Why the Need for Virtual Arrays?

- **Security:**
  - Keep your “secret sauce” design masked from the customer

- **Preserve Performance & Density:**
  - “Harden” your optimized design for best performance & density on EFLX arrays, even when the rest of the design changes

- **Runtime:**
  - Avoid performing P&R on the entire array when your part of the design is unchanged
Virtual Array Solution

• Have your “secret sauce” IP and customer design on one EFLX array
  ▪ Designs inside black-box tiles are not visible by end-user
  ▪ Remaining EFLX tiles remain user-programmable

• Maintain optimal performance and density of EFLX designs
  ▪ Preserve P&R performance in black-box tiles
  ▪ Black-box bitstream is merged directly into the final bitstream

• Improve EFLX Compiler runtime
  ▪ Designs in black-box tiles are not P&R’ed again

Patent Pending on EFLX Virtual Array Technology
Example: EFLX200K Test Chip in TSMC16FFC

- 7x7 Array
  - 114,240 6-LUTs (~183K LUT4s)
  - 560 22x22 MACs
MCU Virtual Array on EFLX 200K Test Chip

EFLX 200K Test Chip

Customer Design
(33 tiles on a 7x7 array)

MCU Virtual Array
(4x4 blackbox)

SILVACO
Peripherals
& Bus IP

Clock Gen
PLL

System Clk

On demo Board

20-pin JTAG

JTAG, 5

20-pin

JTAG

5

5-pin Header

Clock Gen

SRAM

32KB

Internal Mem Ctrl

ROM

4KB

Internal Mem Ctrl

SPI

FLASH

4

Aux Clk

SRAM

5-pin Header

AHB Lite Channel or Fabric

MCU Accelerator Interfaces

APB Bridge

APB Channel

Timers

GPIO

UART

Watchdog

Customer Design
(33 tiles on a 7x7 array)

Copyright 2016 & 2017 Flex Logix Technologies, Inc.
Virtual Array Definition

Each EFLX array can have one or more black-box virtual arrays

- Each virtual array is a sub-module of the top-level, no direct IO access is needed
- Each virtual array should have its virtual I/Os placed towards the other tiles of the EFLX array
Virtual Array Compilation

Other than pin placement, compiling a 4x4 black-box virtual array is no different from a regular 4x4 array

- EFLX Compiler offers complete netlist-to-bitstream solution
A black-box definition: EFLX Precompiled Macro (.epm)

- Defines black-box essentials: module name, footprint, IO location, clock utilization
- The epm is auto-generated with "--GENERATE VIRTUAL ARRAY" flag
Instantiating a Black-box in Customer Design

The black-box module content is removed from the customer design, and replaced with the .epm inference.

- .epm black box is automatically preserved through Synplify
Running Black-box Design in EFLX Compiler

Load a top-level floorplan with the black-box tiles labeled:

- EFLX Compiler connects top-level design with black-box design during P&R
- No user visibility into what’s inside the black-box
Running Blackbox Designs in Silicon

Black-box MCU (CM0) executes the AES enc/decryption in SW
Custom accelerator executes the same enc/decryption in HW

HW encryption accelerator is 31.5x faster than SW
(ARM Cortex-M0)

HW decryption accelerator is 236.5x faster than SW
(ARM Cortex-M0)
Conclusion: Virtual Array Benefits

- Virtual array is a simple way to insert a pre-compiled black-box into the customer design
- **Secure**: User has no visibility into the black-box design
- **Predictable**: Pre-compiled design is frozen, preserves PPA
- **Fast**: Pre-compiled design portion is not P&R’ed again
- **Flexible**:
  - Maintain user-programmability in the rest of the designs
  - User can choose to instantiate the black-box or not
  - Updates to .epm can be downloaded as an image file
- One of many ways to build a flexible, secure, and high-quality EFLX IP ecosystem

*Patent Pending on EFLX Virtual Array Technology*