



A Passel of Processors

NVIDIA's Tesla T10P Blurs Some Lines
(Kevin Morris)

Picture this architecture – a high speed application processor doing control coupled to an accelerator comprised of a mass of processing elements ready to power-parallelize compute-intensive components of a complex problem. Sound familiar? Supercomputers have taken advantage of acceleration using schemes like this for a while. People using FPGAs for co-processors do it all the time.

Now, picture a new chip with 1.4 billion transistors, an array of 240 cores, and a processing throughput equivalent to about 1 TeraFLOPS. Many readers of this publication would probably guess a new FPGA, right?

With the new Tesla T10P GPU, NVIDIA is making a lot of us editors re-work our glossaries. The T10P is a GPU that's aimed directly at the high-performance computing community, not just accidentally clipping it with a bank shot while going after the real target market of graphics acceleration. The T10P represents

NVIDIA's second generation of CUDA (Compute Unified Device Architecture) GPUs (with the Tesla 8 being the first). CUDA is a C dialect with specific constructs for parallelism, and it allows direct access to the low-level hardware capabilities of the processor cores of the GPU. Why would we want that? To do non-graphics applications, of course.

You see – unless your performance-critical application happens to involve a lot of shading and texture mapping, GPUs have traditionally been a locked treasure chest, not ready to share all that parallel-processing goodness with those who aren't trying to blast billions of bits onto a screen. Many people have always known that processing power was in there, though, and an access mechanism like CUDA is the key that lets them get in to put all those processors to work - doing a lot of interesting tasks that are most certainly NOT graphics acceleration.

This idea of using GPUs for general purpose processing is called GPGPU - General Purpose (processing with) Graphics Processing Units. (The acronym department kinda' blew it with that one.) CUDA is not the first effort along those lines. ATI (Now AMD) had a beta API called (OK, these guys are much more rock-n-roll with their acronyms) "Close to Metal" (CTM) that allowed direct access to the low-level instructions in their R580 GPUs. CUDA is the effort that seems to be getting industry traction right now, however, and NVIDIA has beefed up the latest Tesla processor with their sights set straight at that market.

Now, there is a world called "reconfigurable computing" that many of us FPGA folks often visit. In the reconfigurable computing world, people have worked for years (and arguably decades) to harness the inherent hardware parallelism of FPGAs in order to create co-processors in high-performance computers (HPCs). If, they reason, the programming model can be simplified enough, previously unattainable processing power (and, more importantly, processing power per Watt) could be obtained through the magic of FPGAs.

Of course, all those efforts have failed to pull many but the hardest-core performance-hungry over the chasm into the realm of FPGA-based reconfigurable computing. Unless you want to learn HDL or trust a bleeding-edge software-to-HDL compilation/synthesis tool, the task of getting your critical algorithm onto FPGA hardware was untenable - even for high-IQ cognoscenti like rocket scientists, cryptographers, and genetic engineers. When those folks are afraid that your solution is "too complicated to use," you have something of a problem.

High performance computing has chugged along for several years with this gap of capability versus usability. On one side, the challenge is to see how much power one can purchase from the utility company in order to operate and air-condition racks and racks of blades covered with multi-gigahertz one-to-four-core processors. This is the "easy and expensive" solution. Oh the other side, we have reconfigurable computing folks spending tiny fractions of that budget on both hardware and electrical current, but blowing the difference trying to hire VHDL and Verilog experts to code up complex biomedical, geological, and financial algorithms in hardware description languages.

Now, NVIDIA brings GPGPU right into the middle of that gap, and the solution has a number of advantages over either end of the spectrum. The new GPU's processor cores have double-precision floating point math built right in, so there is no special software mangling required for complex scientific algorithms. The lower-demand administrative and control aspects of your problem can be

handled by conventional processors, while the highly-parallelized part (coded with CUDA) can be offloaded to the GPGPU for some mad multi-threaded mayhem. CUDA creates an executable that is independent of the number of processors available, and it can dynamically take advantage of just about any number you want to wire in, so performance scales just about linearly with the number of processor cores. The bottom line is that you can build an impressive supercomputer with a bunch of Teslas on a tiny fraction of the cost and power budget of a massively parallel rack system, and you can still program it using something very close to conventional software engineering techniques.

NVIDIA is also releasing a couple of pre-fab systems for HPC with the new processors. The S1070 1U comes in a 1U EIA 19" rack form factor, and it includes 4 Tesla T10 processors totaling 960 cores and operating at 1.5GHz. This yields an overall performance of 4 TeraFLOPS. System memory of 16GB is included (each T10 can address up to 4GB). The memory bandwidth is 408 GB/sec peak using 2048-bit, 800MHz GDDR3 (512 bits for each T10). The system I/O goes through 2 PCIe x16 Gen2s. Power dissipation for these 4 TeraFLOPS? 700W! Try THAT with your typical blade-based processing system. In comparison with a 4-core 1U CPU server, NVIDIA claims the Tesla-based blade can deliver 17x lower cost and 21x lower power per processing power.

If you'd rather just stick a T10 into your desktop PC at home and blaze away in your basement, NVIDIA also provides the Tesla C1060 Computing Processor. It comes in a dual-slot full ATX form factor, mates up to your favorite motherboard via PCIeX16 Gen2, and boasts a single T10 with 240 cores running at 1.33 GHz. This can add an extra teraFLOP or so to your PC for an investment of only 160W.

NVIDIA says that over 250 customers are already using CUDA-based acceleration for high-performance computing problems, and the addition of the T10 with its increased performance, larger memory space, and double-precision math will almost certainly boost interest significantly. CUDA boasts an impressive portfolio of applications, including problems as diverse as genetic sequence alignment, astrophysics, financial modeling and medical imaging.

Kevin Morris, FPGA and Structured ASIC Journal

June 17, 2008