CLOUD   MOBILE   GRAPHICS   CHIPS   SOFTWARE

Search this website…   SEARCH   LOGIN

CPUS   GPUS   SEMICONDUCTORS   STRATEGY   FORUMS

Intel's Sandy Bridge Microarchitecture

September 25, 2010 by David Kanter

Conclusions and Analysis

Sandy Bridge is one of the most ambitious and aggressive microprocessors designed at Intel. The degree of complexity and integration is simply astounding. It combines a new CPU microarchitecture, a new graphics microarchitecture, each of which is a substantial departure from the previous generation. On top of that, the chip level integration has taken a huge step forward; with a much more complex system agent and a new L3 cache and ring interconnect shared by all the components. Coherent communication between the CPU and GPU in Sandy Bridge is a substantial advance for the industry and presents many opportunities. Dealing with all these different facets of Sandy Bridge in a single discussion is impossible given the scope of changes.

Sandy Bridge is a fundamentally new microarchitecture for Intel. While it outwardly resembles Nehalem and the P6, it is internally far different. The essence of an out-of-order microarchitecture is tracking, re-ordering, renaming and dynamically scheduling operations to achieve the limit of data flow. Nehalem and Westmere rely on the same mechanisms that date back to the original P6. Sandy Bridge changes the underlying out-of-order engine and uses the more efficient approach taken by the EV6 and P4. That one change alone qualifies Sandy Bridge as a different breed entirely from the P6. But, there are changes in almost every other aspect of the design. The uop cache is a huge improvement for the front-end, largely by eliminating many of the vagaries of x86 fetch and decode. The implementation is quite clever and achieves many of the aims of the P4's trace cache, but in a far more efficient and reliable manner. AVX improves execution throughput and most importantly, the more flexible memory pipelines benefit almost all workloads.

In the coming year, three new microarchitectures will grace the x86 world. This abundance of new designs is exciting; especially since each one embodies a different philosophy. At the high-end, Sandy Bridge focuses on efficient per-core performance, while Bulldozer explicitly trades away some per-core performance for higher aggregate throughput. AMD's Bobcat takes an entirely different road, emphasizing low-power, but retaining performance. In contrast, Intel's Atom is truly intended to reach the most power sensitive applications. The two high-end microarchitectures, Sandy Bridge and Bulldozer, are shown below in Figure 7. Note that each Bulldozer module would include two integer cores while sharing the front-end and floating point cluster. Also, the floating point cluster in Bulldozer does not directly access memory, instead it uses the memory pipelines in the two attached cores, which then forward results to the FP cluster.

RELATED ARTICLES

- Sandy Bridge SPECcpu2006 Estimates
- Intel's Haswell CPU Microarchitecture
- Sandy Bridge ISSCC Update
- Sandy Bridge-EP Launches
- Sandy Bridge-EP Review

RWT ON TWITTER

- Looking forward to Haswell-based Surface Pro, it
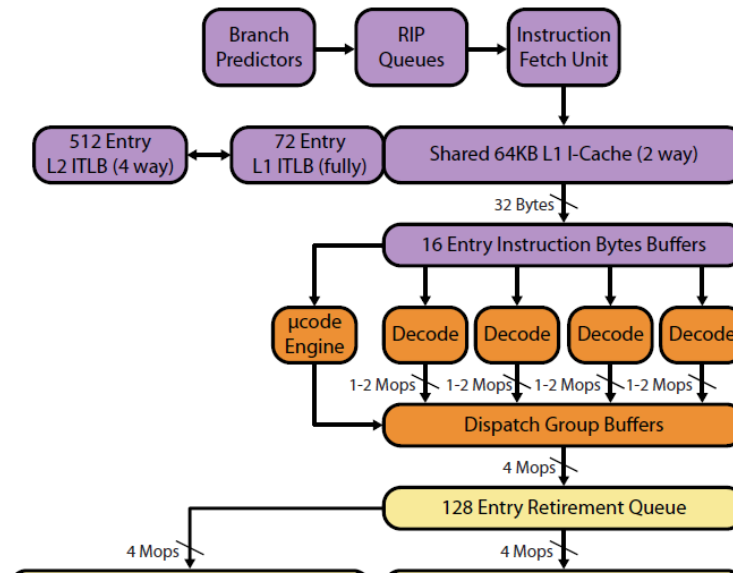
Sandy Bridge
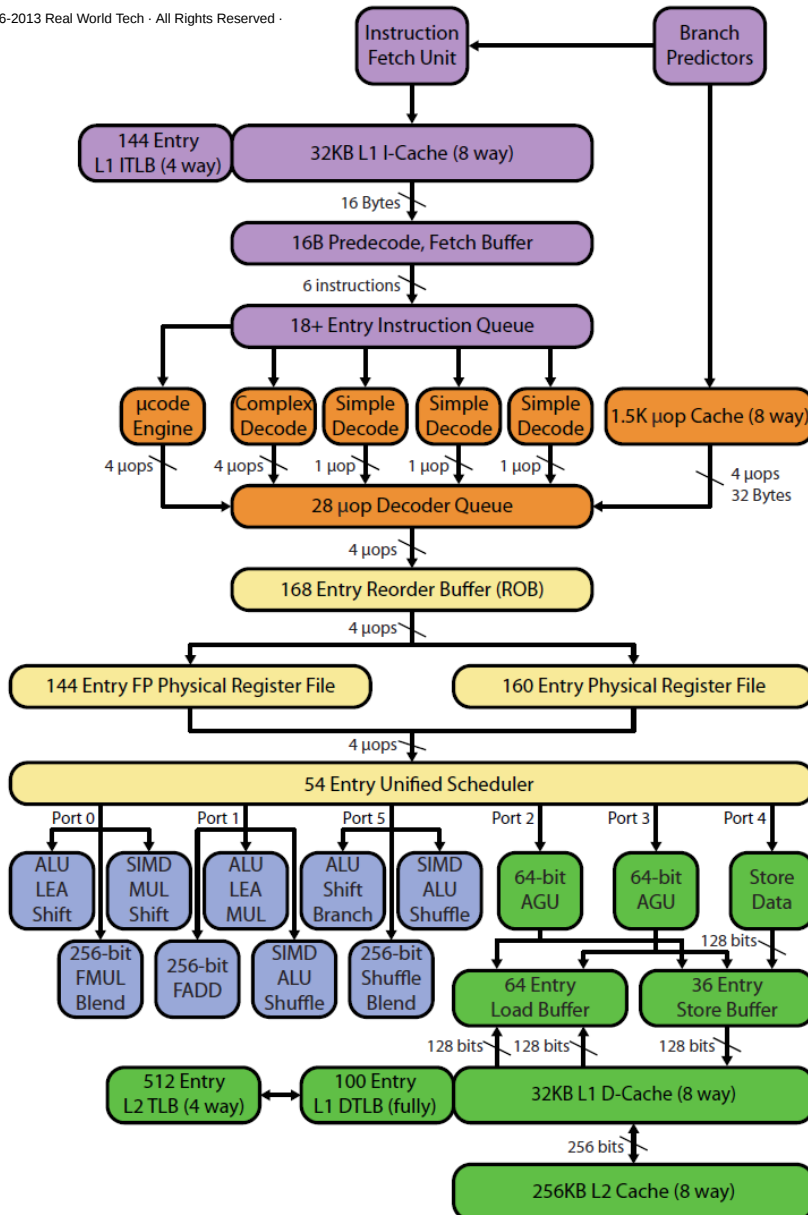
Bulldozer

Figure 7 – Sandy Bridge and Bulldozer Microarchitectures

With the limited details, it is hard to predict the chip level performance for products based on these two microarchitectures. Frequencies are still undisclosed, or have yet to be determined and the client and server products will be rather different. In the case of Sandy Bridge, the clock speed should be in the same vicinity as Nehalem or Westmere – however, Bulldozer is clearly intended to run faster, but the frequency will probably be dictated by power consumption. For Bulldozer, there are also numerous details on the integration (e.g. L3 cache design, snoop filter) that are undisclosed. Nonetheless, it is possible to make some educated estimates about the performance of the two microarchitectures.

In looking at the two designs, it is sensible to compare a multi-threaded Sandy Bridge core to a Bulldozer module and separately consider single threaded operation as a special case. Both support two threads although the resources are very different. At a high level, Sandy Bridge shares everything between threads, whereas Bulldozer flexibly shares the front-end and floating point units, while separating the integer cores.

A Sandy Bridge core should have substantially higher performance than a Bulldozer module across the board for single threaded or lightly threaded code. It will also have an additional advantage for floating point workloads that use AVX, (e.g. numerical analysis for finance, engineering). With AVX, each Sandy Bridge core can have up to 2X the FLOP/cycle of a Bulldozer module, although they would be at parity if the code is compiled to use AMD's FMA4 (e.g. via OpenCL). FMA4 will be relatively rare because, while elegant, it is likely to be a historical footnote for x86, supplanted by Intel's FMA3. For software still relying on SSE, the difference between the two should be minimal. In comparison, Bulldozer will favor heavily multi-threaded software. Each module has twice the memory pipelines and slightly more resources (e.g. retirement queue/ROB entries, memory buffers) than a single Sandy Bridge core with two threads, so Bulldozer should do very well in many highly parallel integer workloads that exercise the memory pipelines.

In many ways, the strengths of Sandy Bridge reflect the intentions of the architects. Sandy Bridge is first and foremost a client microprocessor – which requires single threaded performance. Bulldozer is firmly aimed at the server market, where sacrificing single threaded performance for aggregate throughput is an acceptable decision in some cases. Perhaps in future articles, we can examine the components of performance in greater detail (e.g. frequency, IPC, etc.), but for now, high level guidance seems appropriate – given the level of disclosure from both vendors.

Ultimately, we will be waiting for real hardware to see how the Sandy Bridge client performs in the wild. The base clocks, realistic turbo frequencies and power consumption will all be very interesting to observe – and help estimate server performance as well. For now the hardware certainly looks promising and while we await products, we'll have other reports on different aspects of Sandy Bridge to keep us occupied. The design team certainly deserves a round of congratulations for a job well done, redoing the microarchitecture from the ground up while tackling all the integration challenges.

Discuss (835 comments)