Modular FPGA Makes FPGA Easier to Use

Make FPGA easier to program

Customers love the flexibility and parallel processing of FPGA which enables it to outperform processors on many workloads. But programming FPGA is hard: Verilog is low level and FPGAs are not modular. Modular FPGA makes FPGA easier to use.

Processor Software is Modular and Reconfigurable

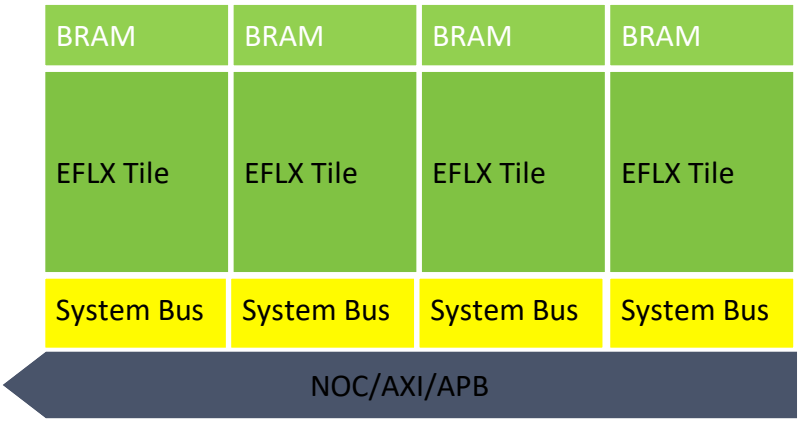
Code for processors is written in subroutines. Rewriting one subroutine doesn't change the characteristics of other subroutines.

Code is stored in DRAM then paged into cache in blocks then paged into executable memory. Wherever code is stored it executes in the same way.

Workloads change dynamically over time in response to changing needs.

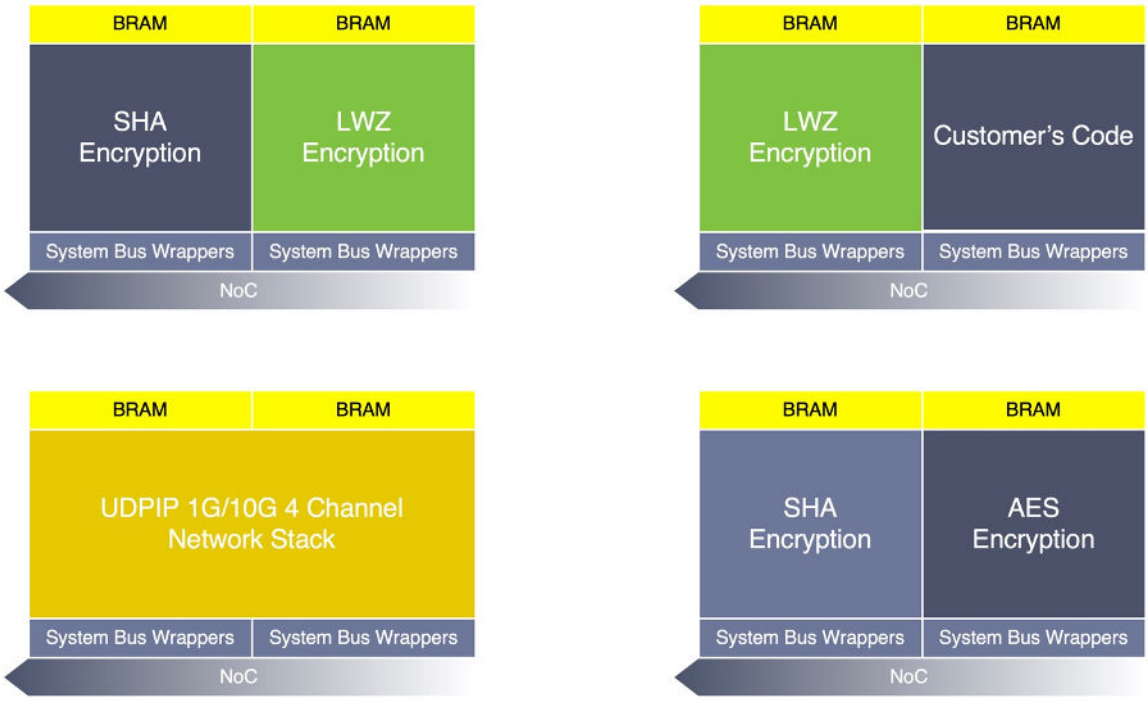
Modular FPGA is Programmed more like a Processor

To have FPGA code that runs the same regardless of location and the presence of other FPGA code we need to make the FPGA Modular.
Flex Logix EFLX® eFPGA is already modular since it is constructed using tiles which are abutted. Interconnects within each tile exist for intra-tile communication; for tile to tile communication a top level mesh interconnect is used.
So if an eFPGA array is constructed with every tile having the same bus connections to processor/memory/IO, then each tile "looks identical" to code running on that tile.



With the FPGA above, it can be programmed as a single large FPGA. But you can also write code that runs just within a single EFLX Tile: each tile has identical resources and bus connections, so the code can run in any of the tiles. And, if you need more LUTs you can write the code to take 2 tile (or more). Now you can offer your customers a library of executables that they can run internchangeably. Or your customer can program themselves. Or a mix of the two.

The modular FPGA can be 2 or more tiles. If you have a library of code modules the customer can figure their silicon as desired at boot time and different customers can use different combinations to customize your SoC for their use case:



Reconfigurable Modular FPGA for Changing Workloads

Traditionally FPGAs are configured once at boot/power-on. This is because they almost always store the configuration file in a Flash memory which is updated from time to time (like your smart phone's OS and apps).

But eFPGA is in your SoC, so you can provide the configuration files from on chip SRAM, on chip NVM and/or off chip DRAM.

EFLX eFPGA is reconfigurable. Process nodes like 40nm reconfigure more slowly than 7nm but all EFLX eFPGA can be reconfigured in milliseconds with configuration bits coming from DRAM.

And with cache SRAM adjacent to the array, reconfiguration for FinFet node EFLX eFPGA can happen in microseconds (we do this now in silicon for our InferX AI/DSP IP).

You can reconfigure one tile at a time. So you can page in new code modules as needed, like a processor does.

Contact us at info@flex-logix.com to learn more how these capabilities can improve the adaptability and ease of use of your SoCs.