# On-Chip FPGA: The “Other” Compute Resource

*Expanding the possibilities in the processor subsystem.*

DECEMBER 2ND, 2021 - BY: ANDY JAROS

When system companies discuss processing requirements for their next generation products, the typical discussion invariably leads to: what should the processor subsystem look like? Do you upgrade the embedded processors in the current subsystem to the latest and greatest embedded CPU? Do you add more CPUs? Or perhaps add a little diversity by adding a DSP or GPU?

One compute resource that rarely (if ever) gets consideration is on-chip FPGA (a.k.a. embedded FPGA or eFPGA). And that’s understandable for two reasons: FPGA that can be integrated on a chip is a relatively new IP and when people think of FPGAs, they don’t necessarily think of it as a compute resource.

Before coming to Flex Logix, I thought of FPGAs pretty much like everyone else. It's an alternative to a custom ASIC for low volume applications that don't have stringent power requirements. While eFPGA functionality is essentially the same as FPGA chips, eFPGAs are actually more powerful because they can go anywhere in an SoC and can execute certain functions better than an embedded processor. Here are a few examples.

**Filtering**: We ran a benchmark comparing the energy consumption required to run several filter operations on a small eFPGA as compared to an embedded CPU with DSP instructions. What we found is that the embedded CPU took 50% to 475% more energy to execute despite the eFPGA using more area. The application note can be found here.

**Packet processing**: Many NIC applications require looking at specific bits in a wide data packet and performing some type of calculation prior to sending the packet on to the next stage in the pipeline. The specific bits that need to be modified change from application to application necessitating programmability. Embedded processors inherently induce latency due to dependencies on local or cache memory which directly impacts throughput. Dedicated state machines give better

performance but are inflexible once committed to silicon. eFPGA offers an ideal solution because they provide near ASIC performance yet have full re-programmability. An added bonus is that designers do not need to overdesign their state machine to anticipate every possible future use case. They simply program in a new state machine.

Analog chips: It seems as though sensors of all different varieties are being added to every conceivable device, generating more and more data. Filtering out the un-needed data is being pushed to the edge to lower the processing load on the main processor and lower system power. Edge processing choices tend to be small, embedded CPUs or state-machines. As with the packet processing example above, a small simple state machines can be programmed into the eFPGA to target a sensor for a specific application, yet the same sensor can be programmed with a different state machine for an entirely different application not conceived of when the sensor was developed thereby extending the market for that sensor. Additionally, with only a few hundred LUTs, eFPGAs can efficiently execute ML models using binary neural networks. Embedded CPUs can be used, but they tend to drive up the chip cost and may be overkill for many sensor applications. As for power, Lattice's iCE40 and Renesas' recently announced ForgeFPGA are both proof points that FPGA technology can be used for low power applications. The is true for eFPGA IP as well.

Accelerators: It's well documented that using dedicated hardware accelerators is how chip architects are driving performance increases due to the flattening of Moore's law. In this case, the eFPGA is used as a "side car" to house the accelerator and provides near ASIC performance and full re-programmability.

eFPGA IP is a flexible technology that can be used just like an FPGA but thinking of it as a flexible fabric limits its potential. Whether it's a low power application or one that needs a high-performance accelerator, eFPGA can be a great alternative to an embedded CPU as a powerful compute engine!


## Andy Jaros

(all posts)

Andy Jaros is vice president of sales and marketing at Flex Logix.