The Linley Group

MICROPROCESSOR report

Insightful Analysis of Processor Technology

BLAIZE IGNITES EDGE-AI PERFORMANCE

El Cano Processor Runs Yolo v3 at 50fps, Consumes Less Than 7W

By Mike Demler (September 7, 2020)

Ten-year-old edge-AI startup Blaize is sampling board-level systems that integrate its El Cano inference processor. The chip integrates 16 of the company's graph-streaming processors (GSPs), delivering peak throughput of 16 trillion operations per second (TOPS). El Cano uses dual Cortex-A53 CPUs to run an operating system and application software, but the programmable GSPs handle all neural-network operations independently, as well as image signal processing, sensor fusion, and other tasks. The first products target commercial and enterprise computer-vision systems, such as retail analytics and antitheft systems. El Cano is also well suited to automotive and industrial applications, including camera/lidar sensor fusion and robotics.

The new processor, named for the often overlooked Spanish explorer who completed the first circumnavigation of the Earth after Magellan died at sea, improves hardware utilization by combining data, instruction, and task parallelism. It can dynamically manage multiple workloads according to their bandwidth and compute requirements, in addition to supporting conditional processing. The chip's hardware scheduler runs multiple neural networks in parallel or sequentially, or it can stagger execution to optimize use of available resources. In one of Blaize's demos, El Cano ran five independent Yolo v3 networks on five separate HD-video streams, delivering inference results on 416x416 subframes at 10fps. It can also run a single copy at 50fps. The startup plans to begin volume production next quarter. Samsung manufactures the chip in 14nm technology.

Blaize began operations as ThinCI in 2010, but it remained in stealth mode until the 2017 Hot Chips conference, where it presented its 28nm test chip. Founders include CEO Dinakar Munagala, CTO Satyaki Koneru, and Chief Scientist Ke Yin, each of whom previously worked as a GPU architect at Intel. Koneru worked at Nvidia as well. The company initially positioned its GSP technology for mobile devices, but after attracting strategic investments from Tier One suppliers and OEMs, it shifted its focus to include automotive and industrial systems. Denso, Daimler, Magma, and the Mirai Creation Fund (Toyota) are among its investors, along with Samsung and several venture-capital funds. In November 2019, the startup's most

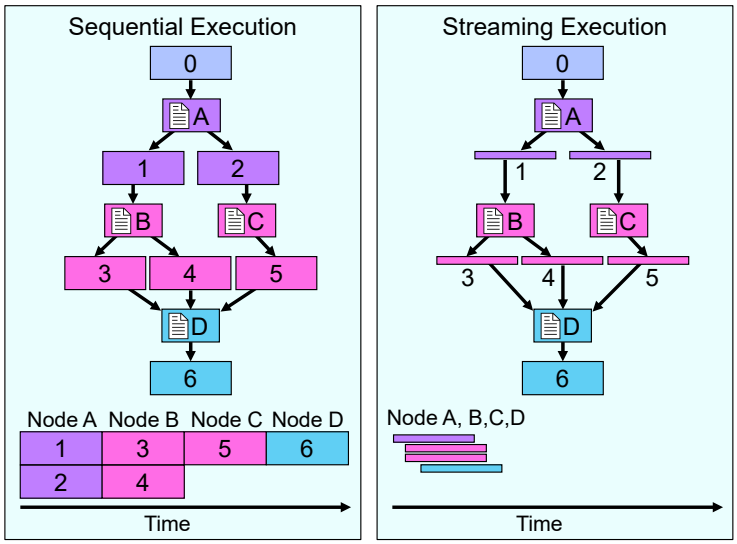

Figure 1. Sequential and streaming neural-network execution. On the left, a TensorFlow-based neural network executes as a series of kernel operations one node at a time. By contrast, Blaize's graph-streaming technology allows multiple nodes to execute in parallel, consuming data as soon as it's produced and thereby increasing throughput as well as reducing on-chip memory requirements.

Price and Availability

Blaize is sampling the El Cano processor and plans to start volume production in 4Q20. The Blaize Xplorer X1600E card sells for $299, the industrial-grade Pathfinder P1600 SoM for $399, and the Blaize Xplorer X1600P card for $999 (all in volume quantities). More information on these products is at *www.blaize.com/products*. For product-demo videos, point your web browser to *www.blaize.com/resource-center*.

recent $65 million funding round brought the total investment to $87 million. Blaize is headquartered near Sacramento, California, but most of its 325 employees work in Hyderabad, India.

A Master of Multitasking

Most deep-learning accelerators (DLAs) employ some form of data-flow architecture to handle large matrix operations, such as the systolic multiply-accumulate (MAC) arrays Google popularized in its tensor processing units (see *MPR 8/31/20,* "Google Details TPUv3 Architecture"). Although that technique saves execution time and power by allowing intermediate results to flow directly from one stage to the next, the size of the data structures is often a poor match for fixed-function hardware. Vendors typically specify theoretical peak performance, but most inference engines run at less than 50% of that number on real-world tasks.

Standard neural-network frameworks such as TensorFlow exacerbate the problem because the graphs they produce employ an entire tensor as the smallest unit of data flow from one kernel operation (such as matrix multiply) to the next. Most neural-network engines only schedule one node and its associated operations at a time, leaving execution units unused. Figure 1 shows an example for three neural-network layers. For the execution sequence on the left, TensorFlow specifies the kernels for Nodes A, B, and C. Because only one kernel runs at a time, Node C must wait until Node B has completed its operations, even if Node A has already stored results in Buffer 2.

To avoid these delays, DLA vendors must create parallelizing compilers that rearrange the TensorFlow graph to execute efficiently on their hardware. Blaize's Picasso software tools analyze a neural-network graph to extract any data dependencies, fracturing each node into smaller block operations. El Cano's hardware scheduler then dynamically distributes the operations to multiple execution units. The company extended OpenVX to take advantage of its task-graph descriptors, adding OpenCL extensions that allow developers to program the chip using C/C++.

El Cano implements fine-grain task-level parallelism based on a producer-consumer model. Rather than wait for the entire matrix multiplication at Node A to finish, for example, it executes smaller block operations that enable Nodes B and C to commence execution as soon as data is available. The GSPs allow instruction-level parallelism, increasing throughput and reducing power compared with conventional data-flow architectures. The technique also saves die area by reducing on-chip-memory requirements. Because execution units consume data as soon as it's available, less temporary storage is necessary.

Other accelerators based on GPUs and VLIW architectures employ instruction and thread parallelism, but the GSP technique is different. In a VLIW design, the compiler-determined thread assignments are static, and they ignore other threads running in parallel. El Cano's thread-scheduling process is dynamic. By scheduling operations on the basis of graph dependencies, the GSPs can look ahead to prefetch data and change context in a single clock cycle. They also support conditional execution, such as retargeting a camera or lidar sensor to acquire a more accurate image of detected objects. This approach is similar to that of AI-specific manycore architectures from companies such as Graphcore and Tenstorrent.

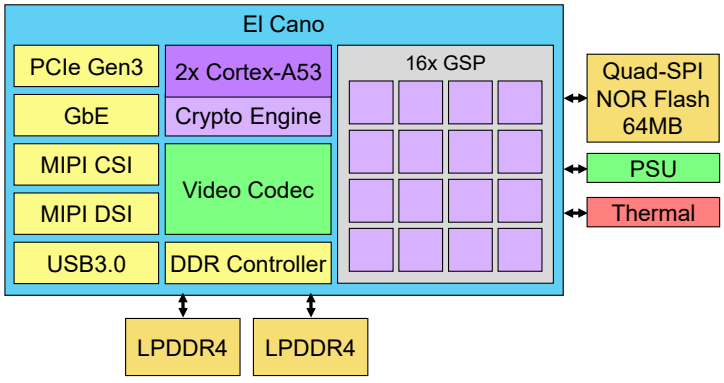

Figure 2. Blaize Pathfinder system-on-module. The processor has 16 of the company's graph-streaming-processor (GSP) cores, which deliver 16 TOPS total. An integrated scheduler distributes the workload by fragmenting the computational graph into smaller subtasks, enabling fine-grain task and thread parallelism.

Inference Plus Sensor Fusion

Blaize offers El Cano in two form factors. The first is the Pathfinder P1600, a credit-card-size system-on-module (SoM) that uses the Cortex-A53s for standalone operation. The second, the Xplorer X1600–series plug-in cards, instead work as coprocessors. The X1600E ships on an EDSFF (enterprise and datacenter SSD form factor) card designed to plug into 1U servers. The X1600P is a half-height half-width PCIe card that can connect up to four El Cano chips. Including the processor, DRAM, and support components, each single-chip El Cano product has a 7W typical power rating. The company backs the lineup with a comprehensive software suite comprising the Picasso SDK for C++ programmers, AI studio for developing applications in a GUI environment without

having to code, and a Netdeploy tool that enables developers to optimize their models through pruning and quantization.

The El Cano processor integrates two Cortex-A53s running at up to 1.0GHz, each with 32KB L1 and 512KB L2 caches. As Figure 2 shows, the CPUs work with an Arm crypto accelerator to guarantee secure operation. The chip offers 2x32-bit LPDDR4 interfaces that support up to 16GB/s of total bandwidth.

The GSPs integrate unified data pipelines that support up to 64-bit integer operations along with nonstandard 8-bit floating point; half, single, and double precision; and Google's Bfloat16 format. The higher-precision floating-point capabilities enable lidar and radar signal processing.

Although the chip is well suited to object-recognition and scene analysis in camera-based systems, the Picasso software also compiles image-processing kernels such as color correction and noise reduction to run on the GSPs. By using the GSPs rather than a fixed-function ISP, El Cano gives customers greater flexibility to configure it for different applications and to future proof their systems. Processing raw data can increase accuracy, and programmability allows image-processing-function changes at run time. El Cano's graph framework can run custom layers, such as for sensor fusion or general-purpose compute tasks.

The processor includes a video codec that can encode or decode a single 4K video stream at 30fps or handle multiple smaller streams at the same rate. The MIPI CSI interface provides four receive channels and a single transmit channel. El Cano also integrates a MIPI DSI display interface along with standard networking interfaces for GbE, PCIe, and USB3.0. The Arm cores run TrustZone, working with the crypto engine to support secure boot from flash memory connected to the quad SPI. For automotive and industrial systems, the SoM adds a three-lane CAN interface and a variety of serial interfaces.

Streaming Through the Quads

El Cano's top-level controller dynamically manages graph execution, as Figure 3 shows. It works with the thread scheduler to implement the parallel, sequential, and staggered workload-management schemes, and it also handles conditional execution. Blaize compares its task scheduling to the scatter/gather techniques that CNNs frequently use to access image data (see *MPR 10/12/15,* "Cadence P5 Boosts Embedded Vision"). As in the data path, the hardware controller and the thread scheduler can fracture (scatter) or aggregate (gather) instruction threads to maximize performance. As a result, the processor can simultaneously execute instructions for multiple neural-network nodes, increasing hardware utilization and throughput.

The DMA- and execution-command rings queue threads for distribution to the GSPs and multilevel cache system. The L2 caches comprise a total of 4MB. The so-called read-only caches hold immutable data written from higher execution-graph levels—weight parameters, for example. The data structures include dependency information, which enables the scheduler to manage the thread-execution sequence along with up- and downstream data flow. On each clock cycle, the thread scheduler can pick from 64 instruction threads to populate up to 24 independent pipelines, including those for custom functions, flow control, memory operations, scalar and vector math, and state management. Threads can spawn other threads, but they always flow back to the scheduler to check dependencies.

Each GSP runs multiple threads in parallel on four sets of quad processors. Every quad integrates four identical cores connected to an arbiter, and every core has an instruction scheduler, thread-state memory, and two execution units: a multiprocessing SIMD unit (MPU) and a scalar processing unit (SPU). Although Blaize calls it a scalar unit, the SPU is actually a narrower (but unspecified) SIMD unit than the MPU. Each quad also includes one shared special-function unit (not shown in the figure) that executes histograms, median filters, and similar DSP operations. The El Cano architecture implements fine-grain task parallelism by enabling threads for multiple neural-network nodes to execute in parallel on multiple cores in a quad, multiple quads in a GSP, and multiple GSPs in the chip.

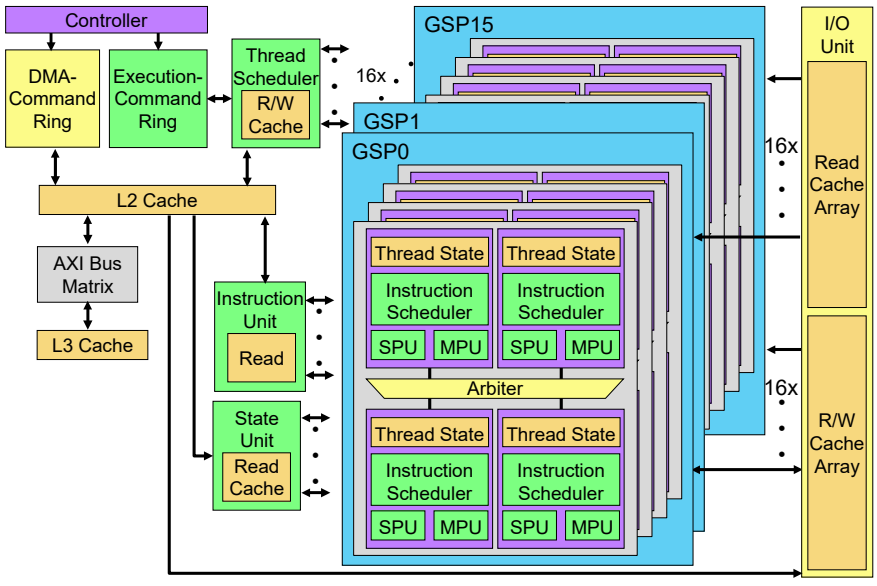

Figure 3. Sequential and streaming neural-network execution. On the left, a TensorFlow-based neural network executes as a series of kernel operations one node at a time. By contrast, Blaize's graph-streaming technology allows multiple nodes to execute in parallel, consuming data as soon as it's produced and thereby increasing throughput as well as reducing on-chip memory requirements.

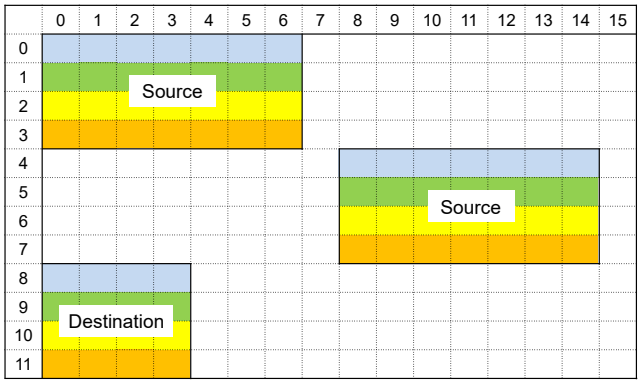

Figure 4. El Cano 2D register-file operations. To accelerate matrix-math operations, the GSP cores execute block operations, reading and writing arbitrarily aligned arrays. This example multiplies a 4x7 and 7x4 matrix to form a 4x4 matrix.

Block-Based Processing Increases Efficiency

El Cano's MPUs can perform operations on unaligned 2D data blocks in configurable register files, such as block move and matrix-multiplication reduction using smaller dot products. The physical registers hold 512 bits, which are dynamically configurable as a vector or a 2D array. All of the persistent data structures are block based. To execute a block operation, the MPUs will automatically iterate, accessing the register file across a variable number of rows and columns, as Figure 4 shows.

The MPUs execute 2D operations, such as matrix multiplication. But unlike most DLAs, they omit systolic arrays of MAC units. Rather than allocate a fixed-size hardware unit for MAC operations, typically yielding poor utilization, the graph compiler fractures MAC-intensive convolutions into smaller dot-product operations that the scheduler distributes for parallel execution. In each cycle, one INT8 dot-product instruction executes a 3x3 convolution, comprising nine multiplications, eight additions, and a register accumulation. By virtue of this flexibility, El Cano can execute MAC operations without leaving any multipliers idle waiting for data.

Using its 16 MPUs, each GSP can perform 64 INT8 dot-product instructions per cycle, producing 0.92 TOPS at the chip's 800MHz clock speed. Blaize includes additional MAC operations from the SPUs to calculate the peak performance of 1 TOPS per GSP or 16 TOPS for the El Cano chip. This performance scales with the width of the data being processed, so the chip can achieve 32 TOPS on INT4 data or 2 TOPS on FP64 data. For any given width, the MPUs handle integer and floating-point data at the same rate.

A New Contender in Automotive

El Cano is well suited to computer vision in a wide range of edge devices, and the initial system products target commercial and enterprise applications. Because Blaize's lead strategic investors are from the automotive space, however, and because it has an automotive-grade product on its roadmap, we compare El Cano with other low-power ADAS processors.

The processor's flexibility and power efficiency are most similar to the Hailo-8 processor, which also targets ADASs and autonomous vehicles (see *MPR 6/24/19,* "Hailo Illuminates Low-Power AI Chip"). Both devices will compete with Mobileye's EyeQ5, the most recent offering from the ADAS leader (see *MPR 12/9/19,* "Mobileye Expands Into Robotaxis").

As Table 1 shows, these three chips offer similar peak TOPS per watt, but in practice, their performance depends on the workload and hardware utilization. Blaize specifies the El Cano PCIe card and module at 7W; we estimate the chip on its own requires about 5–6W, similar to EyeQ5, offering a peak power efficiency of 2.7 TOPS per watt. Actual efficiency will be less. For example, running Yolo v3 at 50fps is equivalent to 3.4 TOPS, but because El Cano's GSPs also handle image processing and other tasks, we estimate the chip will supply around 1.0 TOPS per watt for that application.

Mobileye has yet to publish any neural-network benchmarks, so we're unable to derive EyeQ5's real-world power efficiency. Hailo consumes 1.7W when running ResNet-50 at 672fps, which requires about 5.1 TOPS—about 20% of the chip's compute resources. It calculates Hailo-8's power efficiency for that workload to be about 2.8 TOPS per watt. When running a more compute-intensive task such as Yolo v3, its efficiency will probably be greater, although the company withheld measurements. Hailo-8 integrates a Cortex-M4 controller, but because that CPU lacks the ability to run Linux, we estimate total system power consumption at peak performance will be about 9W.

	Blaize El Cano	Hailo Hailo-8	Mobileye EyeQ5
Main CPU	2x Cortex-A53	1x Cortex-M4	4x MIPS64 Warrior*
CPU Speed	1.0GHz	Undisclosed	Undisclosed
On-Chip Memory	4MB L2 cache	32MB SRAM*	3MB L2 cache
DRAM Interface	2x 32-bit LPDDR4-2133	None	4x 32-bit LPDDR4-2133
Camera Interface	4x MIPI	2x MIPI	16x MIPI CSI
Image Processor	4K @ 60fps	Undisclosed	Undisclosed
I/O Interfaces	1x CAN, GbE, PCIe Gen3, USB3.0	GbE, PCIe	3x CAN, GbE, PCIe Gen4
Max AI Perf	16 TOPS	26 TOPS	12 TOPS
Power (typical)	6W*	9W*	5W
Efficiency	2.7 TOPS/W	2.8 TOPS/W	2.4 TOPS/W
IC Process	14nm	16nm*	7nm
Production	4Q20	1Q20	4Q20

Table 1. Comparison of ADAS processors. These chips employ three different architectures, but they provide similar power efficiency. El Cano is unique for its ability to run image-processing tasks as well as fuse camera and lidar data. (source: vendors, except *The Linley group estimate)

EyeQ5 integrates four dual-thread MIPS cores, delivering the highest CPU performance in this group. It also supports a full set of surround cameras, whereas Hailo-8 supports stereo cameras. Using the chip's GbE interface, Blaize has demonstrated El Cano running five Ethernet-connected cameras. Mobileye's product offers twice the DRAM bandwidth of El Cano, but the latter needs less for its graph-streaming model. Hailo-8 omits a DRAM interface. It can handle an entire ResNet-50 model in on-chip SRAM, but Yolo v3 processing requires multiple processors.

In the Pathfinder P1600 system, El Cano has fused cameras and lidar, computing an occupancy grid by projecting the range information from the point cloud on images from four HD cameras. EyeQ5 and Hailo-8 must derive depth maps using stereo images and structure from motion, likely increasing safety-critical response time compared with El Cano's sensor-fusion approach.

Patiently Gaining an Edge on Competitors

In their eagerness to win a piece of the fast-growing market for edge-AI processors, many startups prematurely launch products without first developing a complete hardware and software package. By contrast, Blaize patiently spent almost 10 years refining its graph-streaming technology. In 2017, it announced a 28nm test chip, which enabled it to optimize the architecture, prototype its PCIe cards and modules, and complete the software tools that are critical to using its architecture. The test chip also provided the necessary proof of concept to attract potential customers and funding for the 14nm version.

Blaize's Picasso software-development platform lets customers program the chip through standard C/C++ tools, and it handles common machine-learning frameworks such as Caffe, Pytorch, and TensorFlow. Developers can also employ the custom-kernel compiler to build proprietary neural-network models or run other functions on the GSPs. Customers can avoid programming altogether by employing the Blaize AI Studio, which works with the Netdeploy tool and a library of precompiled network models. Netdeploy includes pruning and quantization features that enable accuracy/performance tradeoffs. Picasso's AI tool kit provides computer-vision and deep-learning libraries, as well as a linear-algebra library that supports the GSPs' DSP functions.

El Cano is an edge-AI processor that excels at multitasking. It's the first announced product that can simultaneously run multiple Yolo v3 models without needing a massive power supply and many gigabytes of DRAM. El Cano offers flexibility, performance, and power efficiency for a variety of edge-related markets. Its ability to fuse lidar and camera images makes it ideal for automotive, and its ability to do conditional processing enables more-precise object recognition than other inference processors can achieve. It's a good fit for industrial, retail, smart-city, and computer-vision systems that analyze inputs from multiple cameras, but its modules are too expensive to put in each camera. By taking time to develop its first products, Blaize avoided burning through cash too quickly. We expect its patience will pay off. ♦