Imperfect Processing: A Functionally Feasible (and Fiscally Attractive) Option, Says Singular Computing

October 22, 2013 |
Posted in Processors, Software Development

http://www.bdti.com/InsideDSP/2013/10/23/SingularComputing

Conventional wisdom dictates that an arithmetic circuit that generates inexact results is faulty. But Joe Bates, founder and president of Singular Computing, thinks that conventional wisdom may be mistaken, at least for certain classes of applications. Bates, in his own words, has spent roughly half his professional life in academia and the other half involved with various startups. Reflective of the former focus, he is also an adjunct professor at Carnegie Mellon University (CMU) and has held positions at MIT, Johns Hopkins, and Cornell.

Bates has a long-standing interest in various areas of artificial intelligence (AI), and his activities in this field have been notably influenced by well-known computer scientist Takeo Kanade's belief that increases in computing performance are critical to significant AI advancements. Kanade, a fellow professor in the Robotics Institute at CMU, is one of the world's foremost researchers in computer vision, reflected in (among other things) his name being represented in the name of a well-known feature tracker, the Kanade-Lucas-Tomasi (KLT) algorithm. Not surprisingly, therefore, computer vision is one of the key applications that Bates has focused on, along with speech recognition and other "deep learning" tasks.

Around a decade ago, Bates had a breakthrough realization that the human brain's neurons don't do exact arithmetic; they were only about 99 percent right on average. What, he wondered, would happen if he tried to build hardware that wasn't neural in design, but which implemented approximate arithmetic? What Bates discovered was that he could shrink the silicon area consumed by each arithmetic unit by approximately 100x versus the DSP-, FPGA- or GPU-based alternatives, an especially attractive outcome in AI and other highly parallelizable applications. And equally important, he found that the incremental performance requirements of software-based arbitration algorithms run on a higher-level processor, evaluating and selecting among the contending results of multiple approximation arithmetic units operating on the same source data set, were relatively insignificant in these same applications.

Imperfect computing, Bates freely admits, is not an idea that's unique to his startup company, Singular Computing. Less than two weeks ago, for example, Joel Hruska at ExtremeTech published the informative article "Probabilistic computing: Imprecise chips save power, improve performance," which covers the research being done by Christian Enz, the Director of the Institute of Microengineering at the École Polytechnique Fédérale de Lausanne. Hruska's article references similar work being done at Rice University, which he explored in greater detail in a writeup published in May of last year. And Hruska also mentions Intel, which among other things "has explored the idea of a variable FPU that can drop to 6-bit computation when a full eight bits isn't required."

In an interview with BDTI earlier this month, Bates also mentioned Intel's ongoing exploration of approximation arithmetic, intended to reduce required transistor count and/or power consumption, along with the work of IBM and, in a less public fashion, by "other large famous companies." Intel, for example, wants to build an "x86 CPU that uses half the power," says Bates. Although the approximation arithmetic concept may be common to multiple companies and research organizations, specific implementations of the concept vary.

In Singular Computing's case, the core arithmetic unit does "floating point-like" operations (add, subtract, divide, multiply, and square root) in a single cycle and pairs with 256 words' worth of high-speed memory and multiple local registers to form an APE (approximate processing element) (**Figure 1**). APEs communicate with each other over a neural network-reminiscent massively parallel grid interconnect scheme; Bates estimates that modern process lithographies could enable the cost-effective integration of several hundred thousand APEs on a single chip, alongside an ARM or other host processor.

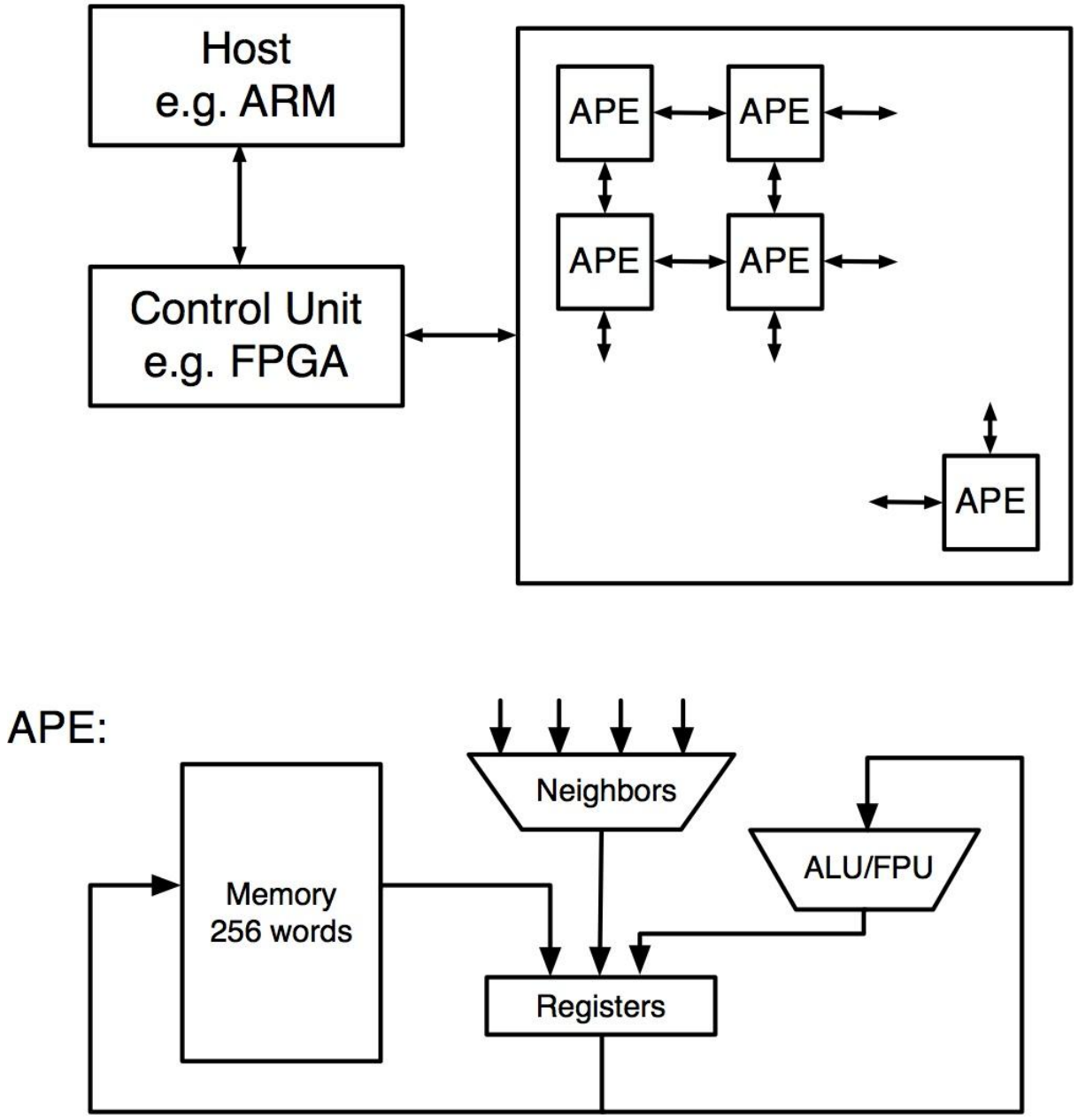

Figure 1. Each APE (approximate processing element) comprises a single-cycle arithmetic unit, multiple registers, and a modest allocation of high-speed local memory. A mesh-based topology connects APEs to each other, to control logic and to the CPU being accelerated.

Bates stresses that the only revolutionary aspect of such a machine is its APE-implemented approximation nature: "everything else about this machine you can read about in a 1994 textbook." The programming model, for example, is by design "straightforward and simple...familiar and old," leveraging conventional software languages such as C instead of some more exotic scheme. Bates claims that a small amount of incremental software can, for specific applications, recover whatever level of precision is needed. There are ways that vary by application, for example, to prevent error from accumulating.

As a case study of the APE-based machine's potential, Bates discusses the classic k-nearest neighbor pattern recognition algorithm. **Table 1** shows the results, using both 200- and 800-element vector lengths, of the accuracy delivered by an APE-based accelerator if it's allowed to only output one result, versus if it generates two, three or four possible results for software running on the host CPU. The hardware's accuracy limitations, according to Bates, can be surmounted using almost no incremental time or energy consumption, while providing substantial silicon area savings versus the conventional, 100 percent-accurate hardware alterative.

Vector length	% correct results if the APE-based accelerator finds the best:			
	1 answer	2 possible answers	3 possible answers	4 possible answers
200	88.6	98.0	99.6	99.9
800	79.5	93.1	97.4	99.0

Table 1. In an APE-based k-nearest neighbor pattern recognition scheme, APE accelerators discern several matching-candidate finalists, with the CPU then selecting the best match.

Here's another example: deblurring using the Richardson-Lucy deconvolution algorithm. The upper left quadrant of **Figure 2** shows the original image, which was then blurred to create the lower-left source data for the experiment. While an IEEE floating point-compliant DSP accelerator produces a reasonable deblurred approximation (in the upper right quadrant) of the original, it requires substantial transistor count (and associated power consumption) to do so. The APE-based alternative approach generates comparable deblurring results (lower right) with ~100x less silicon area, and in only twice the processing time that the DSP needs, according to Bates.

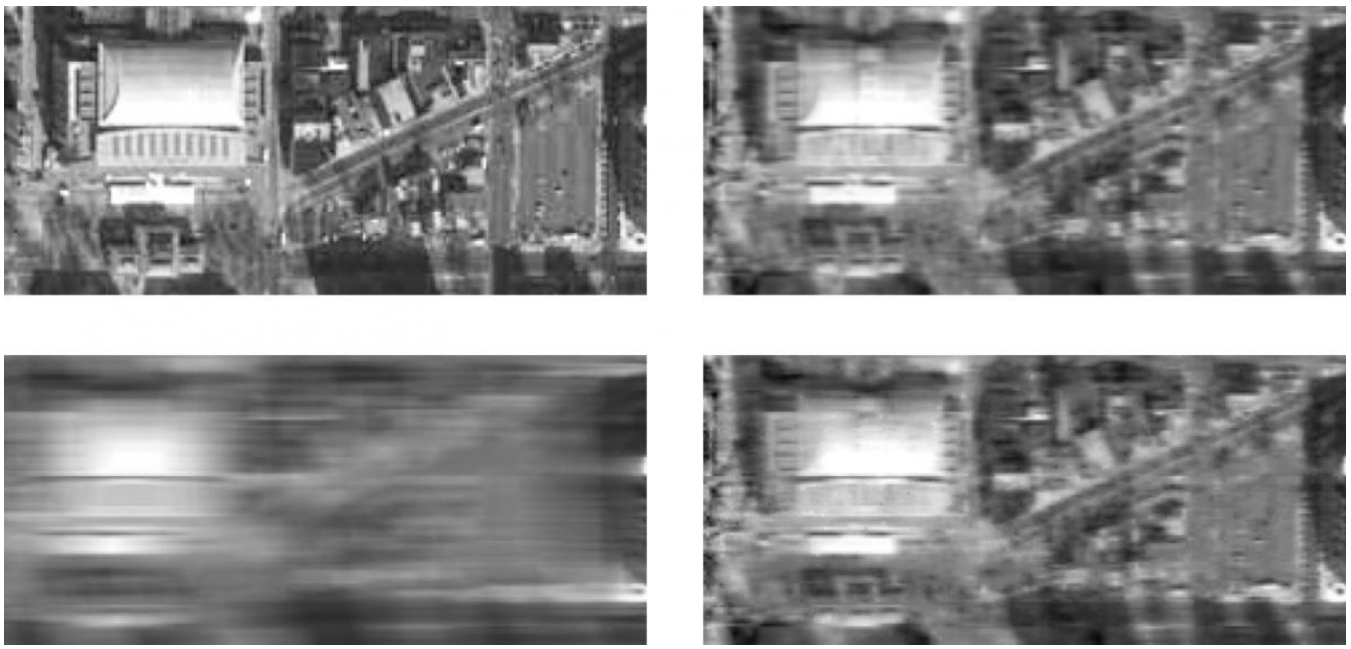

Figure 2. APE-based deblurring (bottom right) delivers roughly comparable quality results to those supplied by a conventional IEEE floating point accelerator (top right), with significant size and power consumption savings, while still being substantially faster than a CPU software-only approach, according to Singular Computing.

Finally, consider an evaluation done for the Office of Naval Research by Singular Computing in partnership with Charles River Analytics (**Figure 3**). The experiment implemented feature-based tracking, first using algorithms running on a CPU alone (on the left) and then by combining the CPU with APE-based hardware algorithm acceleration (on the right). The APE-accelerated scheme delivered comparable quality at 89x the frame rate of the software-only scheme, while consuming 72x less power, according to Bates. As with other examples cited by Bates, these results suggest that a CPU managing low-precision "workers" can yield high-precision CPU-like results, with significant advantages in size, weight, power and cost.

Figure 3. Feature-based object tracking via APE-based hardware acceleration (right) runs at an 89x higher frame rate than the alternative software-only scheme (left), according to Singular Computing's Joe Bates, while consuming 72x less power.

To date, the bulk of Singular Computing's APE-based evaluation work has been done in FPGA-emulated hardware in partnership with Charles River Analytics, and financially sponsored by the Office of Naval Research. DARPA has also more recently funded 45 nm ASIC-based hardware development, which Bates forecasts will result in a 4,000-APE 25 mm^2 single-chip prototype in roughly six months' time. This prototype, extrapolated to a board containing not only multiple Singular Computing accelerators running at greater than 100 MHz, but also an ARM processor and FPGA-based control unit, will then be evaluated by BAE Systems, by Trevor Darrell's computer vision group at the University of California, Berkeley, and by Takeo Kanade and his colleagues and students at CMU, among others.

Bates intends for Singular Computing to be both an R&D firm and an IP provider, and is therefore interested in working with partners to "scale up" the technology. When the under-development ASIC-based prototype boards appear next year, he hopes to join with others to quickly cultivate practical applications for the technology, which will fund further low-volume research hardware. To wit, he notes that lately notable interest has been coming from large companies involved in "smartphone development and sales," and focused on embedded vision, speech recognition and other related AI applications.