The Linley Group

MICROPROCESSOR report

Insightful Analysis of Processor Technology

WAVE ACCELERATES DEEP LEARNING

New Dataflow Processor Targets 10x Speedup for Neural Networks

By Linley Gwennap (October 3, 2016)

Introducing a new architecture for deep learning, start-up Wave Computing has stepped out as a leader in this emerging field. At the recent Linley Processor Conference, CTO Chris Nicol described how the company uses a dataflow processor (DPU) to efficiently execute the neural networks that form the core of most modern deep-learning applications. The design uses self-timed logic that reaches speeds of up to 10GHz. The 16nm chip contains 16K independent processing elements that can generate a total of 180 trillion 8-bit integer operations per second.

As interest in deep learning grows, several companies are investing in hardware acceleration. Google has developed and deployed a deep-learning ASIC in its data centers, but it has withheld information about its design. Intel recently acquired deep-learning pioneer Nervana, which had yet to disclose its architecture. Companies such as IBM, Knupath, and Nvidia have adapted existing architectures to deep learning (see *MPR 6/27/16,* "Learning Chips Hit the Market"), but Wave is the first to unveil a processor architecture designed specifically to accelerate deep learning.

Founded in 2010, Wave is a Silicon Valley company with venture funding from Tallwood and Southern Cross. Nicol has led the development of its dataflow architecture; he formerly developed multicore DSP chips at Bell Labs. Dado Banatao, who founded Chips & Technologies and S3 Graphics, is Wave's chairman. Derek Meyer recently became CEO after executive stints at ARC, Ceva, and MIPS. The startup has grown to 45 employees and has received more than 50 patents.

Instead of selling DPU chips, Wave will sell systems designed to accelerate deep-learning training. It will soon tape out its first DPU chip and plans to offer evaluation systems in 2Q17, with production systems arriving by the end of that year. The company expects its systems to deliver a 10x improvement over current GPU-based designs for training neural networks.

Go With the Dataflow

Neural networks comprise a set of "neurons" that apply weights to a set of inputs and calculate an output value based on a nonlinear function. Many implementations use a convolution function, such as an FFT, to compute the output value; this approach is called a convolutional neural network (CNN). The neurons are grouped into layers such that the outputs from one layer connect to the inputs of the next. A complex network with many layers, as Figure 1 shows, is called a deep neural network.

Neural networks are not programmed in a conventional fashion; rather, they are "trained" using sample data.

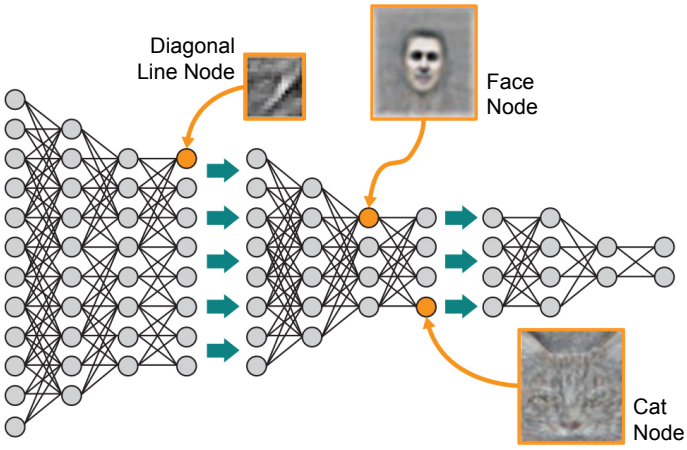

Figure 1. Typical neural network for deep learning. A neural network is often represented as a dataflow graph showing how inputs combine with weights to determine the outcome—in this case, whether an image is a face or a cat. (Source: Wave)

The training process establishes the optimal weights for each node. For example, if the network in Figure 1 detects a diagonal line, the weights will steer the result toward the cat node, whereas a round shape will steer closer to the face node. Once the network is trained and the weights are fixed, it can be deployed to process real data (in this case, images); this task is called inferencing.

This diagram resembles a dataflow graph. Computer scientists use the term *dataflow* to refer to a model that represents a program as a graph of data flowing between operations. In other words, dataflow programming focuses on how the data moves rather than how instructions are sequenced. Like most fundamental computing concepts, dataflow has been around since the 1960s, but it has become more relevant with the rising interest in neural networks. Thus, Wave decided to apply dataflow technology to the problem of accelerating neural networks.

Large neural networks can contain millions of neurons. Evaluating the state of this many neurons is highly compute intensive, and training such a complex network requires millions of iterations. Most researchers employ a combination of general-purpose processors and graphics chips to train large CNNs, but these chips are not designed for such a workload. Researchers at Intel, MIT, and others have demonstrated up to 30x improvements in performance per watt using specialized hardware (see *MPR 3/7/16,* "Accelerating Machine Learning"), but no solution of this kind has yet reached the market.

Wave's approach is to create an architecture that accelerates dataflow processing. The concept of a dataflow processor is far from new. After pioneering the concept of dataflow programming at MIT, Jack Dennis began researching a dataflow processor in the 1970s. Creating a pure dataflow machine proved impractical, however, because such a design would need to broadcast data from each function unit to any other function unit and manage dependencies across the entire chip. Wave instead uses a hybrid approach that combines standard instruction execution with dataflow principles.

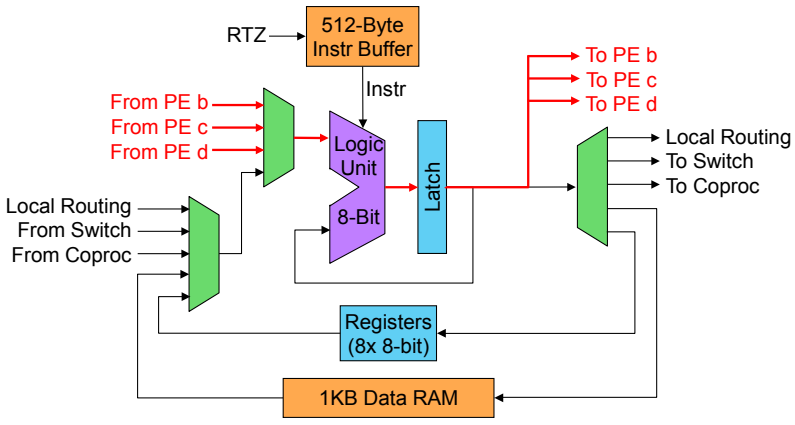

Figure 2. Wave DPU processing element. Each PE is an independent processor with its own instruction memory, data memory, registers, and compute unit. The critical timing path appears in red.

A Tiny Green PE

The processing element (PE) is the smallest part of the architecture. Each PE is an independent processor with its own instruction memory, data memory, registers, and compute unit, as Figure 2 shows. It can emulate a single neuron, performing the desired convolution or other evaluation function. The instruction memory is a simple circular buffer that holds 256 instructions—enough for even relatively complex convolutions. The instruction counter is reset to zero (RTZ) at the beginning of each evaluation; no other branches are implemented. By precluding branches, the PE avoids the need for branch prediction. The instructions are just 16 bits each, but Nicol withheld information about the instruction set.

Unlike modern CPUs, which can reorder instructions and group operations on the fly, the DPU is statically scheduled. The evaluation functions are straightforward and simple to map onto the PE's resources. Decision making in a CNN occurs through these evaluations rather than by explicitly changing the program flow. Furthermore, popular CNN frameworks such as Google's TensorFlow employ static graphs, so the architecture fits the application. Avoiding dynamic scheduling eliminates pipeline hazards and simplifies the processing element, allowing Wave to fit so many of these elements on a single die. A single PE requires less than 0.01mm^2 of die area—smaller than a Cortex-M0+ with a similar amount of memory.

The PE contains an 8-bit logic unit that handles compare (if equal), shift, rotate, and logical operations (e.g., AND, OR). Neural networks are increasingly moving to 8-bit weights to simplify the computation problem, so the PE is designed for this approach. The output of the logic unit is latched and serves as one operand for the next instruction. The output is also available to other nearby PEs, implementing the dataflow model. In addition, results can be stored in an eight-entry register file or a 1KB data RAM, although accessing the RAM takes twice as long as accessing the register file.

Wave implemented the PE using asynchronous logic. Instead of following a central clock signal, the PE computes a new result as soon as it receives the necessary operands. The pipelined instruction buffer can deliver a new instruction every 0.1ns (10GHz), setting the machine's maximum speed. Most instructions exchange data with a nearby PE instead of reading the register file, shortening the critical timing path. The company expects the typical operating frequency to be about two-thirds of the peak rate, or 6.7GHz. Asynchronous logic is a good fit for this dataflow design but rarely appears in commercial processors (see *MPR 11/29/04,* "ARM's Asynchronous Handshake"). Nicol withheld comment on the specific type of asynchronous signaling that Wave uses, but eliminating the clock-distribution tree can reduce die area and power considerably.

Good Things Come in Clusters

The DPU groups 16 PEs into a cluster, which contains additional compute units, as Figure 3 shows. Each pair of PEs connects to an 8-bit adder. For applications that use 32-bit weights, the cluster contains two 32-bit ALUs that can execute a multiply-accumulate (MAC) operation with signed or unsigned saturation. The ALUs lack native support for floating-point (FP) operations, but they support FP emulation. If desired, they can operate together as a single 64-bit ALU. Conversely, the ALUs can process 8- or 16-bit operands in SIMD fashion. The cluster also contains two bit-manipulation units (BMUs) that implement shifts as well as bit-field extraction and insertion.

The cluster is divided into four "quads" of four PEs. Each PE connects directly to the other three PEs in its quad using the connections marked *b, c,* and *d* in Figure 2. Although the PEs can operate independently, the four elements in a quad must work together to feed data to the 32-bit ALUs and BMUs. The static scheduling ensures that these PEs execute in unison, passing data in parallel to the wider units.

Table 1 summarizes the cluster's compute resources. The 16 PEs can generate 16 operations plus 8 operations from the shared adders. The two 32-bit ALUs can perform MAC instructions (which count as a multiply and an add, or two operations) but take seven times longer than a basic add before they can start a new instruction. Similarly, the two 32-bit BMUs take three times longer than a basic add. Finally, the memories contain four auto-increment units. This all adds up to about 31 operations per cycle or 207 billion operations per second (GOPS) at the nominal 6.7GHz clock speed.

The ALU and BMU cannot launch operations simultaneously, and the design has a few other constraints. To account for these factors, Wave subtracts 15% from the absolute-maximum operation rate to quote 176 GOPS for the cluster or 181 trillion operations per second (TOPS) for the chip. Many of these 8-bit operations are shift or logical operations, however; the total arithmetic rate is 110 trillion adds or just 8 trillion MACs per second.

Each cluster has four independent byte switches plus a (32-bit) word switch that connects to adjacent clusters in the north, south, east, and west directions. This configuration allows the clusters to be tiled. The switches can pass data into the cluster for use by the PEs, or it can route the data to another cluster. All of the switches can buffer data to avoid congestion. The switches are pipelined and instruction driven, meaning the compiler programs them to implement the desired algorithms.

The DPU implements power management at the cluster level. When a cluster finishes processing its assigned task, it can place itself in a sleep mode. All of the PEs and byte switches then shut down (via clock gating), but the word switch remains active, routing data to other clusters. If the switch receives valid data designated for one of its PEs, it can wake the cluster.

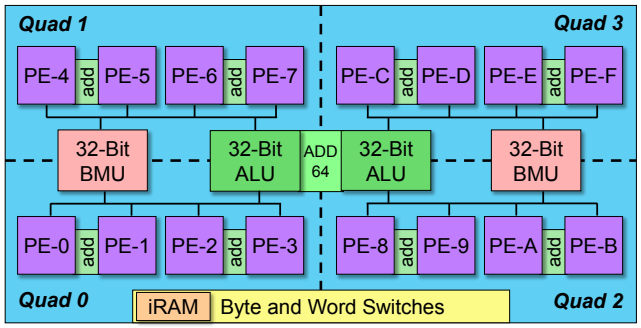

Figure 3. Wave DPU processing cluster. Each cluster contains 16 processing elements plus additional shared compute units. The switches connect to nearby clusters.

The cluster also supports a DMA mode. When the cluster is asleep, the word switch can write into a PE's instruction memory or data memory. Using the switches' fanout capability, a DMA transfer can update all of the PEs with new code, for example. Data can also be transferred to an individual PE without waking the cluster.

A Cast of Thousands

The DPU chip contains 1,024 clusters tiled in a 32x32 array. The word switches allow any cluster to send data to any other cluster, routing the data through a series of hops until it reaches its destination. This type of mesh interconnect appears in many high-core-count processors, but few designers have attempted it with so many cores. The large array extends the average number of hops to about 24, versus about 6 hops for a 64-core array. (The DPU figure excludes additional transit time from the word switch to the individual PEs.) Wave points out that the array's bisection bandwidth is a massive 7.25TB/s, but most of this bandwidth is eaten up by shuffling data from cluster to cluster. Dividing by the average number of hops leaves about 300GB/s of useful bandwidth, or 300MB/s per cluster.

Although each cluster operates asynchronously, the chip contains a self-timed interlocking network that synchronizes neighboring clusters to minimize skew. The array

Per Cluster	Repeat Rate	8-Bit Ops Per Cluster	8-Bit Ops Per Chip	Type of Ops
16 PEs, 8 bits each	1	107 GOPS	110 TOPS	Shift, logical
8 adders, 8 bits each	1	54 GOPS	55 TOPS	Add, subtract
2 MAC units, 32 bits each	7	15 GOPS	16 TOPS	Multiply, add
2 BMUs, 32 bits each	3	18 GOPS	18 TOPS	Shift
4 addr units, 16 bits each	2	13 GOPS	13 TOPS	16b increment
Total		207 GOPS	212 TOPS	
Total – 15%		**176 GOPS**	**181 TOPS**	

Table 1. Wave DPU peak performance. Each of the 1,024 clusters can generate a peak rate of 207 billion 8-bit integer operations per second at 6.7GHz, but the company conservatively derates this number by 15% to account for internal constraints. (Source: Wave)

Price and Availability

Wave Computing has no plans to sell the DPU as a chip, instead offering it as part of a complete system. The company plans to supply evaluation systems in 2Q17 and production systems by 4Q17. To download a free copy of Chris Nicol's presentation from the Linley Processor Conference, access *www.linleygroup.com/processor-conference*. For more information on Wave, access *www.wavecomp.com*.

is divided into 24 regions, each containing 32 or 64 clusters, as Figure 4 shows. Each region, which Wave calls a compute machine, connects to a 128-bit AXI bus at the chip's periphery. These 24 interfaces connect to an AXI4-based network that bridges the compute array to memory and I/O. The network includes 32 AXI4 buses with a total peak bandwidth of 410GB/s. The network also connects to an Andes N9 microcontroller core that configures and supervises the chip.

To execute a neural network, the compiler breaks down the evaluation function into a series of simple steps, such as a matrix multiply followed by a matrix add followed by a sigmoid function. Any of these functions can be encoded into the instruction memory of a single PE or a group of PEs. The compiler assigns some PEs to handle matrix multiplication, others to perform addition, and so on. Data then flows from one group of PEs to the next, creating a software pipeline. Once the PE executes its code, the instruction buffer resets to zero and begins executing again on the next datum. This procedure continues until all data has been processed, then a new set of code can be loaded for the next stage of the network.

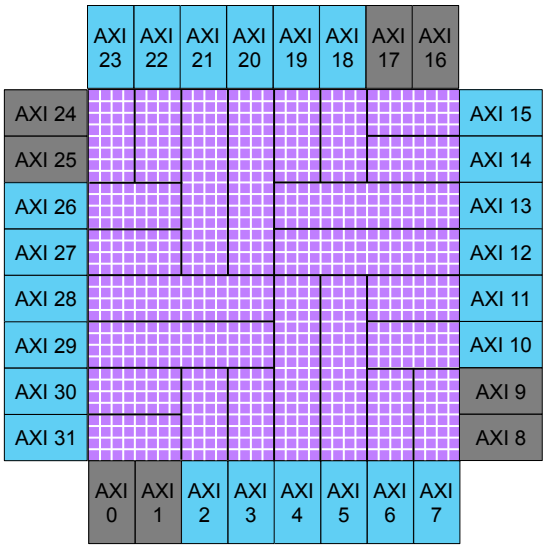

Figure 4. Wave DPU processor array. The chip contains 1,024 clusters in a 32x32 array divided into 24 compute machines, each with its own AXI interface.

The chip contains a total of 16MB of data memory and 8MB of instruction memory, distributed among the PEs. Instead of a cache, it has a separate 1MB program store that, combined with the DMA capability, allows manually reloading the PEs with new instructions. This program memory can be encrypted to protect the customer's code. Other than the local PE data memory, data is not cached on the chip. This approach is tuned for CNN training, which stores the weights in local memory and steps through the training cases one by one without any data reuse. Because the chip lacks caches, it needs no cache-coherency logic.

HMC Pumps Up the Bandwidth

To load data into the clusters, the chip uses four Hybrid Memory Cube (HMC) interfaces. HMC is a high-bandwidth stacked memory that can reside close to the processor to maximize the transfer rate (see *MPR 9/12/11*, "Micron Reinvents DRAM Memory"). The Fujitsu Sparc64 XIfx is the only other announced processor that employs HMC. Using 15Gbps serdes, each HMC 2.0 interface provides 240GB/s of peak memory bandwidth, roughly matching the useful bandwidth of the on-chip interconnect.

Each HMC device holds 2GB, yielding a total memory capacity of 8GB. To store larger amounts of data, the DPU chip also has two 64-bit DDR4-2400 memory interfaces, which support ECC for greater reliability. Using all of these interfaces, the processor can handle 2,048 outstanding memory requests, or two per cluster. The DPU also includes a 16-lane PCI Express Gen3 interface that can connect to a network or host processor.

All of this performance comes at a cost. Nicol said the die is about 400mm^2 in TSMC's leading-edge 16nm FinFET process. The die could have been even larger had it packed in more PEs, but the limiting factor is the power consumption: about 200W (maximum). The package is likely close to 2,000 pins. Because of the design's regular tile structure, Wave can improve manufacturing yield by simply deactivating defective compute machines.

Wave Good Buy

Instead of selling an expensive chip, Wave will build and sell systems, most likely to data-center operators that offer neural-network training in the cloud. The company is developing a rack-mounted 3U system that contains 16 DPU chips mounted onto four boards. As Figure 5 shows, each board contains four DPU chips that share a pool of four HMC memories, although each DPU has its own 64GB of DDR4 DRAM. A quad-core ARM processor manages the board. The system connects to an Ethernet network and is controlled by Wave's host software running on a standard x86 server.

The complete system contains 256K processing elements generating a peak rate of 2.9 petaops (quadrillion 8-bit operations per second) or 126 trillion 8-bit MACs per

second. On one CNN simulation, the design sustained 71% of its peak MAC rate. Supporting this computation speed is 256MB of on-chip data memory, 32GB of fast HMC memory, and 1TB of DRAM. But you might need a bigger plug, as the system burns close to 4kW of power.

The system is designed to run neural networks written to Google's TensorFlow API and similar APIs, such as Facebook's Torch, Microsoft's CNTK, and UC Berkeley's Caffe. These tools allow users to create a neural network and evaluate it using library code. By creating the appropriate library routines for its DPU architecture, Wave can provide a transparent method of accelerating CNN training without changing the customer's workflow. It will enter production with a CNN graph library for TensorFlow and plans to later support CNTK and Caffe.

Researchers have yet to determine the optimal scope for CNN weights. Wave's PEs are optimized for 8-bit integer values, but the DPU can handle 16- and 32-bit integer weights using PE quads to feed the cluster ALUs. As with most processors, the wider values reduce performance. Wave has implemented a 16-bit fixed-point mode that employs a 32-bit accumulator and stochastic rounding to achieve accuracy similar to that of the 32-bit floating-point calculations that many neural networks employ today, but with much better performance.

Using this 16-bit approach, Wave estimates its chip will deliver at least 10x the performance of a high-end GPU. For example, on the Inception v3 deep neural network (a standard TensorFlow model), the company estimates that its 16-DPU system can run 90 training iterations of 1.28 million images each in just 15 hours. By contrast, a system with 16 Nvidia Kepler GPUs takes about 240 hours to complete the same training.

Nvidia's latest Pascal GPU is about twice as fast as Kepler, achieving 10.6Tflops of single-precision FP performance while burning 300W TDP (see *MPR 5/9/16,* "Nvidia Hits HPC First With Pascal"). It's also configurable for 21.2Tflops of half-precision FP. In contrast, the DPU chip delivers 90 TOPS for 16-bit integers and 181 TOPS for 8-bit integers. So, as long as the application can use integer weights instead of FP, the DPU delivers about 12x more operations per watt.

Surf's Up!

Deep learning is appearing in a wide range of applications, such as vision processing (object recognition), speech recognition, and artificial intelligence. Instead of writing code that anticipates every possible situation, a neural network can configure itself using a set of training runs. This approach simplifies the coding task and can provide similar or even superior results. Most CNNs are developed using Intel general-purpose processors or Nvidia GPUs, but training a large CNN in this way can require hundreds of machines or weeks of time on a single machine.

Wave's DPU architecture is quite unlike these traditional methods. Instead of optimizing for 64-bit processing, it's designed for 8-bit values, although it can also handle 16- and 32-bit integers. Since it runs neither operating systems nor applications, the instruction set is pared down, simplifying the decoder and fitting into 16-bit encodings to reduce the program-memory size. Moving scheduling into the compiler eliminates hardware scheduling logic. Static scheduling also allows replacement of caches with simpler RAMs and eliminates cache-coherency logic.

In addition to being small, Wave's processing element is also blazingly fast. Getting all of this complicated logic out of the critical timing path enables the computation to run faster. Using asynchronous logic further speeds the PE, since a computation can start as soon as the previous one finishes instead of waiting for the next clock cycle. The company designed the PE to run at a maximum speed of 10GHz and expects an average speed of 6.7GHz—still far faster than any Intel or Nvidia processor.

The small PE design allows Wave to pack 16,384 of these 8-bit cores onto a single die. Combining this huge number of processing elements with the superfast clock speed yields a tremendous amount of compute capability in a single chip. Unlike GPUs, which are designed for complex floating-point operations, the DPU focuses on integer operations, improving power efficiency.

Wave must still demonstrate that its hardware works as planned; first silicon should be available by year-end. Furthermore, the performance of a statically scheduled design depends heavily on the compiler's ability to efficiently map the code to the compute resources. The high ratio of total operations to MACs (181 to 8) helps the compiler maintain a MAC utilization of up to 71% in simulations. If Wave can deliver, its architecture should provide an order-of-magnitude boost in training performance for those neural-network designers willing to use integer weights. ♦

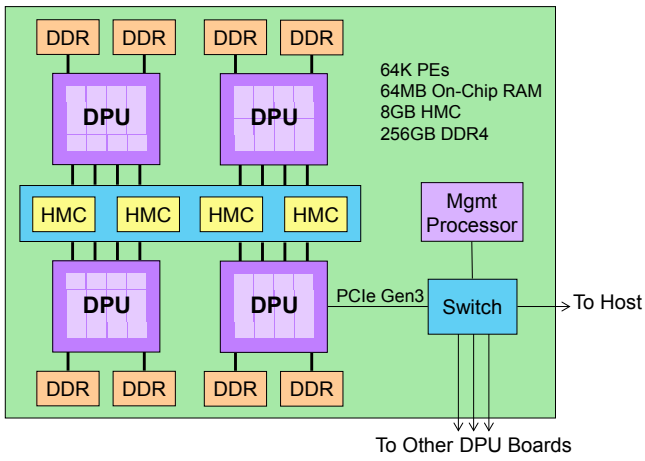

Figure 5. Wave DPU board design. This board combines four DPU chips, which share an 8GB pool of HMC memory. A PCIe switch connects to other DPU boards and a host processor.