Inside 6th-Generation Intel Core: New Microarchitecture Code-Named Skylake & The Celerity Open-Source 511-Core RISC-V Tiered Accelerator Fabric

Yash Shah (ys562)

October 2024

1 Contributions of the Intel paper

Skylake focuses on higher power efficiency and operating frequencies. It is designed to operate at a wide dynamic power-performance range from 4.5W in passively-cooled, small form factor machines to 95W in high performance personal computers.

This has been made possible by a new power management system (Intel Speed Shift Technology Architecture and Algorithms) shifting more of the power management control from the operating system to the chip itself, improving not just efficiency, but also responsiveness.

The interconnect design is scalable with respect to both the number of Intel Architecture (IA) cores, and supports varying GPU configurations in accordance with power and cost. This scalability allows this series of chips to work across a wide range of thermal envelopes and I/O solutions.

2 Contributions of the Celerity paper

The Celerity project is aimed at the acceleration of rapidly emerging workloads. Developed in just nine months, core to the design is cost-effective, fast development through the use of open-source tools and designs; High-Level Synthesis (HLS); reuse; scalable, parameterizable, and modular components, and automated testing.

The design is aimed at meeting strict energy-efficiency and performance requirements, demonstrated through an accelerated implementation of a Binary Neural Network (BNN).

The major contribution of the paper is the Tiered Accelerator Fabric (TAF) which consists of three architectural tiers, each optimized for a different purpose:

- General-purpose tier $\rightarrow$ complex code.
- Specialization tier $\rightarrow$ energy efficiency + performance.
- Massively parallel tier $\rightarrow$ energy efficiency + flexibility.

These tiers are bound through a heterogeneous Remote Store Programming (RSP) model, where the cores and accelerators can write to each other's memories through a partitioned global address space. A novel Load-Reserved, Load-on-Broken-Reservation (LR-LBR) instruction pair is provided for efficient producer-consumer synchronization.

3 Power efficiency

Central to both designs is power efficiency. Skylake is a general-purpose design for varying workloads where there are not only differences between workloads, but dynamic requirements even as a single program executes. For example, a program may be memory-bound at one point, before becoming compute-bound at another. Celerity, in comparison, is designed for specialized workloads with strict energy efficiency and performance/latency requirements and known, fixed program execution characteristics.

Previously, the OS was responsible for managing performance and energy-efficiency by controlling CPU frequency and voltage through P-states reflecting changing program demands. The first problem with this approach is that the OS can only perform this every few tens of milliseconds to avoid being too intrusive. Secondly, the OS has limited visibility on the workload's instantaneous runtime behavior and microarchitectural characteristics.

Now, the OS can define energy performance preferences like the maximum allowed frequency and let the CPU handle power management with substantially higher update frequency and knowledge. Producer-consumer workloads see a substantial benefit, due to improved thread dependency observability. The OS may view threads waiting for each other as the CPU being underutilized, but lowering the P-state will prolong the producer thread, dropping utilization further. Since the CPU can view thread dependencies, it can selectively boost the producer thread. The higher update frequency improves responsiveness of the system as the turbo P-state can be toggled sooner in response to demand.

In comparison, Celerity is designed for tasks requiring consistent performance and with known execution characteristics. Celerity's Rocket cores have a shallow five-stage pipeline and importantly, the cores are in-order, single-issue processors. This provides superior energy efficiency due to omission of hardware to handle out-of-order execution or multiple instruction dispatch. Additionally, lack of this hardware means a reduction in die area as fewer transistors are required, reducing power draw from the leakage current from the 16nm FinFET transistors, as well as cost. Being in-order also means more consistent performance to suit the embedded application domain.

In contrast, another large part of Intel's efficiency gains are improvements to support its out-of-order architecture. More instructions are kept in-flight by a higher bandwidth instruction supply and more aggressive pre-fetching. Miss penalties for I-cache and stores are reduced to avoid stalls. Instruction Level Parallelism (ILP) is increased through deeper out-of-order engine buffers.

4 Use of parallelization and specialization

Skylake's graphics architecture is partitioned into Slice and Unslice. Slice is the massively-parallel, programmable, and scalable part, consisting of Subslices, each containing eight Execution Units (EUs). Similarly, Celerity's massively-parallel tier consists of tiles, each with a simple router and a Vanilla-5 core. Multiple Subslices and tiles help Skylake and Celerity achieve compute and memory scalability.

Additionally, Intel allows each slice, or pair of EUs to be turned on and off dynamically. If a smaller GPU configuration is enabled dynamically, a higher GPU turbo frequency can be achieved through Multiple Voltage-Frequency (V-F) curves and C-state selection. This is again a result of the flexibility with applications that the more general-purpose Intel design needs to cater to, unlike the more predictable tasks dealt with by Celerity.

Skylake and Celerity have more specialized hardware on offer for applications which have less flexibility, but require strict performance (and energy) guarantees due to their real-time nature (video playback/decode and image classification respectively). Skylake offers a pipeline of fixed functions in Unslice and 3D and media samplers within each Subslice with a limited degree of programmability. In comparison, Celerity's specialization tier was synthesized with HLS specifically to be a BNN and thus offers no programmability. Importantly, Skylake's specialized hardware is tailored for graphics and video playback in mind, since being a consumer chip, energy-efficient whole-day 4K video playback was a priority.

5 Fabric, cache, and memory architecture

Intel and Celerity have fundamentally different memory architectures, due to Celerity's focus on Remote Store Programming (RSP) [WAH09] to better support its massively parallel architecture. Intel utilizes a ring topology on-die interconnect, which is scalable with a variable number of Intel Architecture (IA) cores and different graphics configurations. While the Skylake cores have private caches, they all share the same global address space, with the fabric maintaining memory coherency and providing high-bandwidth.

In comparison, Celerity's RSP model assumes that loads are more common than stores, and thus partitions the global address space per core, such that each core is only able to read from its local memory, and provides a unified mesh Network-on-Chip to remotely perform stores in the local memories of other cores. This trades a relatively high store latency for a low load latency.

RSP allows Celerity to omit hardware support to maintain cache coherency and features such as Direct Memory Access (DMA) for the massively-parallel tier saving die area and power. Without support for remote loads and with fast local memory access, there is increased emphasis on locality of reference. Parallel algorithms with gather-like operations may cause programming challenges if cores cannot perform remote-loads.

Celerity also provides a novel Load Reserved, Load-on-Broken-Reservation (LR-LBR) instruction pair, which facilitates efficient producer-consumer synchronization across cores. The LBR instruction places the core's pipeline in a low-power state until another core remote stores to the reserved address (reserved by the LR instruction), breaking the reservation, waking up the core, and performing a load on the target address. This is great when a core needs to wait on another core e.g. a ready flag, or a pointer being sufficiently advanced to indicate space in a producer-consumer queue.

While Intel's memory architecture can potentially allow scaling to a few dozen IA cores, Celerity scales to hundreds of cores, but imposes a locality of reference restriction on algorithm design, trading general-purpose, simpler programmability for increased performance, at lower power consumption.

6 Future development and critique (Intel)

Skylake aims to improve power-efficiency, which affects thermals and battery life. A heterogeneous core architecture, similar to ARM's big.LITTLE could help significantly, where smaller, more power-efficient cores are used for the majority of tasks such as web-browsing, and larger, faster cores used for sustained compute.

This will also require significant work with advanced scheduling techniques to assign the correct type of core for each task, and smart movement of tasks between the two types of cores as the requirements of the program change throughout execution e.g. a shift from being memory to CPU-bound.

Skylake's GPU is predominantly accelerating graphics/video-related workloads. The GPU takes significant die area, but is idle in compute scenarios—even ones which can be easily parallelized. If the GPU could be used for compute and accessible to programmers, similar to how CUDA allows programmability of Nvidia GPUs, it could provide significant acceleration and power-efficiency gains for parallel workloads.

7 Future development and critique (Celerity)

To better support the RSP model, the paper mentions investigation into libraries to enable easier porting of CUDA-style applications. This is important as CUDA is often the standard in highly-parallel code. Additionally, a custom lowering pass could be written for Multi-Level Intermediate Representation (MLIR) to target this accelerator to enable easier porting of hardware-agnostic code.

Additionally, not all of the cores in the massively parallel tier may be necessarily (e.g. when there is substantial thread divergence between cores). Similar to Intel's GPU design, power gating at the core-level can be introduced to reduce the impact of leakage current and save power.

Another research direction could be fabricating a specialization layer for matrix-multiply units. Almost all deep learning architectures rely on matrix-multiply operations making it a great candidate for acceleration. A major down-side of Celerity's specialization layer is that it is not programmable. Since algorithms in the machine learning (ML) space iterate quickly, having programmable flexibility is crucial. This increased flexibility likely comes at a power cost, so it would be interesting to know whether this is justified.

Celerity SoC power estimates do not include DRAM power. Information about total power consumption is very important as DRAM will contribute significantly to the power usage in real-time ML applications where data is constantly streaming-in from DRAM.

Furthermore, the massively-parallel layer is only used to load the weights for the specialization layer. It seems wasteful that each core within the massively-parallel layer is a full Vanilla-5 core. An even lower power alternative could be chosen—perhaps something like a CUDA core which uses predication masking to avoid having expensive branching-related hardware. The paper mentions that the massively parallel layer is a "GPU-killer" without providing much justification. A performance and power-efficiency comparison between this and a desktop-class GPU would justify this claim.

References

[WAH09] David Wentzlaff, Anant Agarwal, and Henry Hoffmann. "Remote Store Programming: Mechanisms and Performance". In: (2009).