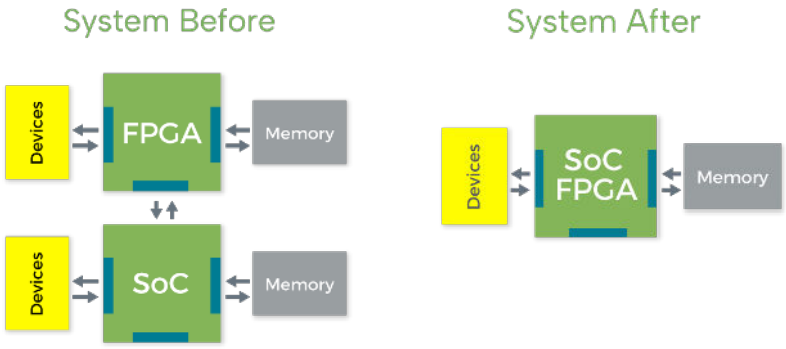
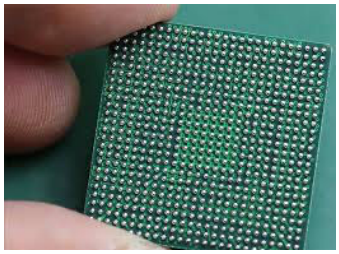
Cut Power+Cost 5-10x: Integrate FPGA in your SoC

You Can Integrate FPGA in Your SoC at Full Speed and Flexibility

Until EFLX eFPGA, it was not possible to integrate full-speed, high density FPGA in an SoC. Now you can.

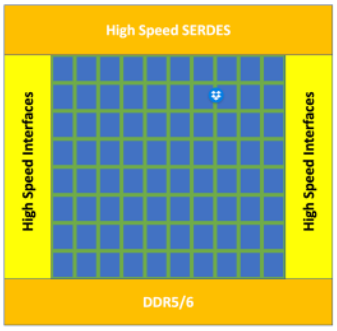


Steps and Savings from FPGA Chip to eFPGA in your SoC



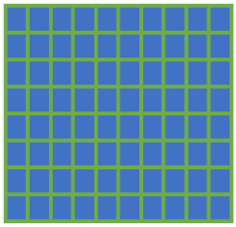
FPGA chips are high cost devices with a high profit margin for the manufacturer: this goes away when you integrate.

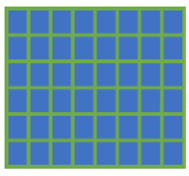
FPGA packages are large and expensive because of the large number of very high speed signals that require expensive signal integrity design and packaging layers. When you integrate this goes away. And you save the board area the FPGA package took; and eliminate expensive voltage regulators.



FPGA die have 20-30% of the area dedicated to various very high speed SERDES, DDR interfaces, PCIe interfaces, JESD and more. When you integrate these go away.

This leaves the digital core. In a typical FPGA you get a fixed ratio of LUTs, DSP MACs and BRAMS of which you may only use some of them. With EFLX eFPGA we can give you a customized array that has exactly the resources you need with no wastage. Finally, portions of an FPGA design are fixed so they can be hardwired in your SoC.





Integrating eFPGA maintains full FPGA speed and flexibility and cuts power and cost by 5-10x. **Really good but you can do better: see over.**

	Cost Saving	Power Saving
No FPGA margin	60%	
No FPGA package, reduce PCB area, voltage converters	15%	10%
Remove •SERDES/DDR/IO •Unused blocks of the FPGA •Harden unchanging FPGA logic	15%	75%
TOTAL	90%	85%

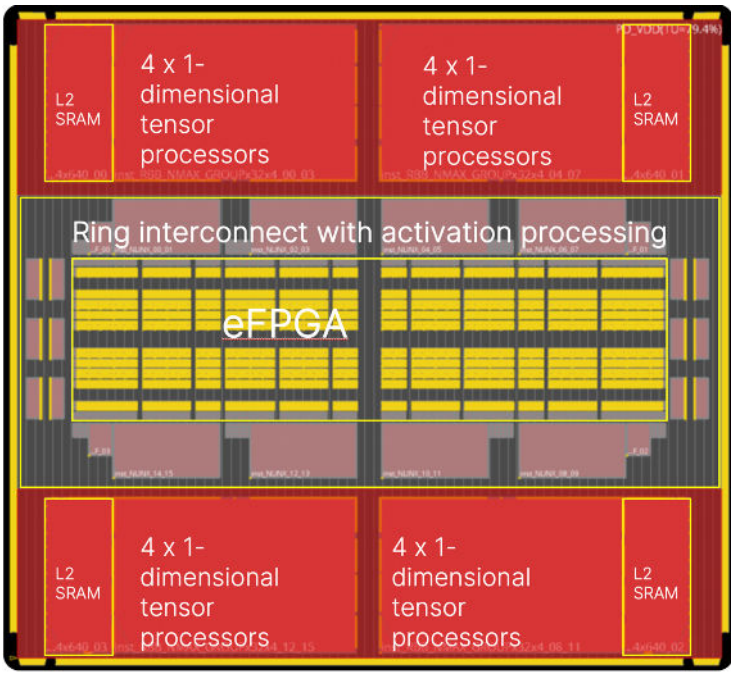
Fast, Flexible Control Path for 10x Better PPA

www.flex-logix.com

Further cut cost+power: Hardwire the Data Path, eFPGA for Control Path

You want to be able to handle changing standards, algorithms and customers needs and run fast: FPGA's flexibility and parallelism enables this. Fully hardwired has good PPA but can't adapt.

When you integrate using eFPGA, you can achieve even better power/performance/area (PPA) than all-FPGA by hardening the compute into modules with a programmable interconnect to achieve higher speed at lower area, while keeping eFPGA for high speed state machines that are reconfigurable to control the data path in a fully programmable way to adapt to changing standards, algorithms and customer needs.



This is exactly what we have done for our InferX AI solution (on the right: to be described in more detail in Q2/2023). An all-FPGA solution with the throughput of InferX would be ~10x more area.

FPGA customers typically get 200-300MHz throughput because of long paths. We run 500MHz in 16nm because we only implement the control state machine in eFPGA; at 7nm we get 800MHz. The eFPGA is reconfigurable to execute hundreds of different inference operators (and can reconfigure in microseconds).

We have 16 hardwired 1-dimensional tensor processors with a programmable interconnect that allows them to be connected in various sets of serial and parallel arrangements to optimize performance for each inference operator.

We can work with you to help you improve your power, performance, are for your architecture while keeping full flexibility for changing workloads, standards, algorithms and customer needs.

	SoC + FPGA Chip	SoC with eFPGA	SoC with hardwired modular compute, programmable interconnect, eFPGA control path
Adaptability	High	High	High
Cost	High	5-10x less	25-50x less
Frequency	Lowest	Higher	Highest
Reconfigurability	Low (Flash boot)	Good (milliseconds)	Best (microseconds)

flexlogix reconfigurable