Tech Insights

MICROPROCESSOR report

Insightful Analysis of Processor Technology

CEVA XC20 ENABLES VECTOR-UNIT SHARING

Initial Model, the XC22, Targets 5G Infrastructure and High- End User Equipment

By Joseph Byrne (March 10, 2023)

Ceva's XC20 vector-DSP architecture implements a new simultaneous multithreading technique, allocating the resources of a single vector unit to two threads instruction by instruction. Vector-unit utilization increases, boosting the licensable DSP's area efficiency. Combined with other updates, Ceva reports the first XC20-based core, the dual-thread XC22, is more efficient than the single-thread XC4500 and the dual-thread XC16.

Sporting 128 multiply-accumulate (MAC) units, the XC22's vector performance slots it between the 64-MAC XC4500 and the 256-MAC XC16. The XC4500 divides its MACs between two vector computation units (VCUs), and the XC16 divides its resources among four VCUs. The XC22, by contrast, has a single VCU.

Although the XC16 and the XC22 both process two threads, having dedicated scalar resources and sharing vector computation units (VCUs), the XC22's approach is easier to harness and more flexible. Before, software could allocate a VCU to one thread or another at run time (see *MPR Mar 2020,* "Ceva XC16 Stays Ahead of 5G Rollout"). Now, however, software needn't explicitly assign a VCU to a thread; an arbitration unit in the DSP manages sharing.

Threads, moreover, can share resources in a VCU, whereas the XC16 allocated a whole VCU to a thread. Compared with the two threads and 128 MACs afforded by dual XC4500 instances, the XC22 has the added benefit that a single thread can potentially claim all 128 when necessary. Therefore, the XC22's sharing approach is more like the function-unit sharing in a dual-thread x86 CPU than in the XC16.

Unlike the XC16, a beefy DSP that has four VCUs and is best suited to 5G infrastructure, the XC22 targets a broader market: Ceva positions it for high-end user equipment (e.g., smartphone modems) and mobile infrastructure. For infrastructure, it would handle physical-layer and digital-front-end functions in distributed units (DUs) and radio units (RUs). Planned for 3Q23, the design's general availability will coincide with the freeze of 5G Advanced (3GPP Release 18). Ceva already claims a Tier One OEM as a licensee.

Conjoined DSPs Move 2 Kb per Cycle

The XC22 effectively comprises two scalar program-execution (PE) units—one for each thread—similar to Ceva's dual-core BX2 (see *MPR Feb 2019,* "Ceva's BX Hybrid Boosts DSP Engine"). Like a RISC CPU, the units implement a load/store architecture and have a 32-entry register file as well as a compiler-friendly nine-slot VLIW instruction set. Basic branch-prediction logic, including a branch-prediction unit, speed up code such as loops, as does a loop buffer—a common DSP feature. Each thread's scalar operations can employ two arithmetic units operating on 8-, 16-, 32-, 48-, and 64-bit integers and fixed-point values, along with an optional FPU handling half-, single-, and double-precision values.

A typical DSP, the XC22 is a Harvard architecture with separate instruction and data memories. The program side includes a cache and tightly coupled memory (TCM). Customers can instantiate the XC22 with up to 128 KB of cache and 128 KB of TCM per thread. A DMA engine can shuttle code into the TCM from off-chip DRAM or elsewhere on chip.

The data side includes four 128 KB (maximum) TCMs, each accessed by a 512-bit bus, as Figure 1 shows. As with the instruction TCM, a DMA engine can offload data transfers between the data TCM and memories outside the core. In addition to a DMA engine, a queue manager (QMAN) can perform data exchanges with hardware accelerators, offloading the cores (see *MPR Mar 2016,* "Ceva's New Gen-X DSPs Target 5G"). Because DSPs handle streaming data in real time, there is no data cache.

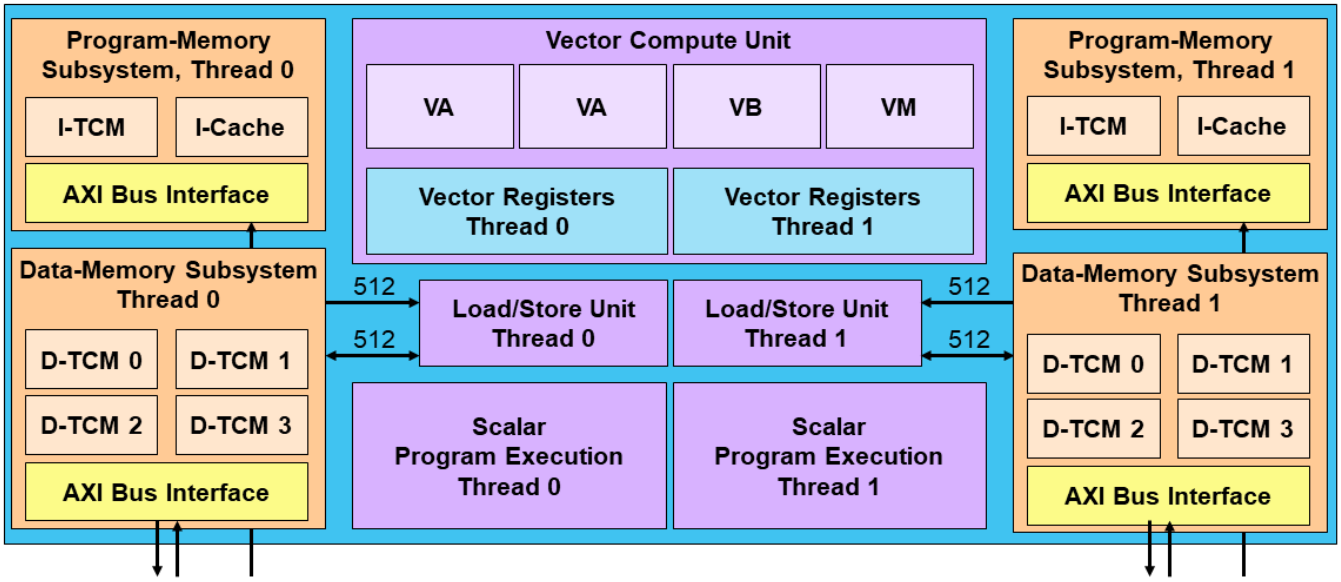

Figure 1. Ceva's XC22 DSP. The design integrates two scalar processors, each executing a thread, and a shared vector engine comprising four function units along with a separate register file for each thread.

Threading Vectors

Within the VCU, each thread has a 16-entry register file, eliminating vector registers from contention between threads. A thread can perform two loads or one load and one store per cycle, transferring a pair of 512-bit chunks at a time between two TCMs and two 512-bit vector registers. Meanwhile, the DMA engines can transfer data to and from the two TCMs unoccupied with load/store tasks.

The registers feed the VCU's four execution units, each operating on 512-bit vector operands. Two vector arithmetic (VA) units each contain 64 INT16 MAC blocks. For a thread to fully utilize all 64, blocks must share operands, which they do when multiplying complex numbers. The vector bit and nonlinear-operation (VB) unit performs logic and nonlinear operations, such as square root, inversion, and trigonometric functions. The vector move (VM) unit performs moves and shifts.

A dynamic vector-unit arbitration block (not shown) mediates access to the execution units. Each thread can issue operations to multiple units per cycle. If the threads require different units, each gets the requested resources. Otherwise, one must wait. If contention never occurred, vector performance per unit area would be twice that of two separate VCUs.

In practice, Ceva has found that performance per unit area for the VCU alone increases 1.7x even on vector-heavy kernels, which it developed in conjunction with customers. Improvements in memory management, the scalar unit, and vector instructions push the typical gain for the entire XC22 DSP to 2.5x compared with the XC16. Relative to the XC4500, efficiency is 1.8x greater, and performance rises 2.3x. The new DSP is also more power efficient. Multiplied by the many DSPs per chip in an infrastructure design, even modest efficiency gains would yield substantial area and power savings.

Ceva rates the XC22 at 1.6 GHz in a 7 nm process compared with the XC16's 1.8 GHz. The XC22 can execute the same vectorized C code that would target the XC4500, as the dynamic VCU-resource allocation is transparent.

Ceva expects some customers to license the XC22 on its own. Others will employ it as part of the company's PentaG2-Max and PentaG-RAN basebands for user equipment and infrastructure (see *MPR Mar 2022,* "Ceva's PentaG2 Reduces 5G Power," and *MPR Oct 2022*, "PentaG-RAN IP Targets Base Stations").

Area Efficiency Differentiates XC20

Ceva faces no direct competition for licensable, off-the-shelf vector DSPs intended to serve mobile infrastructure. For this small market, alternatives include proprietary DSPs and adaptation of Cadence's Tensilica Xtensa core. The Tensilica ConnX B20 comes close, integrating a 512-bit vector engine, but it's only a five-slot VLIW design. It also doesn't target infrastructure but client devices. While the XC22 targets high-end 5G modems, Ceva's other DSPs may be a better fit for such devices.

Ceva has refined its vector architecture over multiple iterations. The fifth-generation XC20 substantially improves multithreading, making it transparent to software and increasing granularity by sharing resources in a VCU. Vector units improve performance on many workloads, but no software can fully utilize them all the time. Sharing vector resources among threads improves their utilization, substantially increasing area efficiency.

Counting Nokia, ZTE, and small-cell startup Picocom among its customers, Ceva optimizes its DSPs to meet the needs of mobile-infrastructure companies. And by developing the dynamic allocation of vector function units, it has improved the XC22's area efficiency, reducing customer manufacturing costs. As the company scales the technology to DSPs with more or wider VCUs, the cost savings will increase, helping it secure wins in higher-performance designs. ♦

Price and Availability

Ceva expects general XC22 RTL availability in 3Q23. The company withheld licensing fees. For more information, access the *XC22 product page.*

