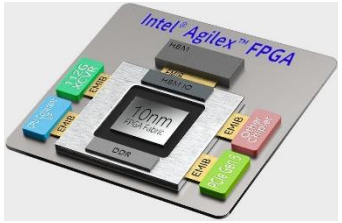
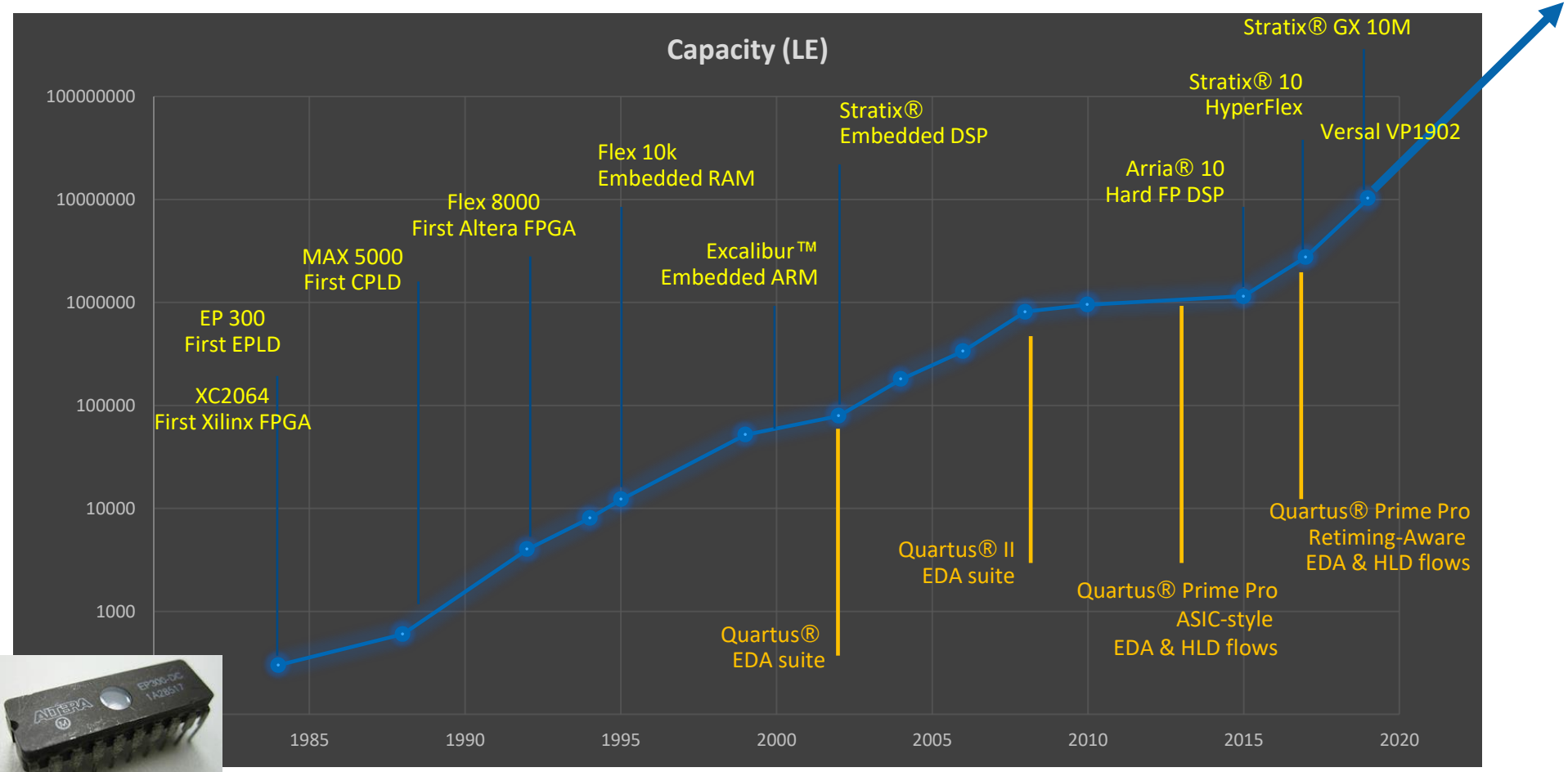
# Grand Challenges for FPGAs in the Next Decade

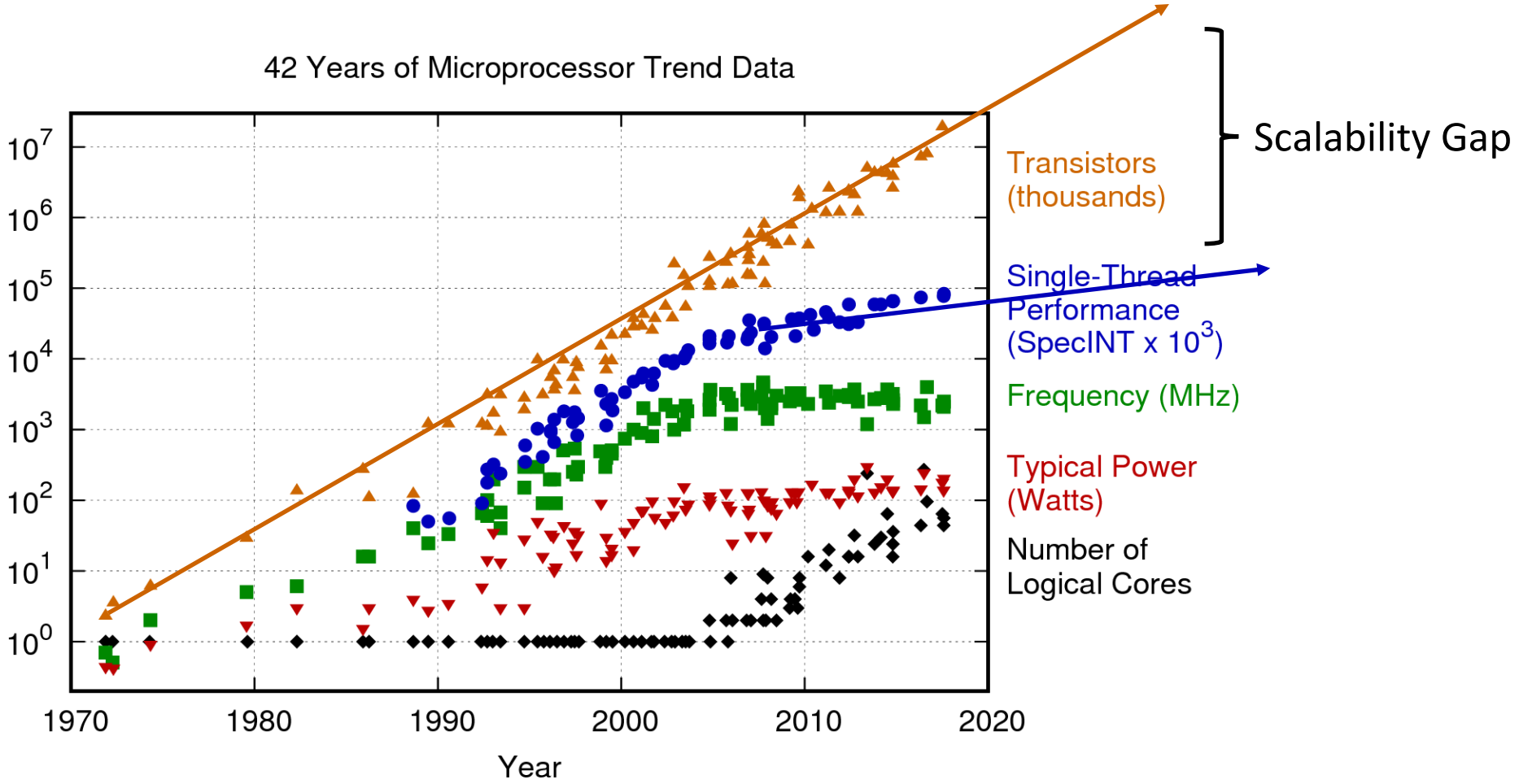

Nabeel Shirazi
Senior Director of Quartus Software
Altera

# Silicon Scale



# Silicon Size vs CPU Performance



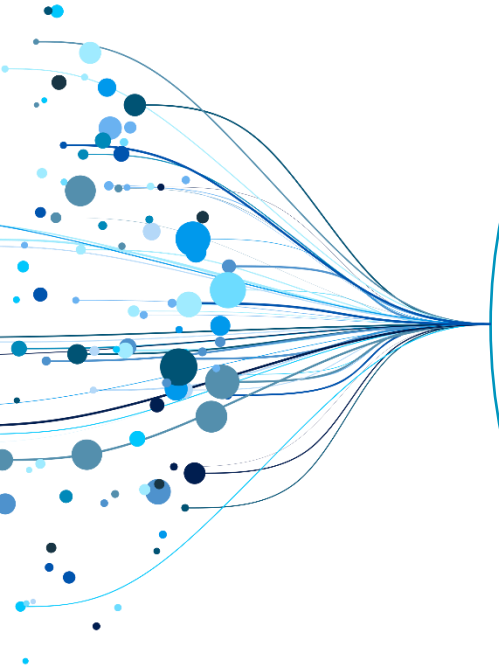

Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

# Compute Requirements

- In 2023, Training of GPT-3 took:
  - 6 FLOPs per parameter
  - 175B Parameters
  - 300B Tokens
  - 6 x 175B x 300B = 3.15 x 1023 FLOPs to train
  - ~2000 H100 GPUs @ 50% utilization
  - **Training time: ~3 weeks[1]**

- Compile for today's 20M Logic Elements (LE) device
  - 4 dies, thus 5M LEs per die
  - Currently Place & Route: **120min for 50K LEs** on a 4-core x86 CPU
  - 200 hours/die = 8 days
  - At least 3 iterations
  - **Compile time: ~3 weeks**
  - How long will it take in the future?

[1] https://blog.apnic.net/2023/08/10/large-language-models-the-hardware-connection/



Growing Complexity
**meets**
**Innovation**

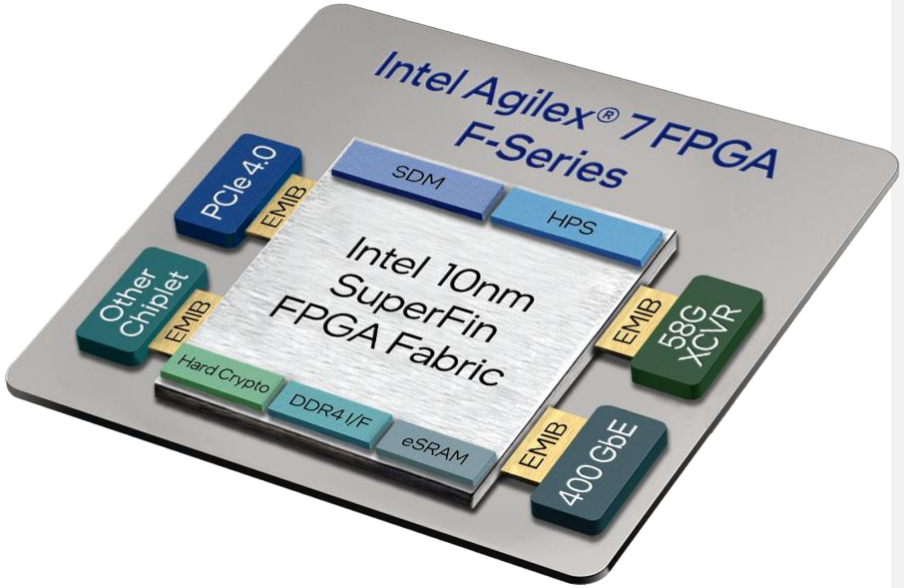
1 Silicon Scale

2 EDA

3 Applications

# Today's Silicon Complexity

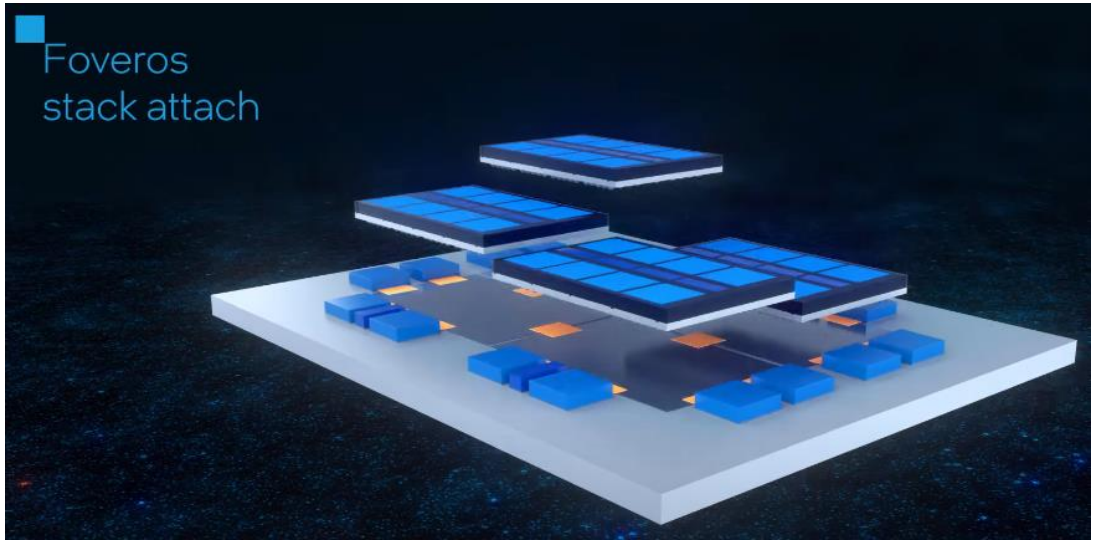
- Mix of Analog and Digital Components, e.g., A2Ds, Gigabit Transceivers
- Different Types of Compute:
  - Traditional FPGA Fabric
  - Hardened Processors w/ SIMD Extensions
  - Soft Processors (Proprietary & RISC-V) w/ Custom Instructions
  - Array of Vector Processors
  - DSP Blocks w/ Variable Precision FP and AI Tensor Blocks
- Memory Hierarchy
  - On-chip Local Memory
  - DDR
  - HBM
- Data Movement via an NoC



# 2.5D and 3D FPGAs

- Today:  2.5D Chiplet Tech.
  - 2D Planar with Silicon Interposer
  - 1st multi-die FPGA over 10 years old
  - Requires a state-of-the-art Design Partitioner
  - Latest 2.5D technology includes:
    - Die to Die Connectivity with embedded multi-die interconnect bridge (EMIB)
    - Standard communication protocol is UCIe with speeds of 10 - 12 gigabits/sec.

- Future: 3D FPGAs?
  - Intel's 3D packaging is Foveros
  - Example: Ponte Vecchio GPU w/ 47 tiles!
  - Much Higher Density & Performance, with less Power



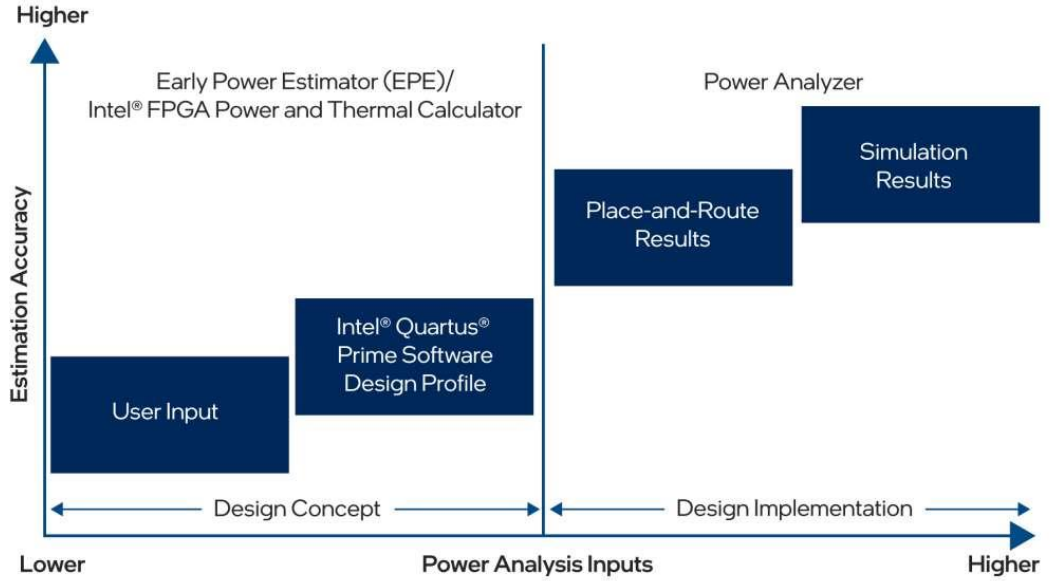

https://www.intel.com/content/www/us/en/newsroom/news/intel-accelerated-webcast-livestream-replay.html#gs.45k5vb

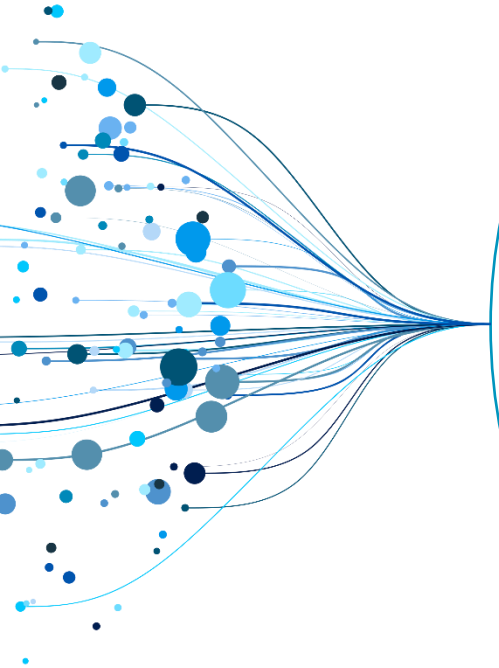
# Power Consumption

- Why is power important?
  - Data Center:  Power Limited, Carbon Footprint & Cooling
  - Edge: Battery Life, Weight & Cooling
- Challenges:
  - FPGAs consume significantly more dynamic power[1] than an ASIC
  - Voltage transients & load surge
  - Maintaining signal integrity due to faster transceiver rates

[1] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4068926

- Improving FPGA Power via EDA
  - Mix of LVT & HVT cells that EDA can choose
  - Dynamic Voltage and Frequency Scaling
  - Upfront estimation, planning, and PR



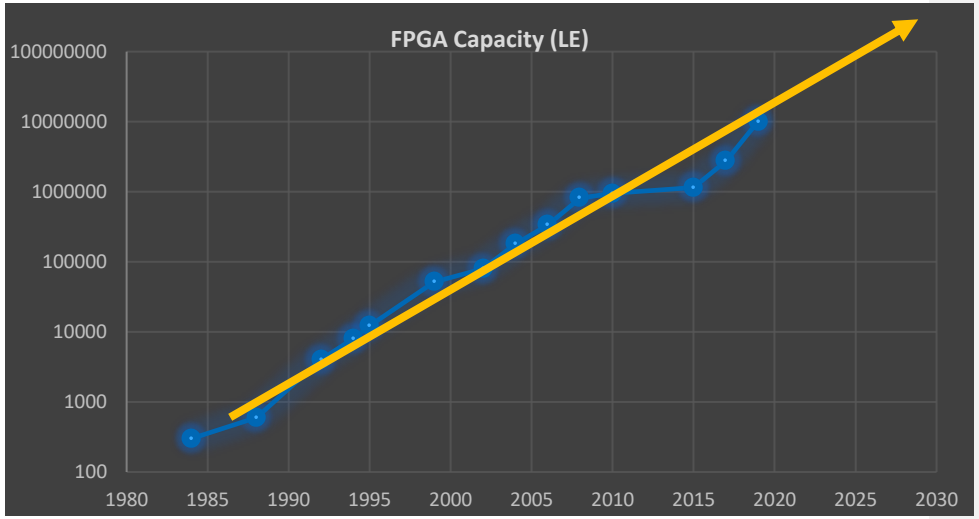


Growing Complexity
**meets**
**Innovation**

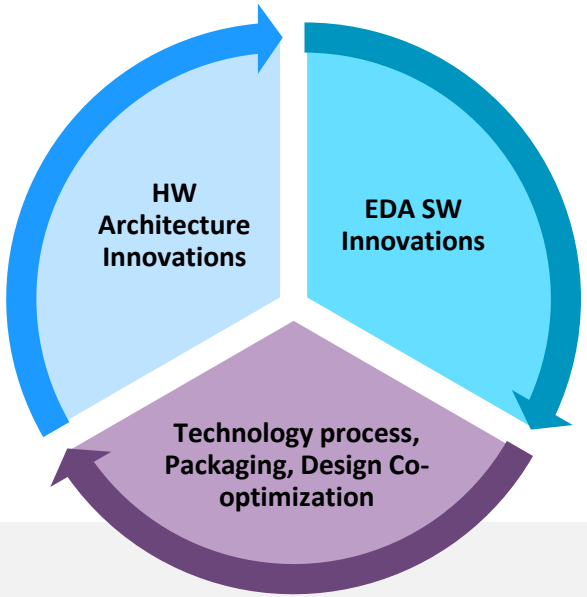
| | |
|---|---|
| 1 | Silicon Scale |
| 2 | EDA |
| 3 | Apps run seamlessly from cloud to edge |

# The Scaling Challenge

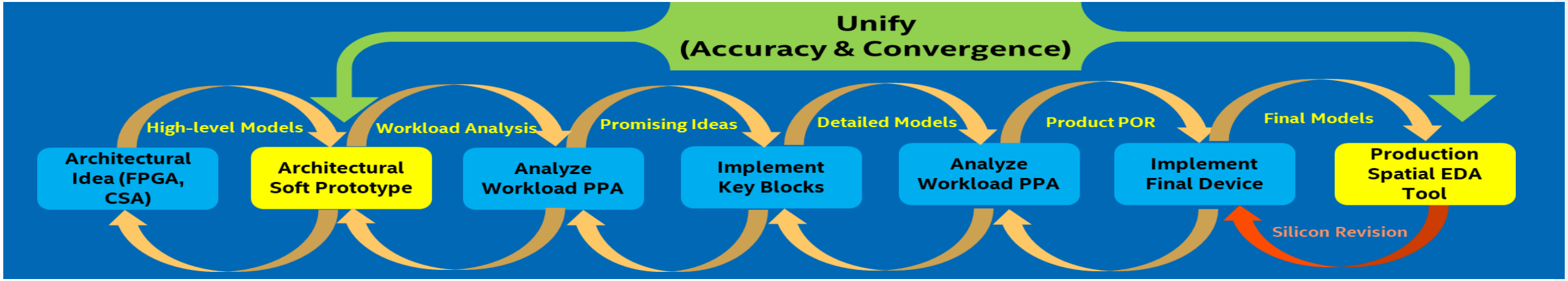
- **FPGAs reaching 100's millions of logic elements in the next decade**
  - How do we continue to scale compile time?
  - How do we improve Design Entry?
  - How do users achieve system-level performance?




- **Co-design and co-optimization of:**
  - HW architecture, technology process, and packaging
  - EDA software
  - Workload Analysis: overall system performance, power, flexibility, and debuggability



# FPGA HW & EDA Compiler Design Cycle




- **Quality of Results (QoR) Optimization**
  - Innovate **advanced EDA Compiler** technologies in SW
  - Process tuning
  - Silicon revision

## Advanced EDA Compiler Features

- c-cycle global and incremental retiming
- Retiming-aware CAD for technology mapping, placement, routing
- Physical clock allocation, sector-based clock routing, and l shift register inferencing
- Rewind verification for retimed circuits
- State-of-the-art placement, clustering, routing, physical synthesis engines

➢ ***Significant improvements in Fmax, wire usage, routing congestion, logic utilization***

# Three Vectors to Address Compile Time

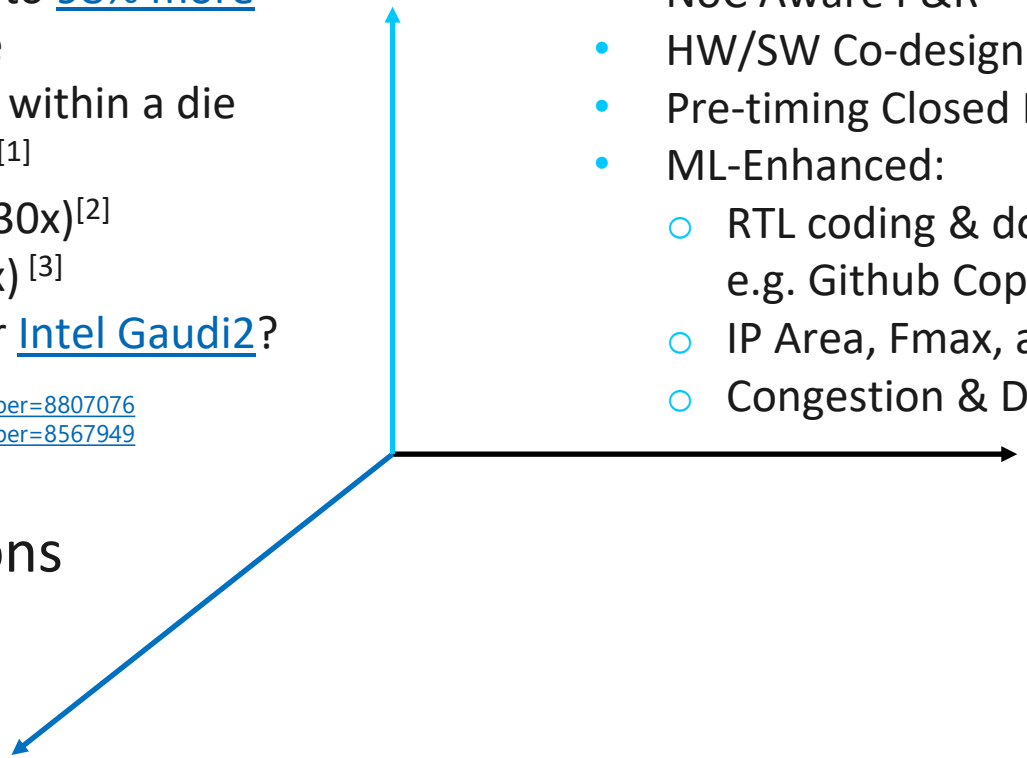
## Hardware Acceleration

- We must harness the power of the cloud
- Cloud infra can be between 72% to 98% more carbon efficient than on-premise
- Distributed compiles per die and within a die
- FPGA Acceleration for FPGA EDA[1]
- GPU acceleration of Placement (30x)[2]
- GPU acceleration of Routing (21x) [3]
- AI ASICs, e.g., AWS Inferentia2 or Intel Gaudi2?

[1] https://ieeexplore.ieee.org/document/8892149
[2] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8807076
[3] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8567949

## Acceleration of a single compile

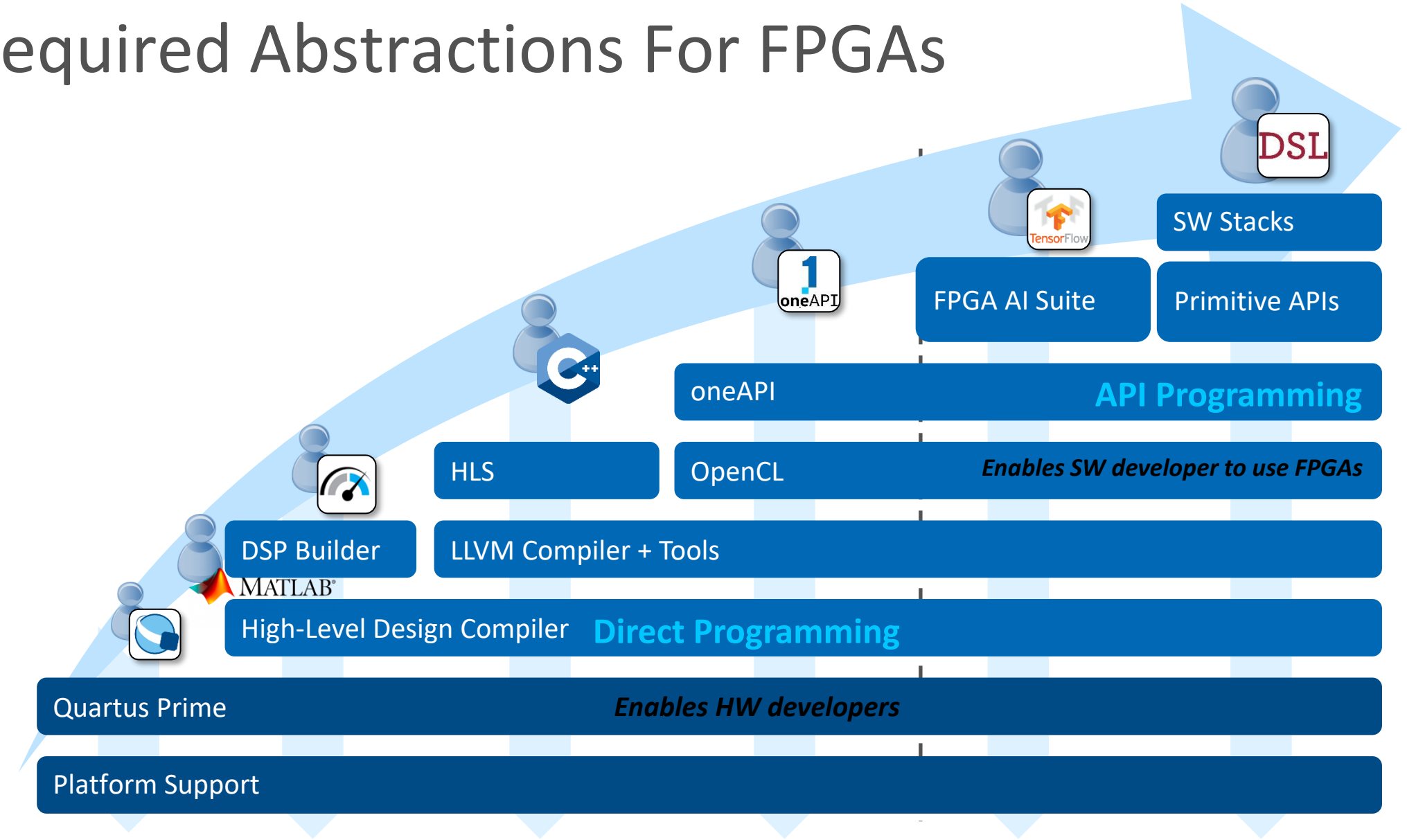
- Fundamental P&R Algorithms
- NoC Aware P&R
- HW/SW Co-design to improve PPA
- Pre-timing Closed IP and Overlays
- ML-Enhanced:
  - RTL coding & documentation assistance, e.g. Github Copilot Chat
  - IP Area, Fmax, and Power estimates
  - Congestion & Delay estimates



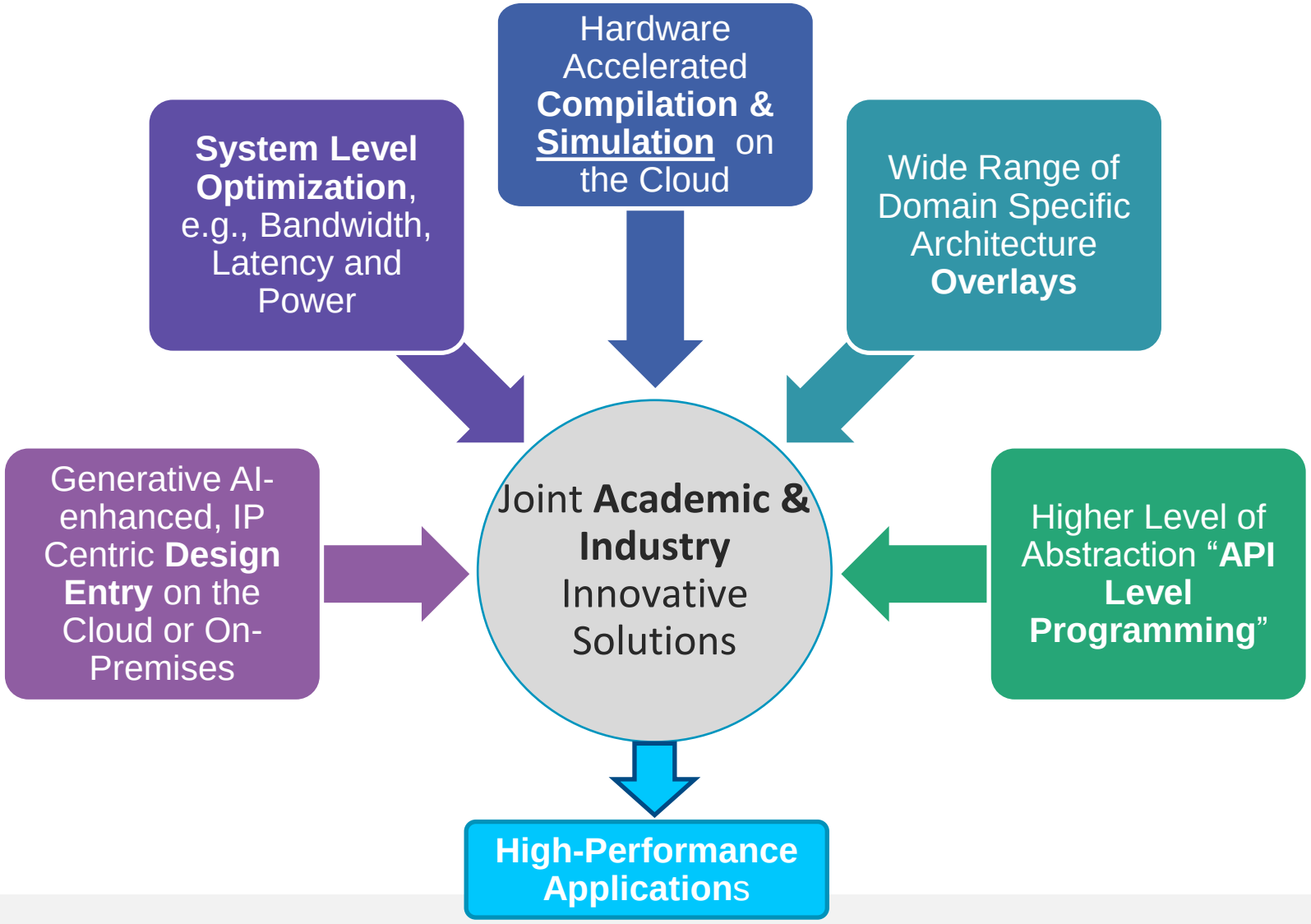
## Fewer Timing Closure Iterations

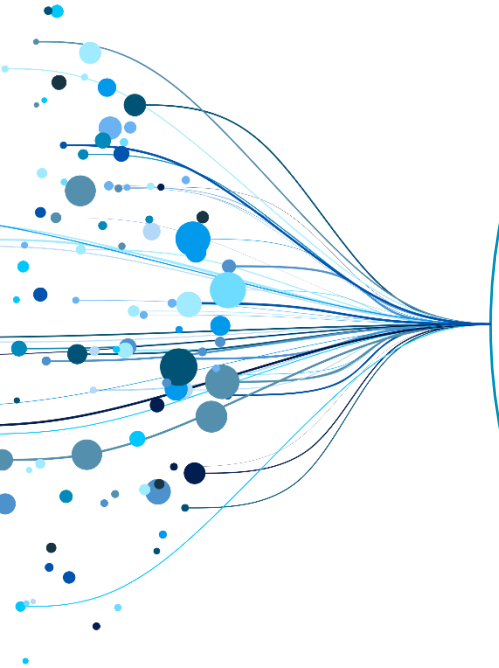
- Function of timing constraints
- ML Based Selection of Strategies
- Incremental Compile
- User guidance on how to optimize source code based on multiple runs

# Required Abstractions For FPGAs



# Evolution of FPGA Design





Growing Complexity
**meets**
**Innovation**

1 Silicon Scale

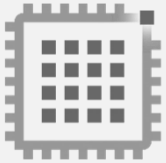
2 EDA

3 Applications

# Need for more Flexibility and Agility

**Market challenge:**

Hardware-driven,
**fixed-function** switch ASICs



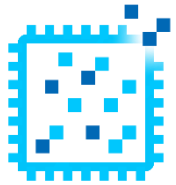
Unable to keep pace with **changing demands**

It takes **years** and **a lot of money** before new features can be enabled

Difficult to **diagnose issues** once in the field

**Solution:**

Software-driven,
**programmable** devices



Customization enables rapid innovation and **differentiation to be Special**

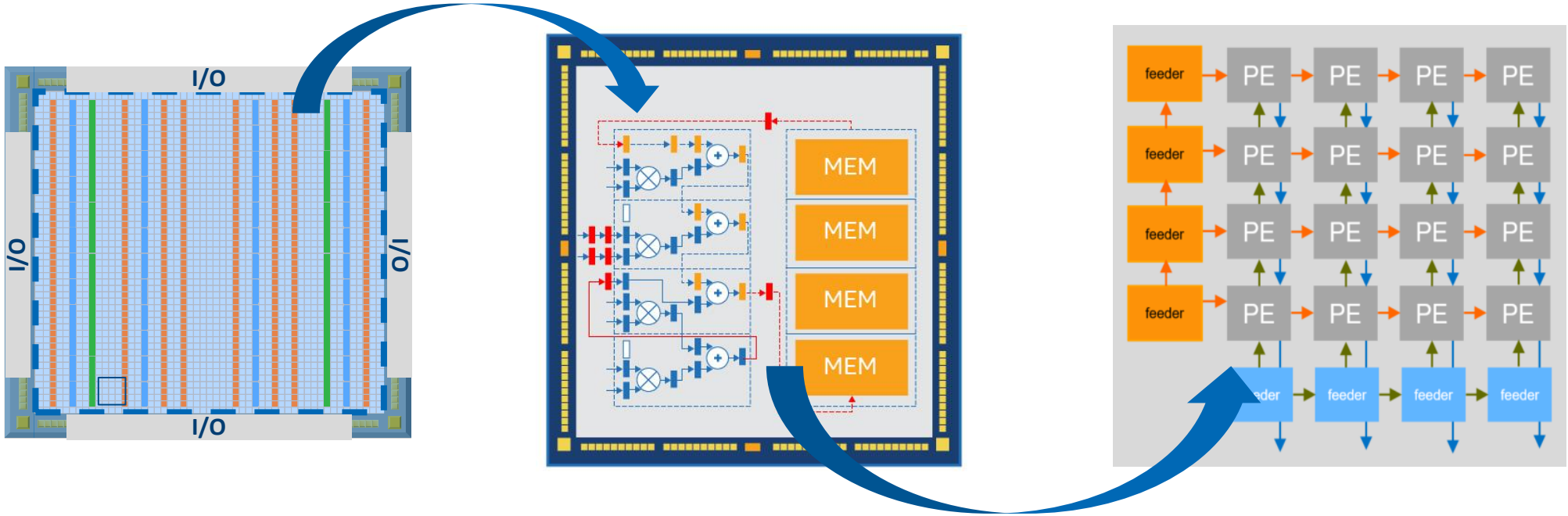
The key is to deploy new hardware as **fast as software dev. cycle**

**Visibility** into performance issues with greater potential for **future optimization**

# FPGAs Enable Deep Learning

- Programmable Logic
- Memories w/NoC
- DSPs w/ Tensor Blocks[1]:
  $(a \cdot b = \sum_{i=1}^{10} a_i b_i)$
- Transceivers

- Programmable Datapath
- Customized Memory Structure
- Configurable Compute

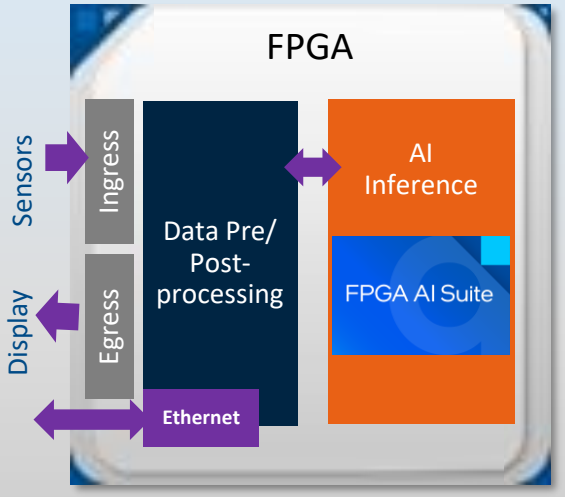



*FPGAs have the ingredients needed to build Deep Learning compute engines*

[1] https://dl.acm.org/doi/10.1145/3431920.3439293 - FPGA'21'

# FPGA Flexibility for Implementing AI

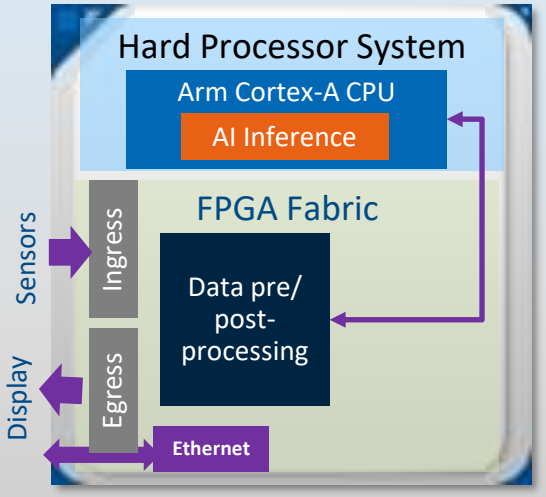
## FPGA FABRIC




FPGA based
AI Inference

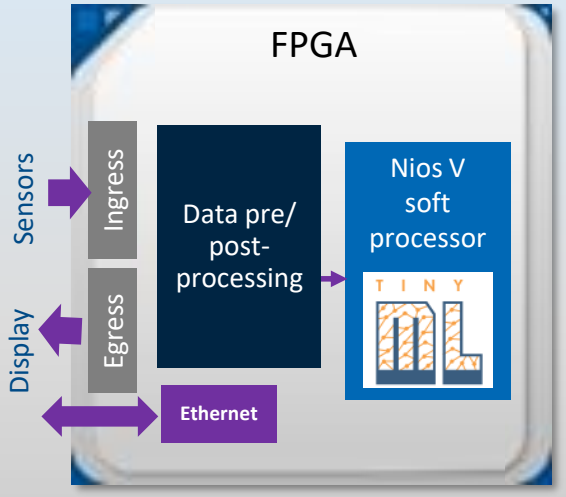
using FPGA AI Suite

## INTEGRATED ARM® SoC FPGA



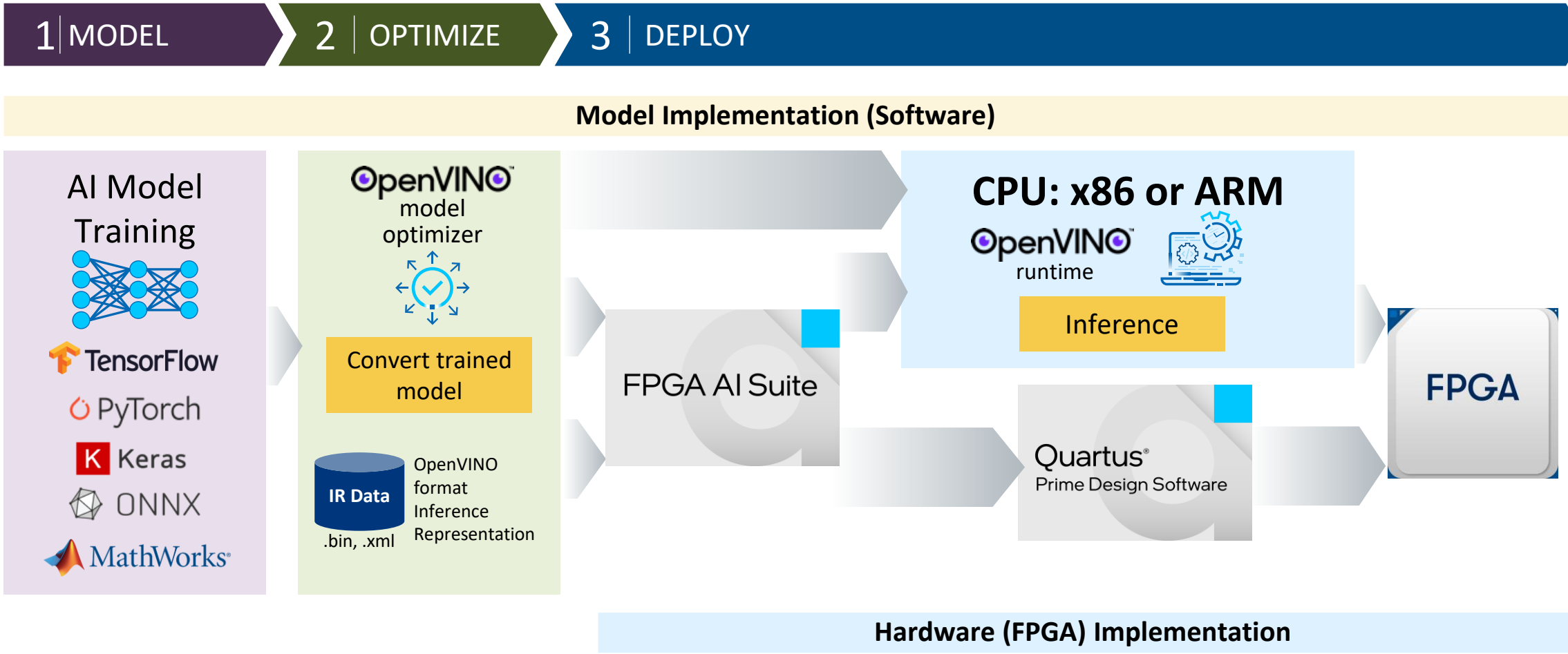

SoC FPGA
Arm® CPU based
AI Inference

## SOFT CPU




RISC-V ISA
soft-CPU based
AI Inference

# FPGA AI Suite Development Flow



# Summary

- FPGA silicon is constantly evolving to deliver more capacity and functionality with better performance
- Investing in EDA is crucial for managing silicon complexity and boosting user productivity
- FPGAs will be the cornerstone of many applications, including wireless communications and ML applications
- Altera will continue to invest in academic research to accelerate innovation to find solutions with you