Tech Insights

MICROPROCESSOR report

Insightful Analysis of Processor Technology

AnDLA Debuts For AI Acceleration

Andes CNN Accelerator Tightly Connects to Its CPUs

By Bryon Moyer (June 9, 2023)

Andes' new deep-learning accelerator addresses convolutional neural networks in edge applications. Accompanied by vector CPUs, it forms an AI subsystem that can be scaled up for higher vision- and audio-workload performance.

Capable of 8 TOPS peak AI performance, the AndesAire AnDLA unit runs autonomously once set up, allowing control and other CPUs to perform additional tasks in parallel. It's a simple core, with no bells and whistles, handling only INT8 data—at least in this first edition. Software support is basic, but it provides flows both for SoC developers putting fixed functions onto their chips and for end users with TensorFlow Lite for Microcontrollers as a model interpreter. Pairing with a vector CPU provides a fallback for nonlinear or other functions that AnDLA currently lacks but may support in the future.

Andes is better known as a provider of configurable CPU blocks—more recently pivoting from its own instruction set to RISC-V. In planned upgrades, the company's custom-instruction capability will allow the accelerator to implement custom operations, with a tightly coupled interface for data exchange that's more efficient than using the system bus.

AnDLA is available now for licensing by early adopters; the company has scheduled production RTL for 4 Q23. By that time, it will be years behind other providers of inference acceleration, which include single-product startups as well as companies offering both CPU and accelerator intellectual-property cores.

The benefit to Andes' CPU customers is an accelerator designed and tested to integrate easily with Andes' CPUs. It's most likely to succeed in small systems where convolutional neural networks (CNNs) are popular, including the upper end of those falling under the TinyML umbrella—applications running on microcontrollers. Typical applications include simple presence detectors and cameras used for factory automation or security.

Optimized for CNNs

Andes optimized its AnDLA intellectual-property (IP) block for convolutional neural networks. Two hardware blocks implement the convolution function: a general matrix-multiplication engine (GEMM) and an accumulator that adds partial sums from the GEMM block, as Figure 1 shows. The GEMM comprises 32–4,096 MACs; it supports only INT8 at present, with other

formats planned in future releases. An elementwise processor implements activation functions, and a separate block handles pooling (maximum or average). Built-in activations include ReLU, leaky ReLU, sigmoid, tanh, ReLU6, and SiLU.

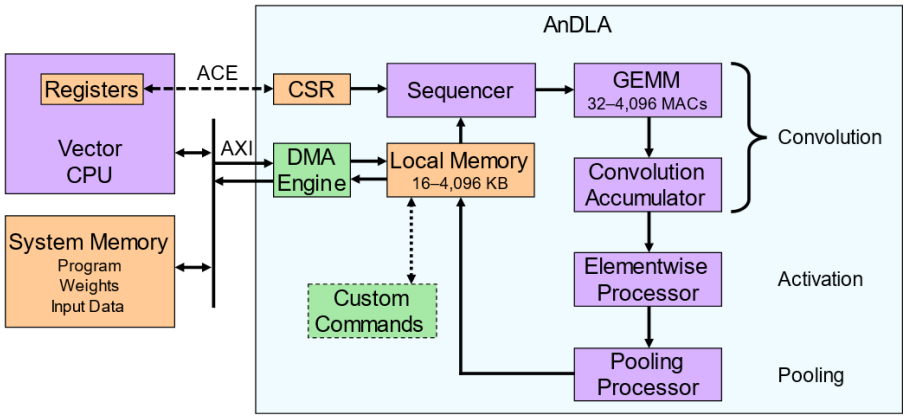

Figure 1. AnDLA convolution flow. CSR=control and status register, ACE=Andes Custom Extension. The general matrix-multiplication block (GEMM) and the accumulator perform convolution; the elementwise processor handles activation prior to pooling.

The program (or command stream) remains in system memory. Prior to executing a workload, the host CPU loads pointers to the program and the data (including a pointer to the location for results) into a control and status register (CSR). Once triggered, the DMA engine loads input data into the local memory, the size of which can range from 16 KB to 4 MB; afterward, the DMA writes blocks back to system memory. AnDLA executes autonomously until the program completes, at which point it interrupts the CPU. Data transfer occurs over the AXI port connecting the AnDLA core to the system bus.

The GEMM also computes fully connected layers. Although a principal use of the elementwise processor is for activations, it can serve for any elementwise function. A developer may also create custom commands using Andes' custom-instruction capability. In this case, the custom logic resides in the DLA instead of the CPU, triggered by custom commands in the program. This arrangement allows the DLA to operate independently of the CPU even with custom commands.

Convolution windows top out at 15 × 15 in the default configuration. The engine breaks the full image frame into 1,023 × 1,023 tiles. These limitations come from the registers storing the sizes (4 bits for the window and 10 for the tile); those can be configured to different numbers.

Andes conceives a processing element as an AnDLA block paired with a CPU supporting vectors, as depicted in Figure 2. The CPU's scalar and vector engines can implement pre- and postprocessing functions as well as any other functions not implemented in the AnDLA block, like softmax (hardware support of which the company plans in the future). The CPU also serves for aspects of an algorithm that may still be in flux, unlike the well-established functions already hardened in the DLA.

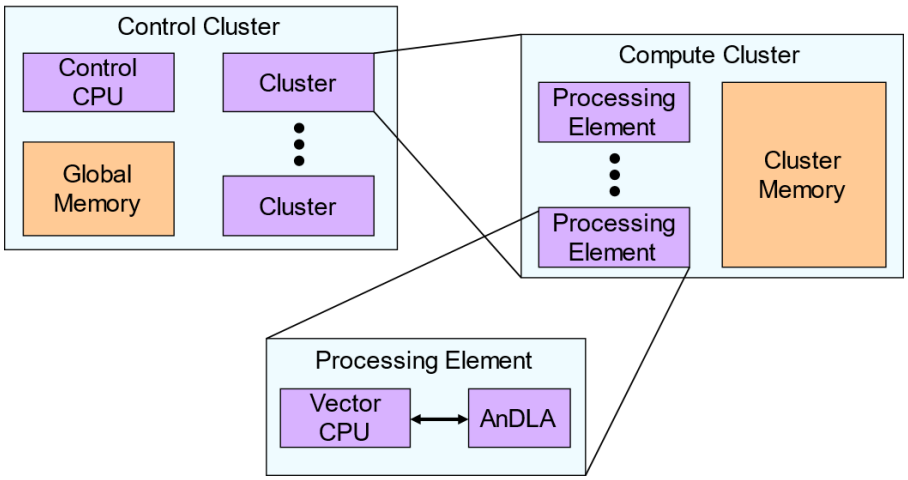

Figure 2. Scaling CNN computing. The DLA block pairs with a CPU containing a vector engine. In future versions, multiple instances of that pair, along with memory, form a compute cluster. A top-level control cluster can contain multiple compute clusters. The cluster memory may act as the system memory shown in Figure 1.

To increase performance in a future upgrade, a licensee can replicate processing elements and add a shared memory, forming a compute cluster. That cluster can replicate in a higher-level control cluster containing a control CPU and global memory. Although the TOPS metric will scale linearly with replication, actual performance scaling remains unknown.

To C or Not to C

Andes has two development-flow options: one targeting the end users of an SoC integrating AnDLA and one targeting makers of such an SoC. The former flow compiles models to run on TensorFlow Lite for Microcontrollers (TFLM), and it maximizes flexibility because an updated model works with existing firmware. The latter implements a C program for efficiently executing models that are less likely to require updates. Once compiled down to C, a model update changes the firmware.

The development flows are identical, differing only in the final output. The tools accept models in PyTorch, TensorFlow, or ONNX, as Figure 3 shows. An optimizer performs network pruning and fuses operations where feasible. The following block performs compensation (which attempts to reduce the dynamic range of weights in a layer by redistributing them between layers) as well as calibration and quantization. The final optimizer allocates memory and operators, tailoring the implementation for the specific platform. It generates either a C version of the model or one that can run on TFLM.

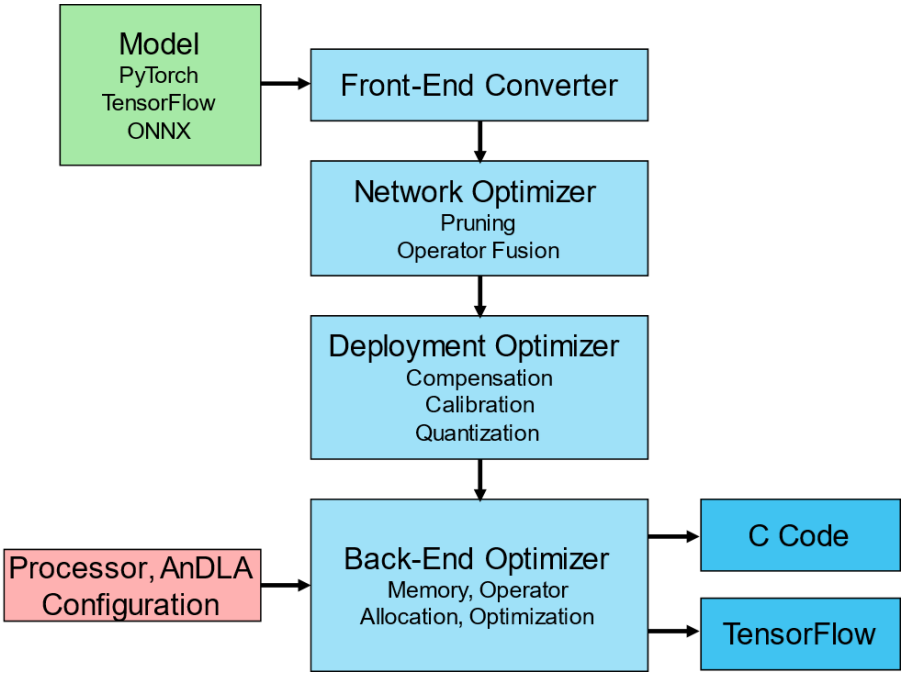

Figure 3. AnDLA development flow. The final output can be C code for a bare-metal execution environment or a model suitable for running on TensorFlow Lite or TensorFlow Lite for Microcontrollers.

The run-time environment is more complex, with much of the stack running in the control cluster, which delegates to the compute cluster. Typical execution assumes a Linux operating system running TensorFlow Lite (TFL), with a delegate layer dispatching commands either to a driver, which forwards them to the compute cluster, or to a stub for the run-time library, as Figure 4 illustrates. The library itself resides in the compute cluster, communicating via remote procedure calls (RPC).

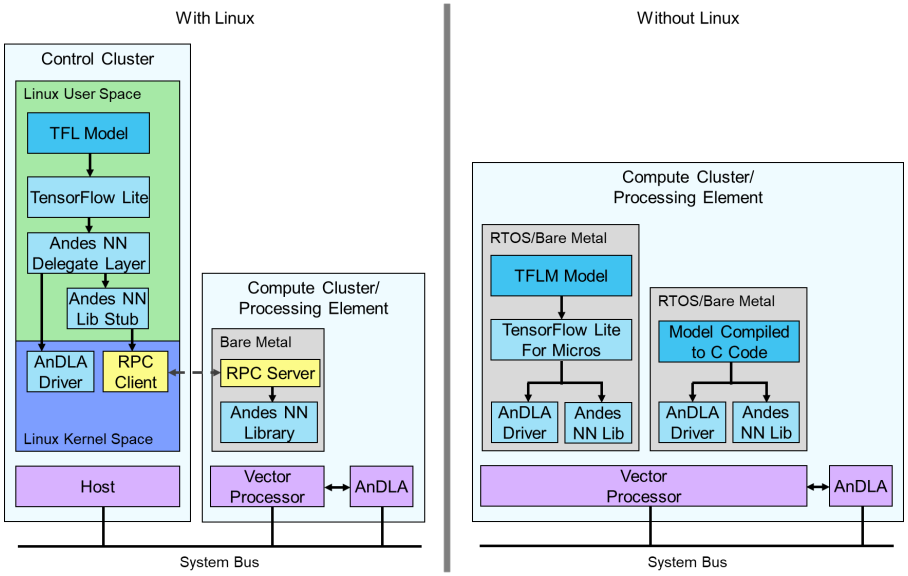

Figure 4. Andes' AI run-time environment. TFL=TensorFlow Lite; TFLM=TFL for Microcontrollers. NN=neural network; RPC=remote procedure call; Lib=library. Typical user models will run on TFL (in a system running Linux) or TFLM in the control cluster, dispatching commands to the compute cluster. SoC developers may implement built-in models using the C flow.

For smaller bare-metal systems or those running an RTOS, Andes provides two options: TFLM or C. In the former case, models execute directly on an instance of TFLM; the latter case sees the model compiled into C code for direct execution. The C approach requires much less memory given the lack of an interpreter and more efficient overall code but is less flexible in the event of future model updates.

To test the flow, Andes executed 231 unmodified models taken directly from TensorFlow Hub; 215 were successful. Vision models worked best with a success rate of 96%, followed by audio (75%) and text (63%). Most failures were the result of TensorFlow Flex operators, which TFL doesn't support, or proprietary operators, according to the company.

A Simple Unit by Comparison

Andes competes with the likes of Cadence and Ceva for CNN inference IP at the edge. These competitors have had offerings in this space for many years, giving them time to improve their software capabilities and add hardware features. Cadence, for example, implements structured sparsity and tensor compression to reduce memory, memory bandwidth, and power (*MPR Oct 2021,* "Cadence Takes Tensilica AI to the Max"). Ceva, meanwhile, implements Winograd convolutions, unstructured sparsity, and INT4/INT8 mixed precision to speed performance and reduce power and memory bandwidth (*MPR Jan 2022,* "Ceva Tackles Unstructured Sparsity"). By comparison with these older engines, Andes' DLA is a simple, bare-bones model that implements structured pruning, although the company has hinted that some of the other features are on its roadmap.

In its immature state, Andes' ResNet-50 performance remains at roughly half the level that Ceva provides and two-thirds that of Cadence. Software tool maturity is critical here; Ceva's performance has increased by 50% since MPR's January 2022 coverage due to software improvements. AnDLA performance should improve over time if the tools evolve.

	Andes AnDLA I350	Cadence Tensilica DNA 100*	Ceva NeuPro-M NPM11
Peak AI Perf (INT8)	8 TOPS	8 TOPS	10 TOPS
Clock Freq	1 GHz	1 GHz	1.25 GHz
MAC Units	4,096 MACs	4,000 MACs	4,000 MACs
ResNet-50 Perf	1,756 IPS	2,550 IPS	3,734 IPS
Power	Undisclosed	0.90 W	0.89 W
Power Efficiency	Undisclosed	2,833 IPS/W	4,195 IPS/W
Production RTL	4Q23	1Q19	1Q22

Table 1. AnDLA vs competing AI IP. Andes performance is roughly half that of Ceva's; that company's results improved by 50% since prior coverage owing to software improvements. *Cadence numbers reflect a discontinued model that's been replaced by the NNA 110, which tops out at 4 TOPS. Performance and power numbers, however, are unavailable for the newer version. (Source: vendors)

Andes claims small silicon area as a primary benefit. Neither Cadence nor Ceva discloses area, making a direct comparison impossible. Expedera's Origin E6 DLA IP core has peak performance of 32 TOPS, so it's in a larger class of DLA. In a 7 nm process, its area is 7.0 mm^2; simplistically scaling the size by three-quarters to 8 TOPS would make it 1.75 mm^2, which is overly optimistic. AnDLA occupies 1.4 mm^2 in a 28 nm process—smaller yet than the scaled Expedera number. That result suggests that AnDLA competes well on die area.

Late to the Game

Andes' new AndesAire AnDLA deep-learning accelerator enters the market years after other similar IP has matured. Implementing a bare-bones feature set, at least in its inaugural configuration, the accelerator is equipped to handle basic vision-oriented machine learning with CNN models, dipping into the upper range of applications dubbed TinyML and scaling up from there. Typical examples are doorbells, presence detectors, and surveillance cameras—applications also supported by its competitors.

In its current implementation, the hardware implements no advanced features—supporting only INT8 data. INT8 is, for now, a popular data type, and for SoC makers simply looking to have an accelerator that's "good enough," INT8 will likely grab plenty of opportunities. The promised addition of other data types, such as INT4, FP8, FP16, and BF16, is necessary for covering a broader range of applications.

Scaling the DLA works best in conjunction with a vector CPU, which makes for a larger unit of replication than just the DLA. But even when building the most rudimentary implementation, a

CPU is required for any operations not natively handled by the DLA, including the softmax function. The company promises additional features and model support, like Transformers, in future editions, but given that the current DLA won't become fully available until the end of the year, those upgrades would appear to be far in the future.

DLAs rely as much or more on their supporting software stacks than they do on their architectures. A superior block of hardware with immature software tools can still lose out to a better-established solution. Andes hasn't had time yet to prove out either its hardware or software, and it competes with alternatives that have been on the market for many years; the company has a lot of catching up to do.

Andes' customers requiring basic CNN support can integrate other DLAs with an Andes CPU, but an Andes DLA already engineered to pair with its CPU will reduce integration work. For those customers, Andes is providing an easy path—potentially giving others a reason to choose an Andes CPU.

Price and Availability

Andes withheld licensing fees for its AndesAire AnDLA inference accelerator IP. The DLA is available now for early adopters; production RTL is scheduled for 4Q23. For more information, access the *company's website*.

