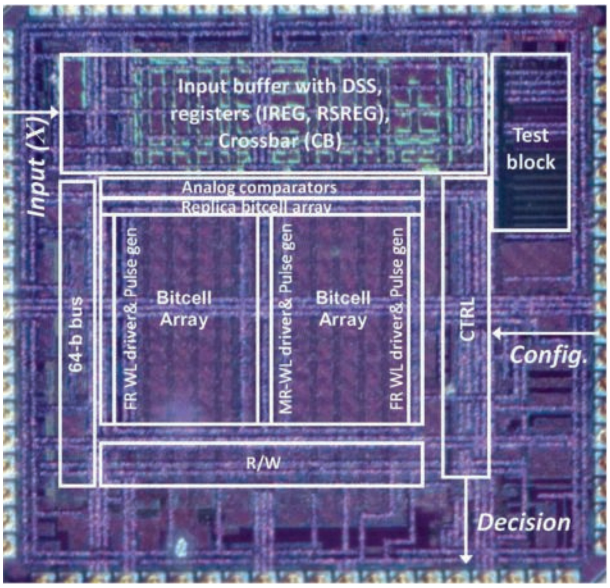
Mixed signals for **machine learning**

Maybe AI-enabled systems are here to stay, but the cost of computation remains an issue. Could processing in the analogue domain provide a solution? By **Chris Edwards**



Above: One of the Shanbhag group's DIMA test chips used for doing random forests

Multiple technological winters have stymied the development of machine learning. But now that smart speakers have invaded the home and Amazon has decided the time is ripe for a $60 microwave that can take orders from Alexa, maybe AI-enabled systems are finally here to stay.

The problem that continues to face embedded applications of AI is the cost of computation. Much of the work has to be performed using high-speed digital processors, often in the cloud because the battery will not sustain local processing. Developers are looking to methods to cut the energy bill. Processing in the analogue domain is one possibility.

If you look at the gate count of a high-speed multiplier, a fundamental building block for most AI algorithms used today, it is easy to believe a simple analogue equivalent would be more energy efficient.

At the SysML conference earlier this year, nVidia chief scientist Bill Dally talked of running SPICE simulations to work out whether analogue is a viable approach when it comes to performing machine learning. But he quickly saw problems emerge. One is the issue of matching analogue devices to the required level of accuracy though this is not the primary concern.

One big difference between computation for machine learning and conventional signal processing is that systems such as neural networks are highly tolerant of errors. This has led companies pursuing embedded machine learning, such as IBM, to look at techniques such as approximate computing. This may involve letting arithmetic units make mistakes as long as they are not significant. This could let circuits operate very close to their voltage limits, which will help to save power. If an operation does not complete in time because it is voltage starved, it does not matter much to the overall answer.

Another approach is simply limited precision.

Limited precision

Low-precision is a problem for training neural networks. But researchers have been surprised by how low the precision can go during the inferencing phase when the network is comparing real-world data to the model on which it was trained. The precision came down quickly from full double-precision floating point to 16bit and then to 8bit. This makes neural networks fit well on SIMD pipelines. But some proposals have gone further to binary and ternary resolution with little adverse effect on performance in some applications.

Low-quality analogue should have little difficulty keeping up with those levels of accuracy and, in principle replace the high-speed multipliers with circuits that could be as simple as voltage-controlled amplifiers. But Dally noticed that a comparison of the energy each operation needs at the same level of precision, digital works out better because it is quicker.

"If you look at things like the fact that these circuits are leaking over the 10 microseconds it takes to do the computation it's actually way higher energy to do it in analogue if you take the leakage into account," he argues. "Digital CMOS logic is amazingly efficient especially at very low precision. If I'm doing 2 or 3 bit operations the arithmetic is actually almost in the noise and then it's leakage that dominates."

Digital processing has the advantage of being able to cut leakage by turning unused elements off. This is much harder with analogue techniques. But analogue machine learning is not dead.

Although digital processing makes it easier to control leakage, there are many situations in system design where leakage is inevitable. So, you might as well take advantage of it. Researchers such as Boris Murmann of Stanford University have been focusing their effort on hybrid systems that use analogue where it does turn out to be more efficient. One area is preprocessing.

Preprocessing

Security cameras and smart speakers are systems that need to maintain a low level of activity almost constantly simply to work out whether something important is happening nearby.

"It makes sense to have a wakeup algorithm. For it, you may sacrifice programmability for extreme energy efficiency. If I only want to wake up when I see a face I don't necessarily need a very sophisticated deep learning-based algorithm," Murmann explained at a workshop on machine learning at the VLSI Circuit Symposium in Honolulu. "I'm not trying to do a humungous calculation. I'm just trying to work out what is in the field of view and decide: should I wake up my big brother?"

Processing in the analogue domain makes it possible to work at comparatively low resolution because it can support a high dynamic range. "When analogue people hear 'dynamic range' they know what to do," Murmann says.

High dynamic range makes it easier to deal with images that may contain strong shadows that might otherwise disrupt object-recognition models. "With optimal exposure, I can reliably detect objects even with 2bit resolution," Murmann says.

Real-time threshold adjustments coupled with logarithmic processing in the analogue domain makes it possible to reduce the effects of shadows massively.

"These analogue circuits are a bit tricky and require a lot of analysis," Murmann says, but the team has sent test silicon to the fab. The result of using the front-end is a 20-fold reduction in the data that passes into the next stage compared with conventional sensor interfaces.

The next step is to work out where to put analogue circuitry in the machine-learning cores. MIT associate professor Vivienne Sze points out that the energy consumption of most AI algorithms is dominated by the shifting of data to and from memory. She says there is a threefold increase in memory transactions for even a relatively simple deep-learning system such as AlexNet compared to what was needed for traditional image-recognition algorithms such as histogram of oriented gradients (HOG). To overcome the high cost of off-chip accesses to main memory, digital implementations try to cache as much data as possible. This works reasonably well for the convolutional layers in deep neural networks. Fully connected layers are more troublesome for caching strategies but convolutional layers tend to dominate the processing in these systems.

For front-end processing, the amount of data that needs to held for neuron weights and for the preprocessed source image is relatively small: it should fit into local memory, which reduces the power overhead. But with analogue processing, it is possible to go further.

Professor Naresh Shanbhag of the University of Illinois says: "With conventional digital design, the memory read is highly curated. That hurts energy efficiency."

The group's approach is to avoid needlessly converting the charge levels stored in SRAM into digital ones and zeros and simply process them directly in the analogue domain. "The computation wraps its arm around the bitcell array. We don't touch the bitcell itself: everything we do stays on the periphery. The computational signal-to-noise ratio drops but we can take advantage of the error tolerance of machine-learning algorithms," Shanbhag explains.

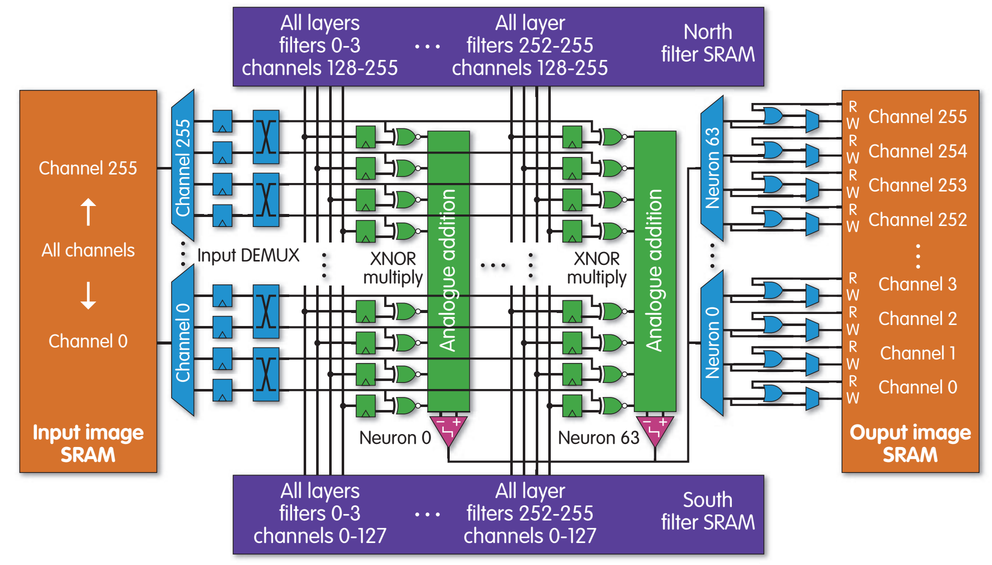
"Digital CMOS logic is amazingly efficient especially at very low precision. If I'm doing 2 or 3 bit operations the arithmetic is actually almost in the noise and then it's leakage that dominates."
Bill Dally

The first test devices used SRAM but Shanbhag says the team is working with memory maker Micron Technology to develop a flash-compatible version, which would reduce leakage energy for always-on systems. The regular nature of the design lends itself to the compilation and synthesis techniques used by SoC designers to generate on-chip memory arrays, he adds.

Murmann's group has taken a similar approach, developing an architecture that looks like a memory turned inside-out. Data from the image is passed through a network of demultiplexers and into XNOR gates that act as analogue multipliers. These low-resolution multipliers process each input against a set of weights with the results forwarded to an analogue summing bus. The final stage is through a second mux network to the output memory.

Although analogue computation is not going to revolutionise machine learning for embedded systems, work on hybrid architectures could provide a way to bring AI to always-on devices.

Below: A diagram of the Murmann architecture



Weight-stationary architecture with input reuse and binary-CNN-specific sliced datapath