM, albeit with variable latency. The mesh design can be easily expanded to include additional cores.

For optimal performance, instructions and data must be preloaded into the CPU's local SRAM. Each core contains a DMA engine that can be configured to autonomously prefetch data under software control. The SRAM is divided into four 64-bit-wide banks, allowing it to ideally support an instruction fetch, a load, a DMA access, and an external access (e.g., from another core) on each cycle.

Designing for Low Power

Adapteva's goal is to maximize Gflops per watt, so simply creating a scalable FP architecture was not enough. The simplified CPU design yields many power savings compared with a typical high-performance design such as ARM's Cortex-A9. Epiphany's short pipeline reduces the need for latches and bypass logic, and its simpler instruction set avoids functions such as ARM's preshift that are difficult to implement in hardware. Unlike the A9, it does not waste power reordering instructions; having no legacy code base, Adapteva can rely on the compiler for this task. The flat unprotected address space eliminates the power that a memory-management unit (MMU) would consume.

A big power savings comes from the use of SRAM instead of cache. A cache burns power on each access to search through all the tags to find a match (or not). Multicore designs typically snoop the caches to maintain cache coherency, thus using more power. A cache must also determine when to transfer data to and from main memory and then perform such transfers. In a cacheless design, software takes on the burden of managing the SRAM and programming the DMA engine to handle the transfers. Some CPU cores can be configured with tightly coupled memory (TCM) instead of cache, achieving a similar power savings. Most designers prefer cache, however, because it simplifies the software.

In Epiphany's unified-memory design, programmers must also be concerned about SRAM-bank conflicts. Avoiding conflicts between instruction fetches and data loads can be challenging. When working with long data vectors, however, the data loads and the DMA accesses tend to naturally synchronize, using different banks on each cycle.

The mesh network reduces power compared with a traditional crossbar interconnect. All signals travel from one tile to its immediate neighbor, minimizing signal length and thus the drive current. These short signals also enable the network to operate at the same high clock speed as the CPU. The tile approach also simplifies physical design, as the designer can connect each tile to the next simply by placing them beside each other.

The downside is that transactions to cores other than immediate neighbors require multiple cycles to complete. In a 16-core mesh, the average number of hops is 2.625, assuming a completely random distribution of accesses. Optimized programming can greatly reduce this figure. For a read request, the latency is twice the number of hops, since the transaction must travel from the target to the source and back again. This latency is not guaranteed, since congestion on the mesh can cause delays.

Clock distribution often consumes a sizable portion of total chip power, because large drivers are required to minimize skew. For clock distribution, Adapteva chose a simple wave-propagation scheme, as Figure 2 shows.

In this approach, clock skew accumulates as the signal progresses through the cores. Because the tiled design eliminates global signals, this accumulated skew is not important. The skew between any two neighboring cores is still minimal. In addition to reducing clock power, this method simplifies routing.

Epiphany also saves power by using the now-common techniques of extensive fine-grained clock gating, shutting off the clock to unused function units and entire cores on a cycle-by-cycle basis, and using low-V_T transistors only when necessary. Although these techniques are helpful, the company believes most of the power savings comes from its unique CPU microarchitecture and low-overhead mesh network.

Silicon on a Shoestring Budget

Like many IP vendors, Adapteva is a small company with minimal funding. The basic Epiphany architecture is essentially the work of a single engineer, Andreas Olofsson, who formerly worked on the TigerSharc DSP at Analog Devices. Drawing on his life savings, Olofsson worked on Epiphany for two years before raising $2 million in funding. The company is now up to four employees.

Unlike many IP vendors, Adapteva doesn't merely have RTL; it has working silicon and an initial customer. Producing a chip on a shoestring budget was a huge challenge. First, Olofsson convinced Magma Design to give him access to a suite of chip-design tools for far less than the usual $1 million or so. Using these tools, he created a physical design for a 16-core test chip in just six weeks. The design comprised 40 million transistors, including 512KB of SRAM. The test chip includes no memory controller or peripherals and has only 8GB/s of LVDS I/O to move data into and out of the mesh network.

Fabricating a chip usually incurs million-dollar maskset and tapeout fees. Olofsson instead used a shuttle run, which allows multiple companies to share the tapeout cost. Since the test chip measured only 11.5mm^2 in a 65nm process, it could be combined with several other chips in a single mask set. In this way, a typical shuttle run costs $50,000 to $100,000. After the wafers are built, the foundry dices the individual chips for each of the shuttle-run customers. Olofsson then located a small consulting firm that was able to design a custom 484-contact BGA and deliver 50 packaged parts for only $15,000.

With test chips in hand, Olofsson could measure the actual power consumed by the design, proving that he had met his goals of power reduction. While running an FFT program on all 16 cores, the chip consumes 450mW at 500MHz and 1.0V. It can run as fast as 1.0GHz, but this requires increasing the supply to 1.1V, slightly impairing the power efficiency. At 500MHz, the 65nm chip has a theoretical maximum performance of 16Gflops, or 35Gflops per watt. Each core can complete a 1,024-point complex FFT in about 40,000 clock cycles. Bittware, the company's first customer, will sell these chips in a DSP accelerator card.

Now Available for Licensing

With its new funding, Adapteva retargeted its 16-core design to GlobalFoundries' 28nm SLP technology. Moving from a high-speed 65nm process to a low-power 28nm process will boost the clock speed from 500MHz to 600MHz at maximum power efficiency. Since the CPU and the network run at the same speed, FP performance scales accordingly. At this speed, the design is rated at 70Gflops per watt, owing to the lower capacitance of the 28nm transistors. More significantly, the 28nm process reduces the die area of the 16-core design to 2.05mm^2 (unlike the test chip, this area does not include a pad ring). The clock speed can be boosted to 700MHz using a higher supply voltage. These figures are not validated in silicon but are based on a fully routed layout.

The company is licensing the 16-core design as a hard macro, which is now available for design starts. Because the tiles connect by abutment, customers can scale the design to larger numbers of cores (as many as 4,096) using multiple instances of the same macro. Adapteva also plans to release the Epiphany architecture as a soft core, enabling customers to target Epiphany to any desired foundry, simplifying integration into an existing processor design. The soft core can also vary the amount of SRAM from 32KB to 64KB per core. The company has not disclosed a schedule for the soft core.

Although Bittware is using the Epiphany design for traditional DSP applications such as radar, ultrasound, and cellular base stations, Adapteva feels its power efficiency is best suited to the mobile market. In this market, Epiphany could be incorporated into a mobile application processor or integrated smartphone processor, acting as an accelerator to the main CPU. In this way, the operating system and application software could continue to use the main CPU, but FP-intensive applications could access the Epiphany accelerator though an API, much like the way a

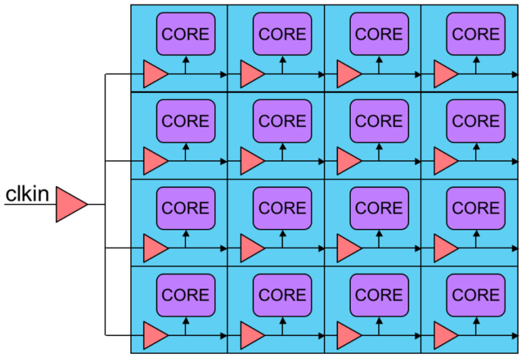

Figure 2. Low-power wave-propagation clocking scheme. Instead of minimizing global clock skew, this approach synchronizes only the neighboring cores, greatly reducing the power consumed by the clock tree.

Price and Availability

Adapteva provides the Epiphany design as a number of hard macros ranging from 16 to 4,096 cores. It is now available for licensing. The company also plans to offer Epiphany as a soft IP design. For more information, access *www.adapteva.com.*

graphics or video accelerator is used today. The API could provide access to a library of routines for specific functions, such as voice or face recognition, or like OpenCL, it could provide access to basic computational capabilities.

This type of software package will have to be developed by the licensee. For its custom instruction set, Adapteva provides a standard C programming environment based on GCC, GDB, and Eclipse. It has developed a few test routines, such as the aforementioned 1,024-point FFT, to validate the design's performance, but it offers no library code.

More Efficient Than CPUs or GPUs

Although Adapteva claims an advantage of as much as 50× in performance per watt, the company compares its basic 16-core chip, which lacks so much as a memory controller, against complete system-on-a-chip (SoC) processors. Since Adapteva is now marketing Epiphany as an IP core, we compare it against two other IP cores: ARM's Cortex-A9 CPU and Vivante's GC2000 GPU. Unlike Epiphany, neither of these designs is optimized solely for floating-point performance, but they perform surprisingly well. Both use quad-core single-instruction multiple-data (SIMD) designs to achieve FP performance similar to that of the 16-core Epiphany.

ARM offers Cortex-A9 as a hard macro designed for TSMC's 40nm LP process. The power-optimized version of this design consumes 4.6mm^2 for a dual-core implementation that includes two Cortex-A9 CPUs with Neon units, 32KB instruction and data caches for each CPU, an L2 cache controller (but no L2 cache), and coherence logic. The Neon SIMD unit supports eight single-precision FP operations per cycle using a 128-bit data path. This version is rated at 500mW (typical) at 800MHz. This power does not include Neon, so we have added an extra 30% (150mW). We have scaled these figures to 28nm by multiplying the die area by 0.5 and the power per megahertz by 0.7 to produce the data in Table 1. We pushed the clock speed to 1.2GHz to better align with Epiphany's performance, but the speed does not affect flops per watt.

	Adapteva Epiphany*	ARM Cortex-A9*	Vivante GC2000
# of Cores	16 cores	4 cores	4 cores
Total SRAM	512KB SRAM	256KB L1$	152KB SRAM
FP Ops/Cycle	32 FP	16 FP	16 FP
Clock Speed	600MHz	1.2GHz†	1.5GHz†
Peak Mflops	19,200	19,200	24,000
Power (typ)	270mW	1,350mW†	650mW†
Mflops/W	71,000/W	14,000/W	37,000/W
Die Area	2.05mm^2	4.6mm^2†	2.8mm^2†
Mflops/mm^2	9,400/mm^2	4,200/mm^2	8,600/mm^2

Table 1. Comparison of Epiphany to other IP cores. For floating-point performance, Epiphany has a 2× to 5× advantage in power efficiency and a small advantage in die-area efficiency. All data converted to 28nm LP for comparison. *Power-optimized hard core. (Source: vendors, except †The Linley Group estimate based on vendor data for 65nm or 40nm designs)

Vivante's GC2000 is a 3D-graphics accelerator used in Freescale's i.MX6 application processor (see *MPR 4/25/11,* "Freescale's i.MX6 Graphically Detailed"). The GC2000 is designed to render 200 million triangles per second, but it is based on four programmable shaders that operate at 1.5GHz in 40nm LP and are each capable of four single-precision FP operations per cycle, operating in SIMD fashion. Vivante rates its dual-core GC1000 design at 5.6mm^2 and 219mW (typical) at 500MHz in a 65nm LP process. We have doubled these numbers for a quad-core design and then scaled them down to 28nm LP. We held the 1.5GHz clock speed constant from 40nm to 28nm to provide some headroom for operating at a lower voltage; again, this choice of speed does not affect flops per watt.

This comparison shows that Adapteva delivers on its goal of having the best power efficiency for floating-point calculations. At 71Gflops/W, it is twice as efficient as the Vivante GPU and five times better than Cortex-A9. Vivante's GPU is designed for graphics, not pure FP performance. Cortex-A9 is hampered by its complex CPU design and its use of cache memory instead of SRAM. Cortex-A9 implements a 64-bit Neon unit; the newer Cortex-A15 includes a 128-bit Neon unit that will double FP performance, albeit at somewhat higher power (see *MPR 11/22/10*, "Cortex-A15 'Eagle' Flies the Coop").

As one might expect, Cortex-A9 also uses much more die area; the quad-core configuration requires 4.6mm^2 to match the FP performance of the 2.05mm^2 Epiphany design. This comparison is somewhat unfair, however, as the A9 is a powerful CPU complete with an MMU and an L2-cache controller. Die area for Vivante's GPU is only slightly larger than for Epiphany. Although the GPU includes texture units, controllers, and other graphics accelerators that Epiphany does not require, it also includes much less SRAM.

The complexity of the Cortex-A9 design greatly simplifies software development. Programmers do not need to manage the local SRAM or worry about bank conflicts; the cache subsystem does this work. Many vendors provide compilers and other software-development tools for the popular ARM instruction set. Sample code and drivers are also widely available. The SIMD design of the Neon unit requires some accommodation, but simple loop unrolling generally provides adequate parallelism. Using the FP performance of the GC2000 is also straightforward, because Vivante supports the OpenCL 1.1 API. Programming with OpenCL is more complicated than writing C code, but software written for this standard API will run unmodified on many types of floating-point accelerators.

Showing the Way to Low Power

Our analysis shows that Adapteva has met its goal of delivering industry-leading Gflops per watt. On this metric, the Epiphany design performs far better than popular ARM CPUs and significantly better than even GPUs, which do most of their work in floating-point math. Adapteva has achieved this advantage using a CPU that is optimized for floating-point MACs, eschewing unnecessary instructions and complications such as cache controllers and MMUs. These choices, however, complicate the programming task.

Epiphany requires less die area than a CPU or GPU for the same level of floating-point performance. This comparison is somewhat moot, however, because any high-end mobile processor already has a CPU (probably with the Neon accelerator) and a GPU. With GPUs moving to a programmable architecture, we are already seeing a trend toward vendors such as Vivante supporting general-purpose GPU (GPGPU) computing using the OpenCL API. Using the CPU or GPU for FP-intensive tasks requires zero incremental die area, whereas Epiphany increases the die area by a nominal amount.

The 16-core Epiphany design provides FP performance no better than that of high-end CPUs and GPUs that are already sampling or will sample later this year, so a processor using this design will not have any exceptional capabilities, other than longer battery life when performing these tasks. Designers can implement 32 or even 64 Epiphany cores to gain a clear performance advantage for FP-intensive tasks, but these designs consume considerably more die area.

Before committing extra silicon cost and design time to Epiphany, we expect mobile vendors will wait until the usage model for FP becomes better established. But if visual computing becomes an integral part of the user interface and application software, an FP accelerator could become as common as today's video accelerators. ♦