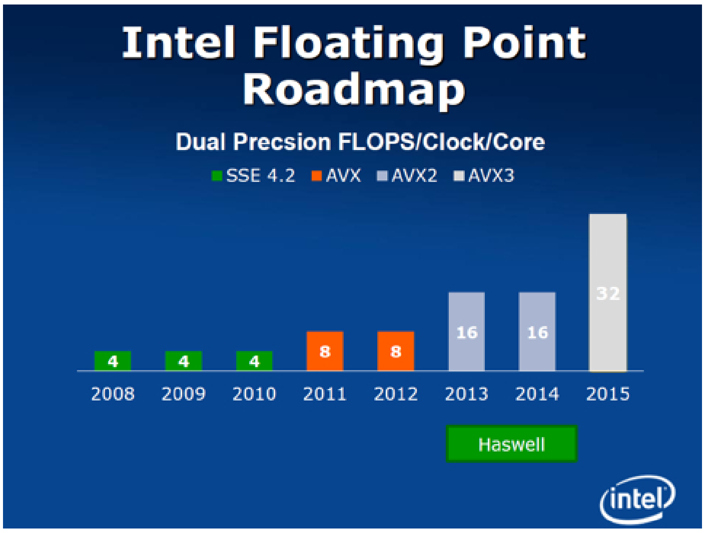
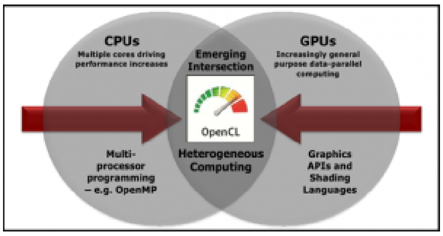
Is the CPU slowly turning into a GPU?




CPUs
Multiple cores driving performance increases
Emerging Intersection
GPUs
Increasingly general purpose data-parallel computing
OpenCL
Multi-processor programming – e.g. OpenMP
Heterogeneous Computing
Graphics APIs and Shading Languages

It's all in the plan?

Years ago I was surprised by the fact that CPUs were also programmable with OpenCL ? I solely chose that language for the cool of being able to program GPUs. It was weird at start, but cannot think of a world without OpenCL working on a CPU.

But why is it important? Who cares about the 4 cores of a modern CPU? Let me first go into why CPUs have had mostly 2 cores for so long, about 15 years ago. Simply put, it was very hard to program multi-threaded software that made use of all cores. Software like games did, as they needed all the available resources, but even the computations in MS Excel are mostly single-threaded as of now. Multi-threading was maybe used most for having a non-blocking user-interface. Even though OpenMP was standardised 15 years ago, it took many years before the multi-threaded paradigm was used for performance. If you want to read more on this, search the web for "the CPU frequency wall".

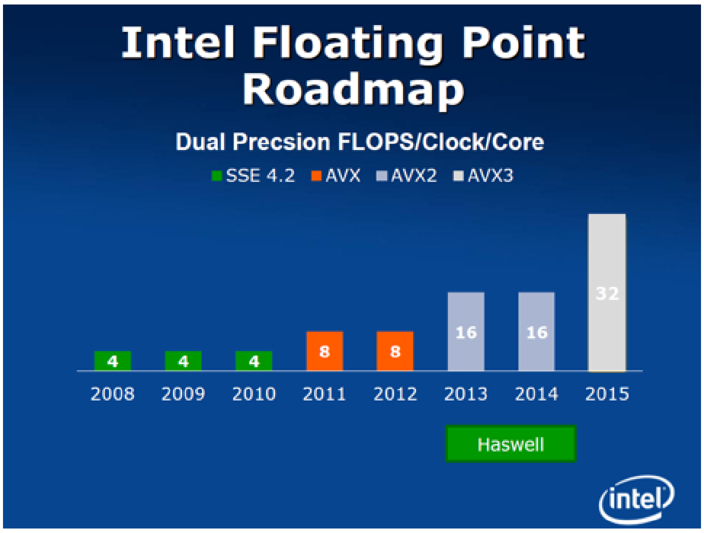
More interesting is what is happening now with CPUs. Both Intel and AMD are releasing CPUs with lost of cores. Intel has recently a 18-core processor (Xeon E5 2699-v3) and AMD was offering 16-core CPUs for a longer time (Opteron 6300 series). Both have SSE and AVX, which means extra parallelism. If you don't know what this is precisely about, read my 2011-article on how OpenCL uses SSE and AVX on the CPU.

AVX3.2

Intel now steps forward with AVX3.2 on their Skylake CPUs. AVX 3.1 is in XeonPhi "Knight's Landing" - see this rumoured roadmap.

It is 512-bits wide, which means that 8 times as much vector-data can be computed! With 16 cores, this would mean 128 float operations per clock-tick. Like a GPU.

The disadvantage is alike the VLIW we had in the pre-GCN generation of AMD GPUs: one needs to fill the vector-instructions to get the speed-up. Also the relatively slow DDR3 memory is an issue, but lots of progress is being made there with DDR4 and stacked memory.



So is the CPU turning into a GPU?

I'd say yes.

With AVX3.2 the CPU gets all the characteristics of a GPU, except the graphics pipeline. That means that the CPU-part of the CPU-GPU is acting more like a GPU. The funny part is that with the GPU's scalar-architecture and more complex schedulers, the GPU is slowly turning into a CPU.

In this 2012-article I discussed the marriage between the CPU and GPU. This merger will continue in many ways - a frontier where the HSA-foundation is doing great work now. So from that perspective, the CPU is transforming into a CPU-GPU; and we'll keep calling it a CPU.

This all strengthens my believe in the future of OpenCL, as that language is prepared for both task-parallel and data-parallel programs - for both CPUs and GPUs, to say it in current terminology.