Open-Source FPGA Development

Can **Anyone** Build an FPGA Now?

FPT’23 Conference

December 14, 2023

Who I Am

- 25+ years of experience in the FPGA industry
  - 7+ years President and CEO
  - 20+ years Engineering, Product Line Mgmt, Marketing, Sales
- Board Director of the Global Semiconductor Alliance (GSA)
- Founding Board Director of the Open-Source FPGA Foundation
- Patent in Configurable Computational Unit Embedded in Programmable Device
- B.S. Computer Engineering from Santa Clara University
- Previously, an Adjunct Lecturer at Santa Clara University for Programmable Logic course

Thank You To All the

Open-Source

Developers

QuickLogic – All Things FPGA

- Founded in 1988
- Headquartered in Silicon Valley

Can we become the “RedHat” of FPGA Technology?

Trivia Question:

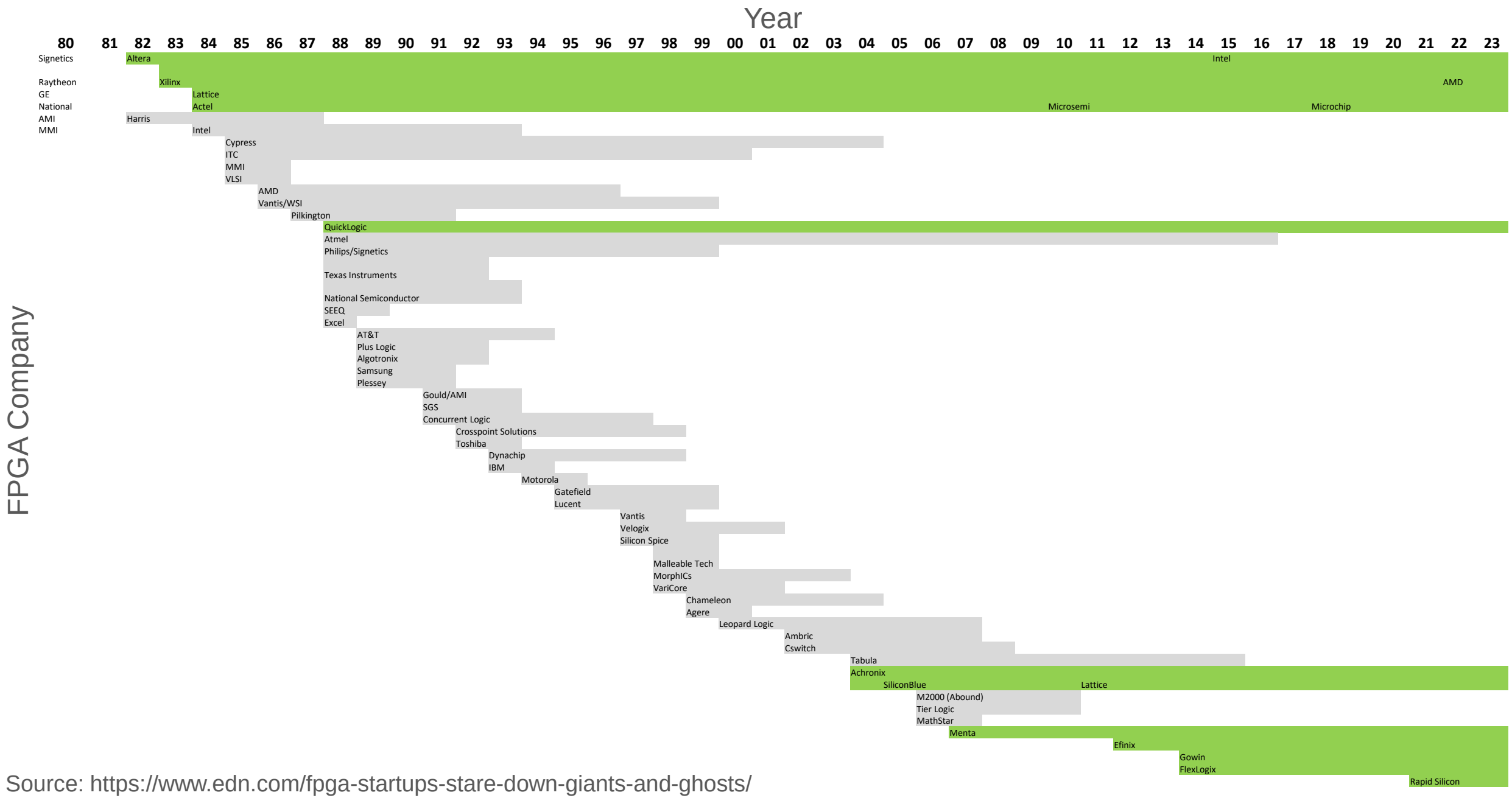
How Many Companies Have Tried to Build FPGAs...?

…and are defunct?

slido

How many companies have tried to develop and sell FPGAs and failed?

ⓘ Start presenting to display the poll results on this slide.



Source: https://www.edn.com/fpga-startups-stare-down-giants-and-ghosts/

An FPGA Survivor

~50M units shipped, **>20** ***production*** FPGA tape-outs

~2,000 customers

8 foundries

13 process nodes

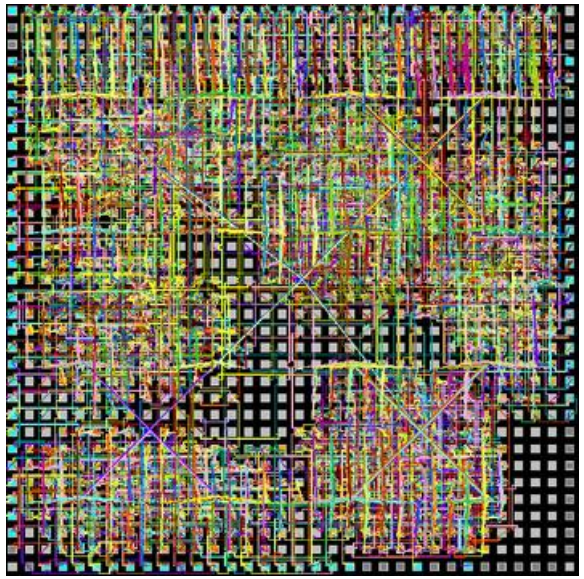
6/7nm*, 12nm, 16nm, 22nm, 28nm, 40nm, 65nm, 90nm, 130nm, 180nm, 0.25µm, 0.35µm, 0.65µm

Bulk CMOS, FDSOI, Radiation-Hardened

500+ person years invested in our FPGA architecture and software

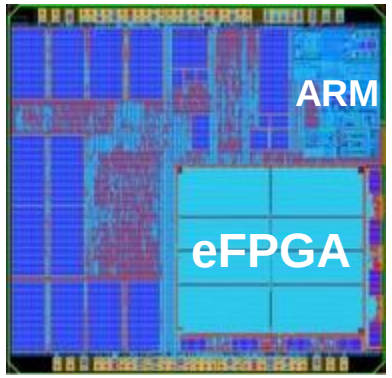
From eFPGA IP to FPGAs to FPGA-Based SoCs

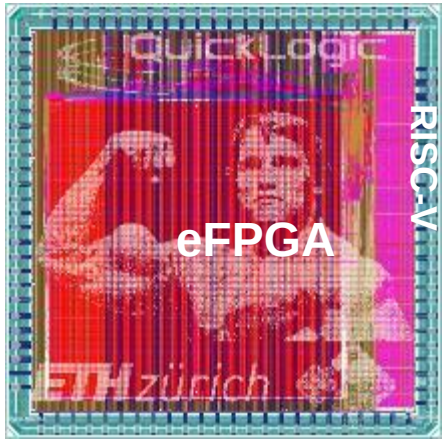
eFPGA IP
Process: Multiple



Low Power FPGAs
Process: Multiple

FPGA-Based SoCs
Process: 40nm and 22nm





Why So Difficult?

Trivia Question:

Is the Challenge Software or Silicon?

slido

Is the challenge FPGA User Tools Software or Silicon Design?

ⓘ Start presenting to display the poll results on this slide.

Bringing Open-Source to FPGA Technology

Google

UNIVERSITY OF TORONTO

CHIPS ALLIANCE

OPENHW GROUP™ PROVEN PROCESSOR IP

SymbiFlow

CROWD SUPPLY

QuickLogic®

RENODE™

ETH zürich

THE UNIVERSITY OF UTAH

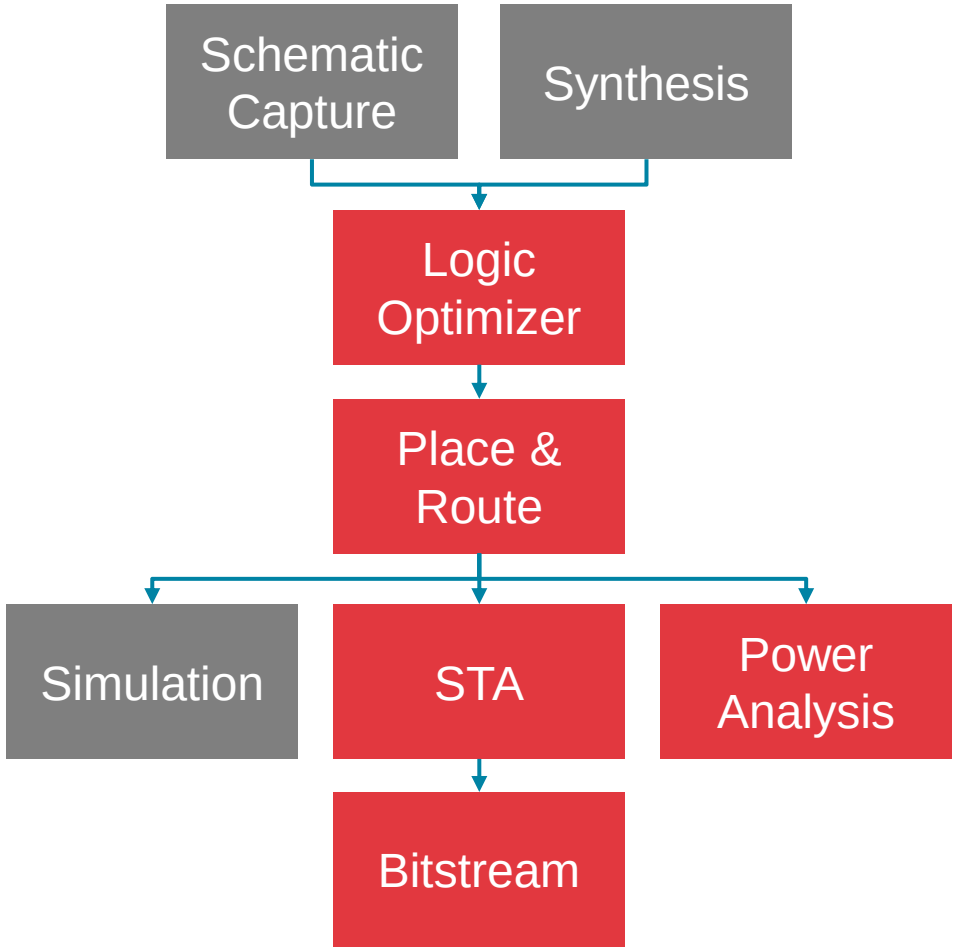
antmicro EMBEDDED SYSTEMS

FOSSi Foundation

Software:
Proprietary FPGA
User Tools


THE Walled Garden

Proprietary FPGA Tools 25 Years Ago…



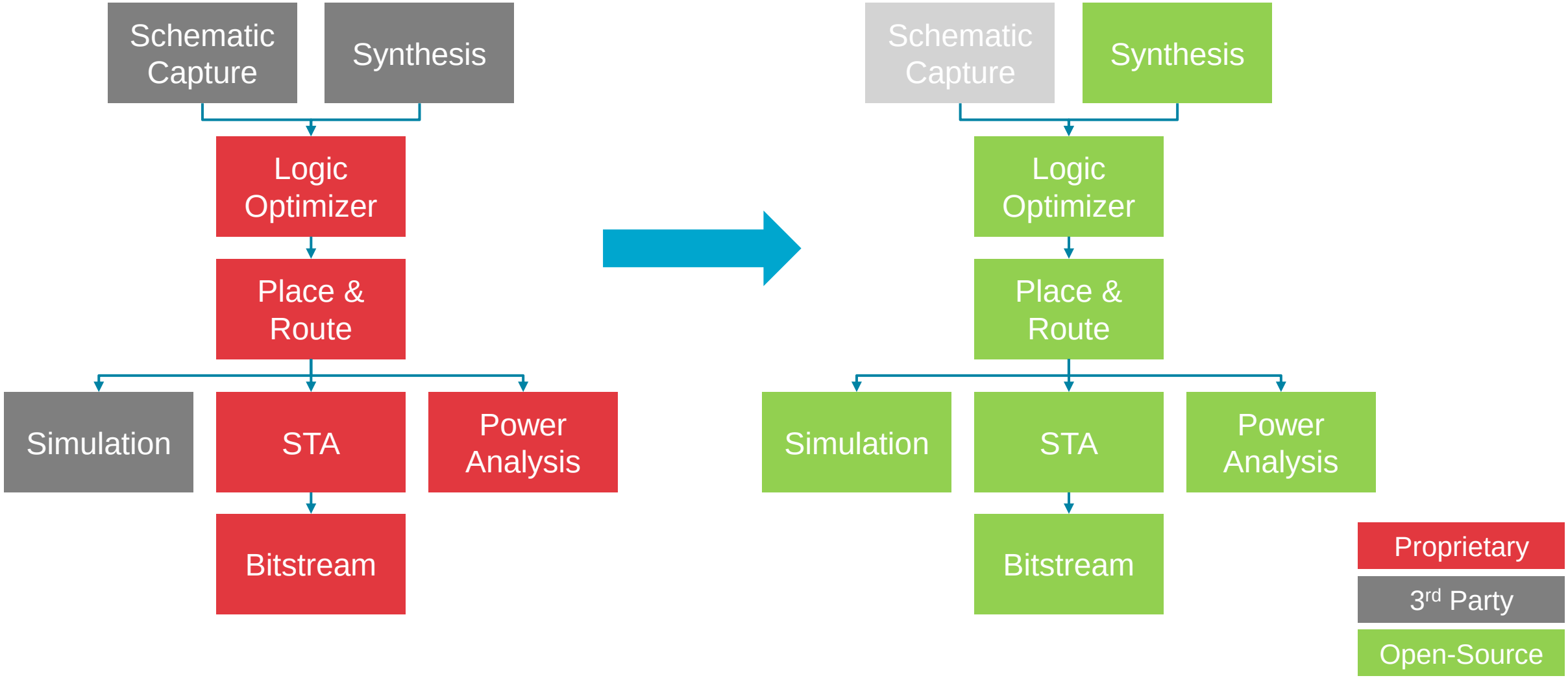
Proprietary

3rd Party

Open-Source

Software:
Open-Source FPGA
User Tools

Options for Open-Source FPGA User Tools Today…

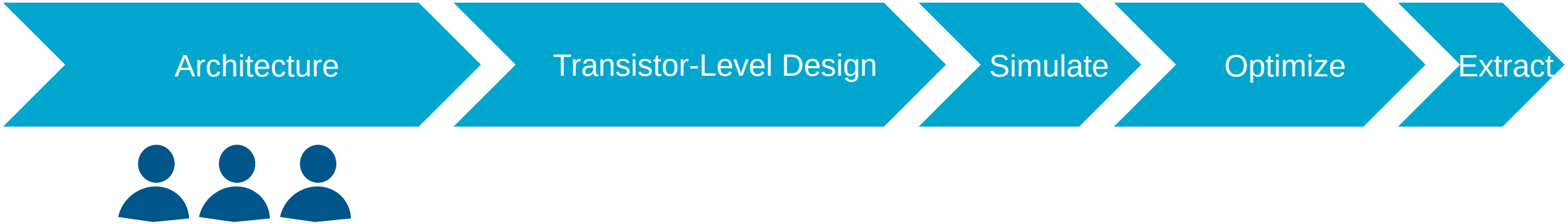


FPGA Silicon: The “Old Way” of Design

Choose Configuration Memory

Technology Attributes	Reprogrammable	One-Time Programmable (OTP)
Non-Volatile	Flash, MRAM, ReRAM	Antifuse
Volatile	SRAM	n/a

Model Architecture



1+ Years

Develop on Target Foundry / Process

1+ Years

Simulate – Optimize – Extract

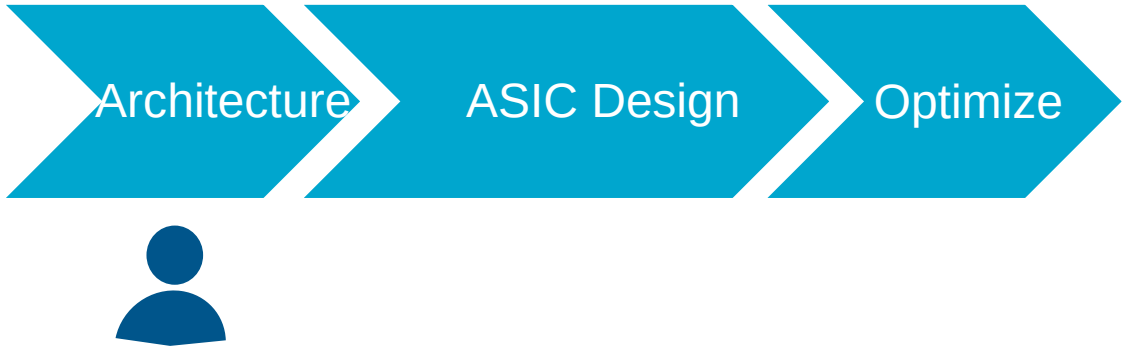
1/2+ Year

Result – Entire Team

1-2+ Years

FPGA Silicon: Open-Source, Automated Workflows

Model Architecture



1+ Months

Develop on Target Foundry / Process

2+ Months

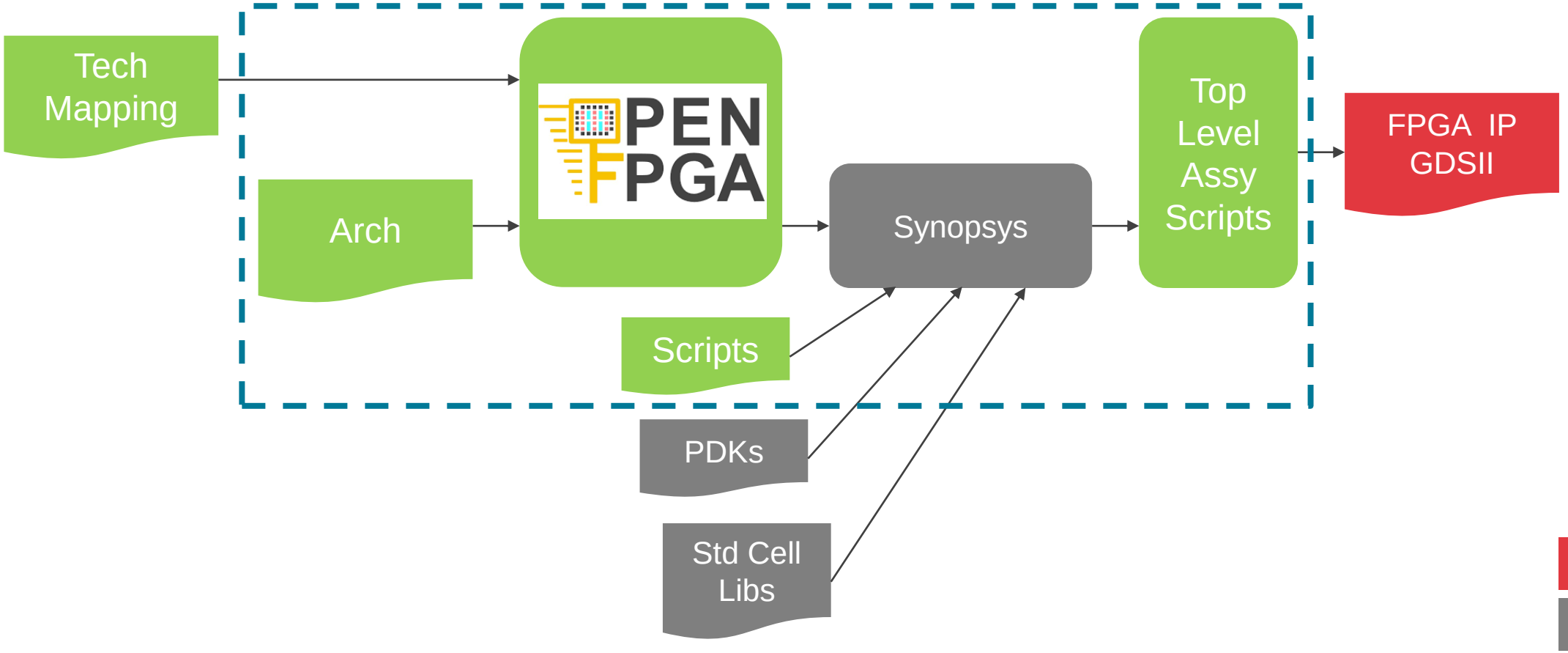
Optimize FPGA Core

1+ Months

Result – Entire Team

4+ Months

OpenFPGA Development Workflow



Proprietary

3rd Party

Open-Source

If Open-Source Tools Exist, Why Not Adopt Them?

Quality of Results

Humans like Routine

Change is Painful

Change is Risk

Forcing Function for Change

Is data the new oil?

No, Metadata is.

Data itself isn’t as valuable as the information latent in the data.

Metadata is the
“Data about Data”.

A DAY IN DATA

The exponential growth of data is undisputed, but the numbers behind this explosion – fuelled by internet of things and the use of connected devcies – are hard to comprehend, particularly when looked at in the context of one day

500m

tweets are sent every day

Twitter

4PB

of data created by Facebook, including

350m photos

100m hours of video watch time

Facebook Research

294bn

billion emails are sent

Radicati Group

320bn

emails to be sent each day by 2021

306bn

emails to be sent each day by 2020

3.9bn

people use emails

4TB

of data produced by a connected car

Intel

DEMYSTIFIYING DATA UNITS

From the more familiar 'bit' or 'megabyte', larger units of measurement are more frequently being used to explain the masses of data

Unit		Value	Size
b	bit	0 or 1	1/8 of a byte
B	byte	8 bits	1 byte
KB	kilobyte	1,000 bytes	1,000 bytes
MB	megabyte	$1{,}000^2$ bytes	1,000,000 bytes
GB	gigabyte	$1{,}000^3$ bytes	1,000,000,000 bytes
TB	terabyte	$1{,}000^4$ bytes	1,000,000,000,000 bytes
PB	petabyte	$1{,}000^5$ bytes	1,000,000,000,000,000 bytes
EB	exabyte	$1{,}000^6$ bytes	1,000,000,000,000,000,000 bytes
ZB	zettabyte	$1{,}000^7$ bytes	1,000,000,000,000,000,000,000 bytes
YB	yottabyte	$1{,}000^8$ bytes	1,000,000,000,000,000,000,000,000 bytes

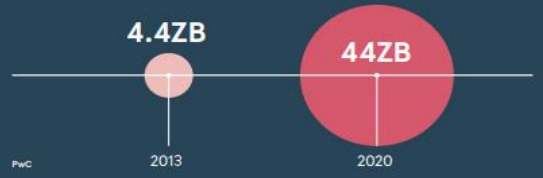
*A lowercase "b" is used as an abbreviation for bits, while an uppercase "B" represents bytes.

65bn

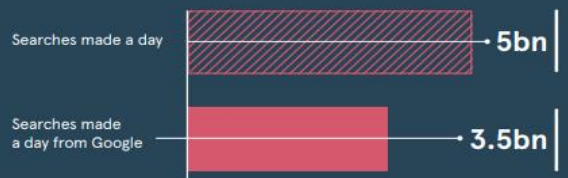
messages sent over WhatsApp and two billion minutes of voice and video calls made

Facebook

ACCUMULATED DIGITAL UNIVERSE OF DATA



PwC



Smart Insights

463EB

of data will be created every day by 2025

IDC

95m

photos and videos are shared on Instagram

Instagram Business

28PB

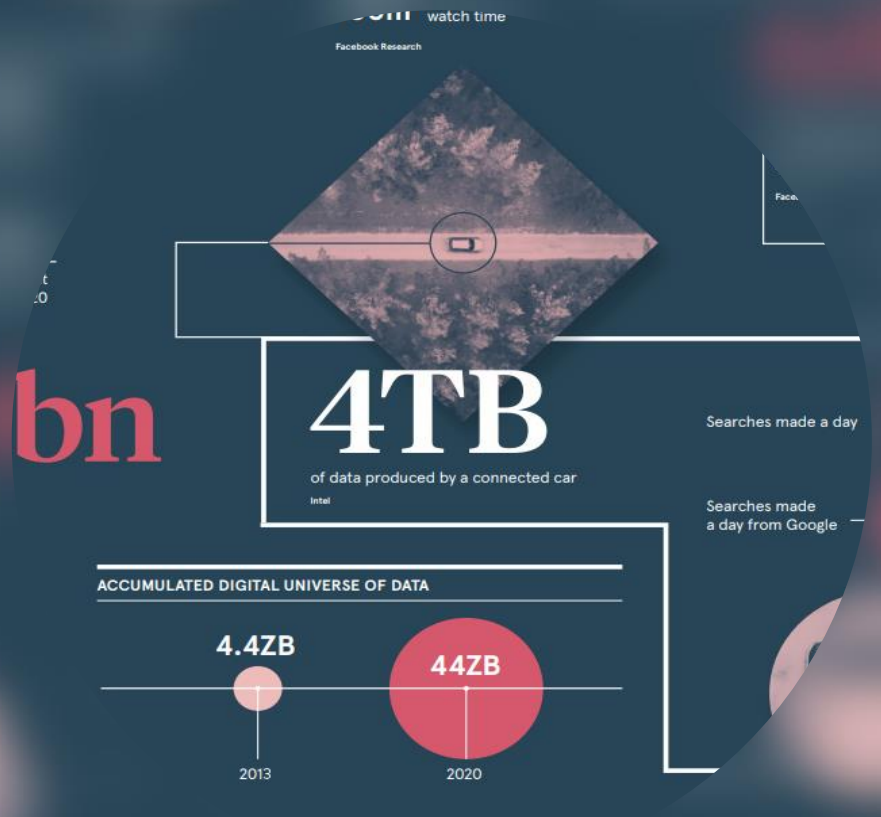
to be generated from wearable devices by 2020

Statista

RACONTEUR

“Expectations are for tens or hundreds of billions of devices over the next few years.”

Peter Warden, Google

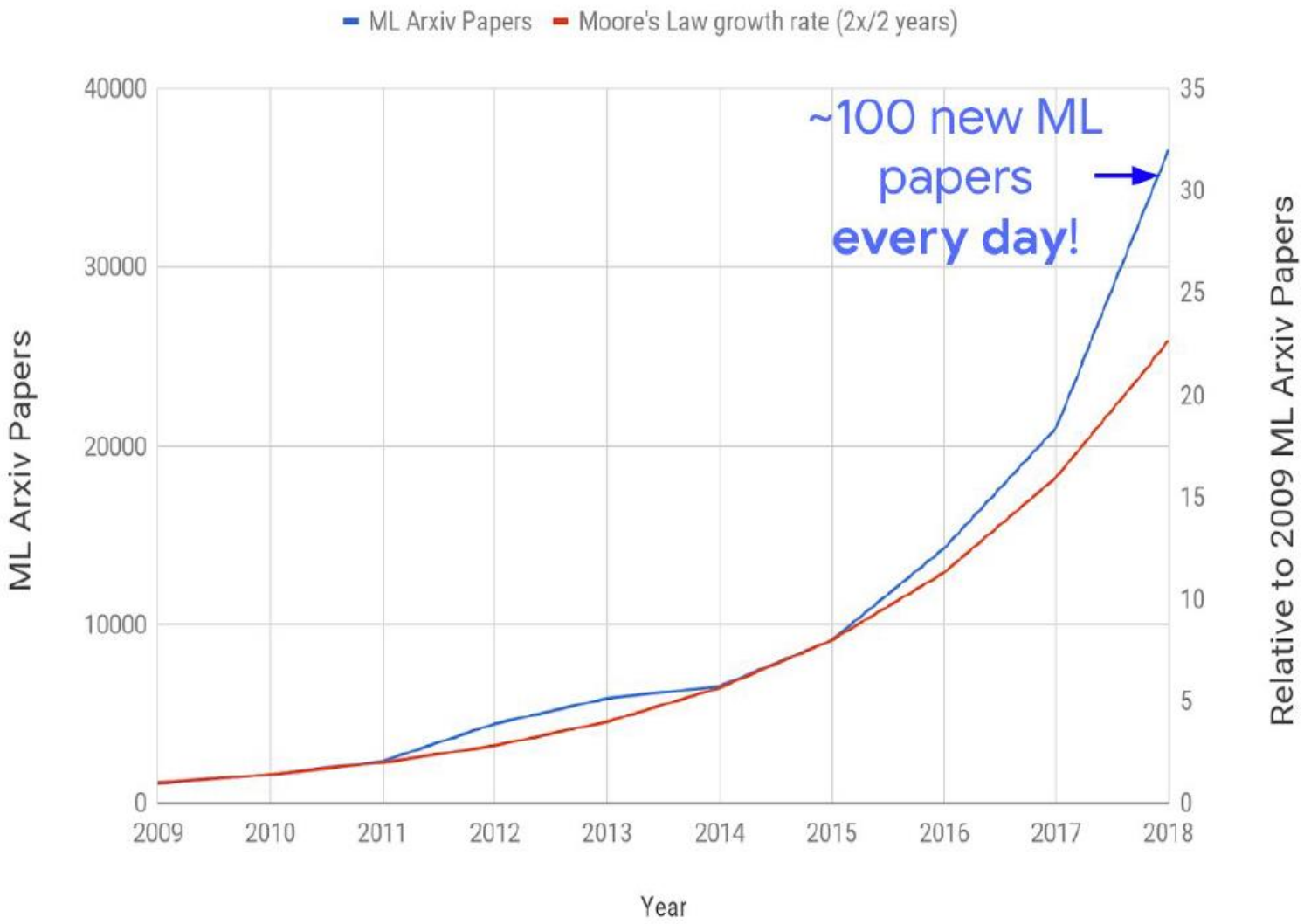


How Do We Scale the Metadata Economy?

Machine Learning
Transforms Data into
Metadata

Examples of Machine Learning Making an Impact

Explosive Growth of Machine Learning Research



Source: https://arxiv.org/ftp/arxiv/papers/1911/1911.05289.pdf, Jeffrey Dean, Google Research

What Will Unleash the Potential of ML?

Technology Stack for AI

The technology stack for artificial intelligence (AI) contains nine layers.

Technology	Stack	Definition
Services	**Solution and use case**	Integrated solutions that include training data, models, hardware, and other components (eg, voice-recognition systems)
Training	**Data types**	Data presented to AI systems for analysis
Platform	**Methods**	Techniques for optimizing weights given to model inputs
	Architecture	Structured approach to extract features from data (eg, convolutional or recurrent neural networks)
	Algorithm	A set of rules that gradually modifies the weights given to certain model inputs within the neural network during training to optimize inference
	Framework	Software packages to define architectures and invoke algorithms on the hardware through the interface
Interface	**Interface systems**	Systems within framework that determine and facilitate communication pathways between software and underlying hardware
Hardware	**Head node**	Hardware unit that orchestrates and coordinates computations among accelerators
	Accelerator	Silicon chip designed to perform highly parallel operations required by AI; also enables simultaneous computations

Memory
- Electronic data repository for short-term storage during processing
- Memory typically consists of DRAM[1]

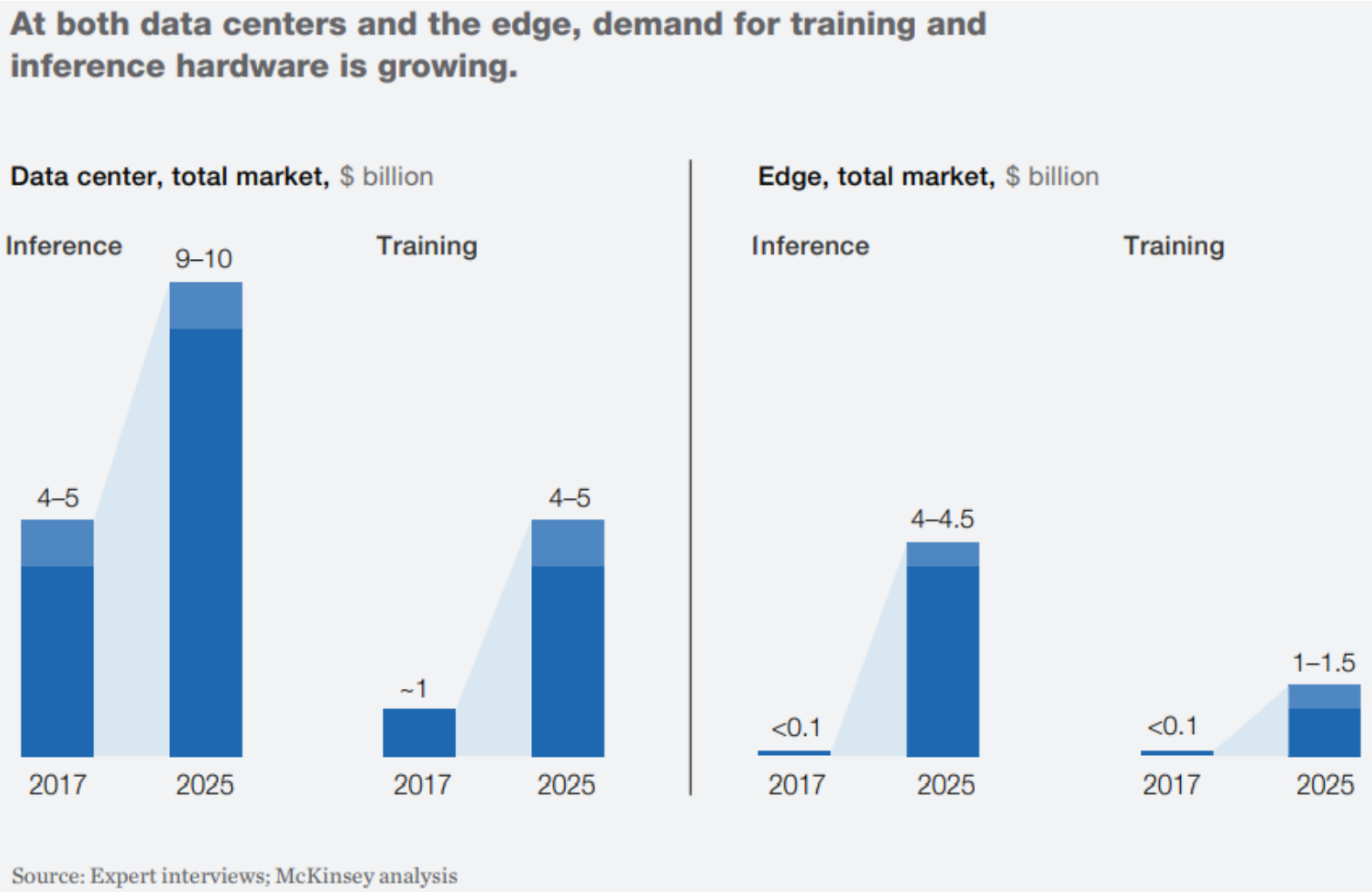
Storage
- Electronic repository for long-term storage of large data sets
- Storage typically consists of NAND[2]

Logic
- Processor optimized to calculate neural network operations, ie, convolution and matrix multiplication
- Logic devices are typically CPU, GPU, FPGA, and/or ASIC[3]

Networking
- Switches, routers, and other equipment used to link servers in the cloud and to connect edge devices

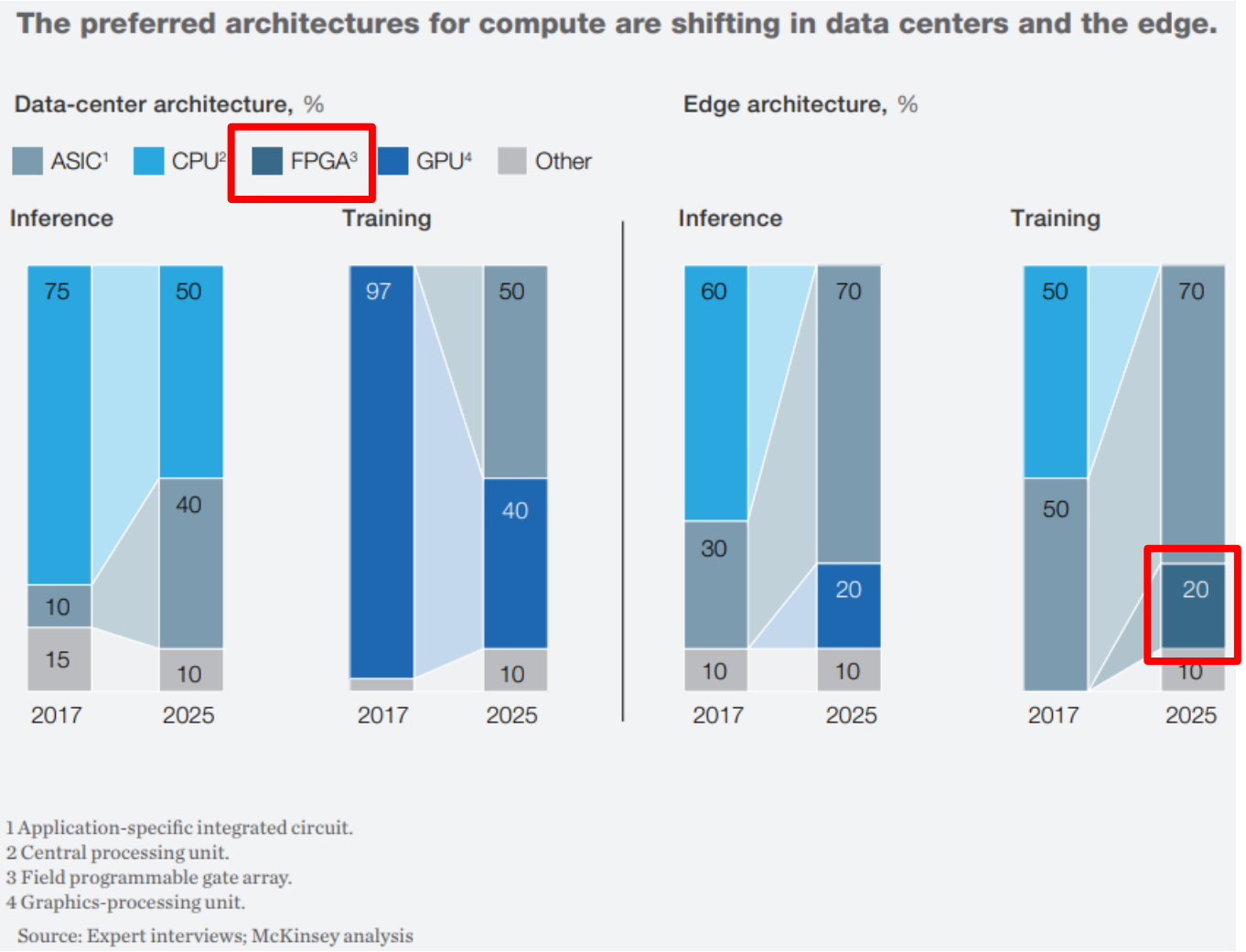
Source: Artificial-intelligence hardware: New opportunities for semiconductor companies, McKinsey

Multi-Billion $ Semiconductor Opportunity



Source: Artificial-intelligence hardware: New opportunities for semiconductor companies, McKinsey

Customization Driving Need From Standard to Custom?

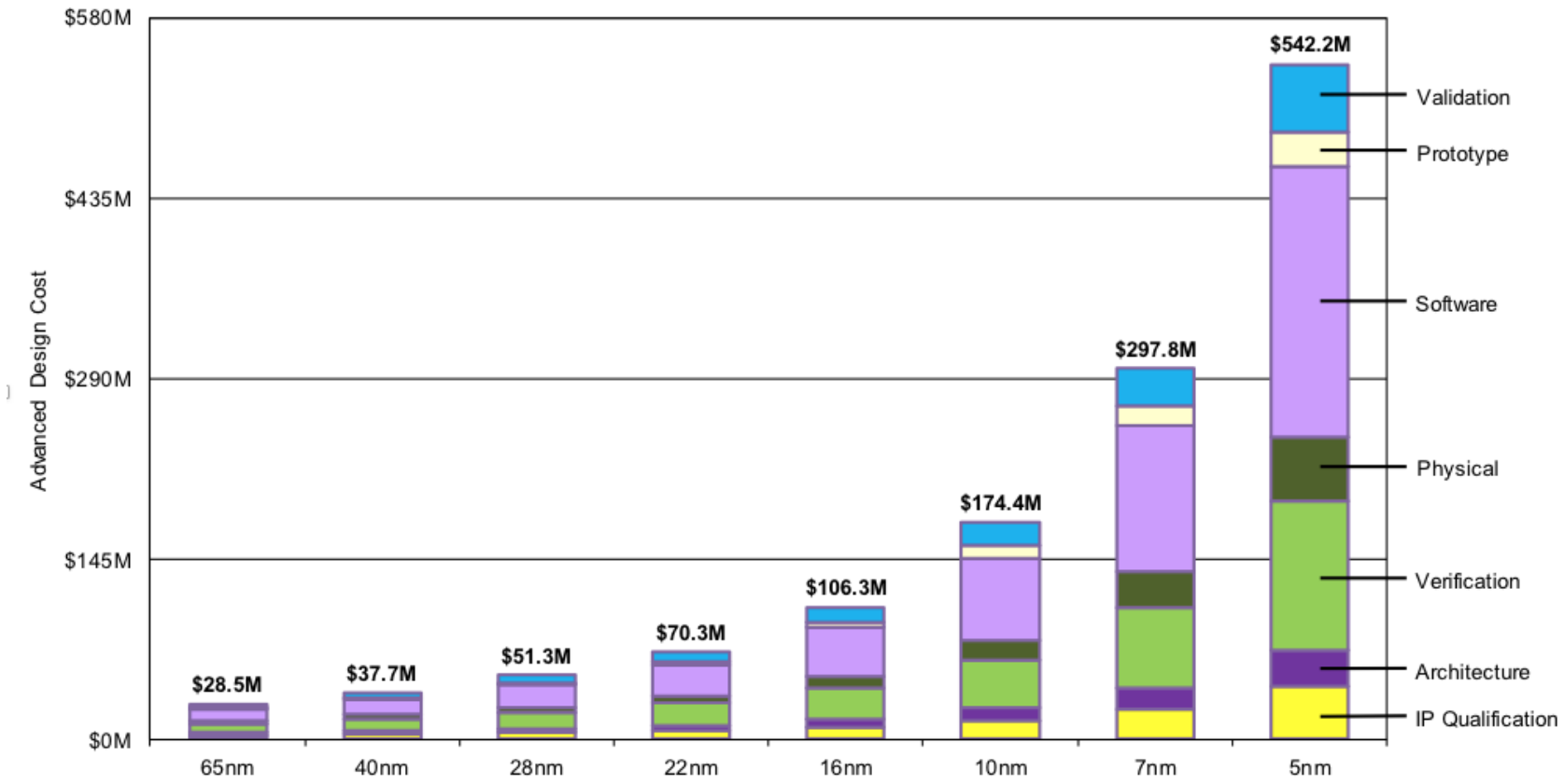


Tradeoffs in Semiconductor Design

Flexibility & Customization

Cost & Time of Chip Design

“Traditional Semiconductor Design” Cost Inhibits Innovation



Favors large players or billion-unit markets

A Broken Semiconductor Economic Model for IoT

- If IoT is a 100B unit / $10T market, that is roughly $100 per unit
- Main processor is approximately 2-3% of that $100 per unit, ~$2-3
- If it costs $50M to design and build that chip, companies must sell at least 25M-50M just to break-even
- How many 50M unit markets are there in IoT?

Changing the Calculus

What If We Drove Down Chip Design Costs by 10X?

- Reducing development cost to <$5M would reduce required shipments to 2.5M units for payback on investment
- Enables more innovation and customization

Focus on mass customization & narrow optimization?

Bring the agility of software to semiconductor design?

Leverage the power of open source?

In “post-Moore” era, Hardware Needs to be Much More Like Software

MVP 1:

MVP 1:

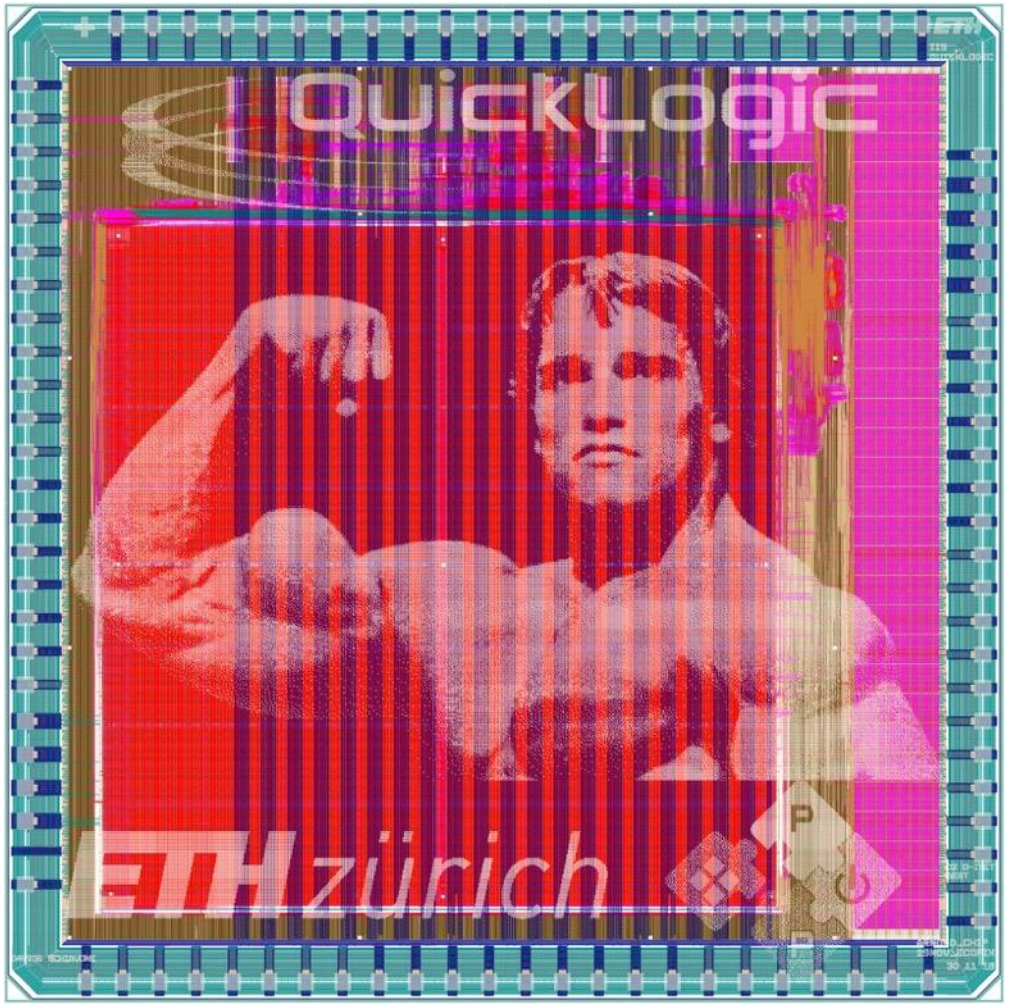
Improve Dev and Cost by **10X**

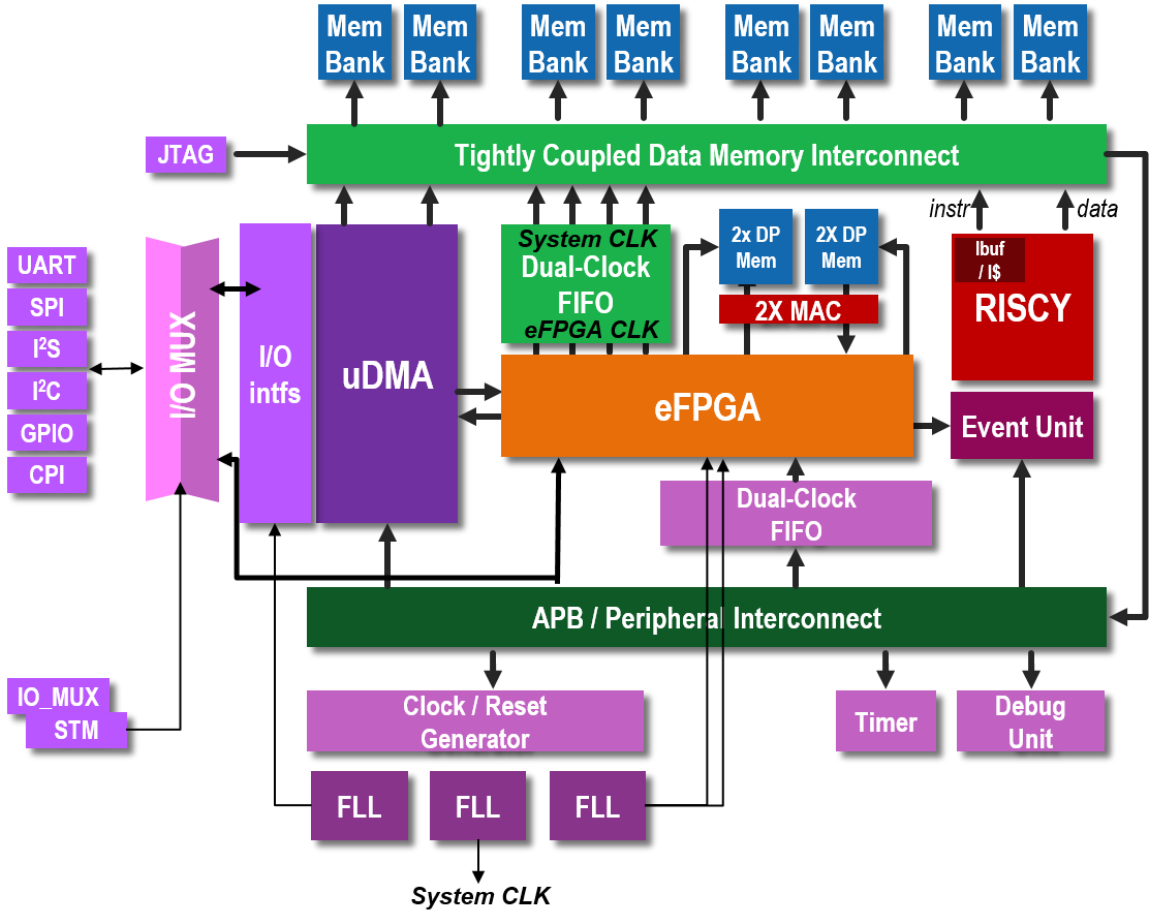
Constraint #1:
Use Open-Source where Possible

Constraint #2:

Automate, Automate, Automate

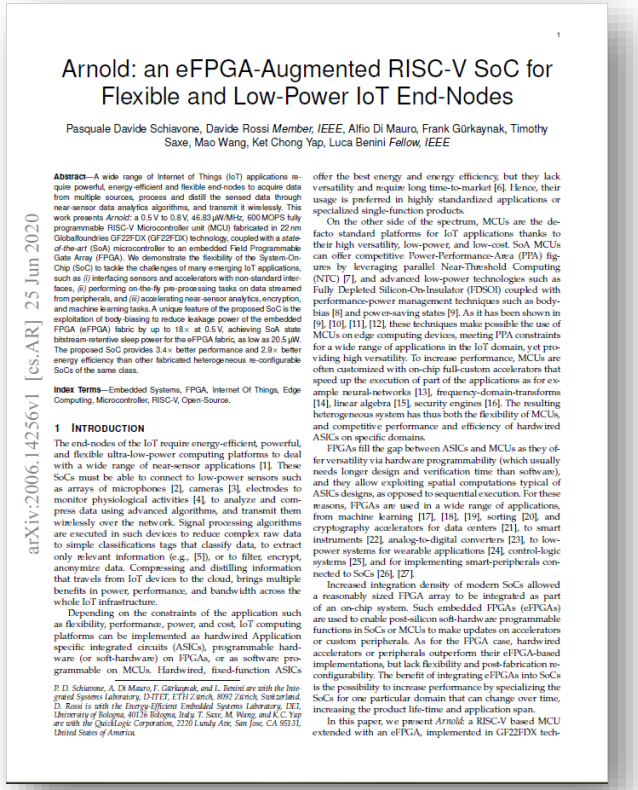
Starting Point: ETH Zurich “Arnold” Test Chip Platform





Arnold – Heterogenous, Energy-Efficient Architecture

- Features
  - RISC-V General Purpose Processor
  - 512 KB Onboard Memory
  - Broad set of peripheral I/O with memory access via µDMA
  - Tightly coupled eFPGA that supports
    - Direct connection to I/O
    - Shared memory accelerator interface
    - I/O filtering functions
    - Config and control interface to/from system
- Benefits
  - Energy efficient architecture enables flexibility to implement hardware partitioning of software requirements
  - Lower unit cost than vs discrete MCU / discrete FPGA implementations
  - OTA hardware upgrades
  - Lower NRE cost vs ‘spinning an ASIC’ for each derivative



1

Arnold: an eFPGA-Augmented RISC-V SoC for Flexible and Low-Power IoT End-Nodes

Pasquale Davide Schiavone, Davide Rossi *Member, IEEE*, Alfio Di Mauro, Frank Gürkaynak, Timothy Saxe, Mao Wang, Ket Chong Yap, Luca Benini *Fellow, IEEE*

Abstract—A wide range of Internet of Things (IoT) applications require powerful, energy-efficient and flexible end-nodes to acquire data from multiple sources, process and distill the sensed data through near-sensor data analytics algorithms, and transmit it wirelessly. This work presents *Arnold*: a 0.5 V to 0.8 V, 46.83 µW/MHz, 600 MOPS fully programmable RISC-V Microcontroller unit (MCU) fabricated in 22 nm Globalfoundries GF22FDX (GF22FDX) technology, coupled with a *state-of-the-art* (SoA) microcontroller to an embedded Field Programmable Gate Array (FPGA). We demonstrate the flexibility of the System-On-Chip (SoC) to tackle the challenges of many emerging IoT applications, such as *(i)* interfacing sensors and accelerators with non-standard interfaces, *(ii)* performing on-the-fly pre-processing tasks on data streamed from peripherals, and *(iii)* accelerating near-sensor analytics, encryption, and machine learning tasks. A unique feature of the proposed SoC is the exploitation of body-biasing to reduce leakage power of the embedded FPGA (eFPGA) fabric by up to 18× at 0.5 V, achieving SoA state bitstream-retentive sleep power for the eFPGA fabric, as low as 20.5 µW. The proposed SoC provides 3.4× better performance and 2.9× better energy efficiency than other fabricated heterogeneous re-configurable SoCs of the same class.

Index Terms—Embedded Systems, FPGA, Internet Of Things, Edge Computing, Microcontroller, RISC-V, Open-Source.

1 Introduction

The end-nodes of the IoT require energy-efficient, powerful, and flexible ultra-low-power computing platforms to deal with a wide range of near-sensor applications [1]. These SoCs must be able to connect to low-power sensors such as arrays of microphones [2], cameras [3], electrodes to monitor physiological activities [4], to analyze and compress data using advanced algorithms, and transmit them wirelessly over the network. Signal processing algorithms are executed in such devices to reduce complex raw data to simple classifications tags that classify data, to extract only relevant information (e.g., [5]), or to filter, encrypt, anonymize data. Compressing and distilling information that travels from IoT devices to the cloud, brings multiple benefits in power, performance, and bandwidth across the whole IoT infrastructure.

Depending on the constraints of the application such as flexibility, performance, power, and cost, IoT computing platforms can be implemented as hardwired Application specific integrated circuits (ASICs), programmable hardware (or soft-hardware) on FPGAs, or as software programmable on MCUs. Hardwired, fixed-function ASICs offer the best energy and energy efficiency, but they lack versatility and require long time-to-market [6]. Hence, their usage is preferred in highly standardized applications or specialized single-function products.

On the other side of the spectrum, MCUs are the de-facto standard platforms for IoT applications thanks to their high versatility, low-power, and low-cost. SoA MCUs can offer competitive Power-Performance-Area (PPA) figures by leveraging parallel Near-Threshold Computing (NTC) [7], and advanced low-power technologies such as Fully Depleted Silicon-On-Insulator (FDSOI) coupled with performance-power management techniques such as body-bias [8] and power-saving states [9]. As it has been shown in [9], [10], [11], [12], these techniques make possible the use of MCUs on edge computing devices, meeting PPA constraints for a wide range of applications in the IoT domain, yet providing high versatility. To increase performance, MCUs are often customized with on-chip full-custom accelerators that speed up the execution of part of the applications as for example neural-networks [13], frequency-domain-transforms [14], linear algebra [15], security engines [16]. The resulting heterogeneous system has thus both the flexibility of MCUs, and competitive performance and efficiency of hardwired ASICs on specific domains.

FPGAs fill the gap between ASICs and MCUs as they offer versatility via hardware programmability (which usually needs longer design and verification time than software), and they allow exploiting spatial computations typical of ASICs designs, as opposed to sequential execution. For these reasons, FPGAs are used in a wide range of applications, from machine learning [17], [18], [19], sorting [20], and cryptography accelerators for data centers [21], to smart instruments [22], analog-to-digital converters [23], to low-power systems for wearable applications [24], control-logic systems [25], and for implementing smart-peripherals connected to SoCs [26], [27].

Increased integration density of modern SoCs allowed a reasonably sized FPGA array to be integrated as part of an on-chip system. Such embedded FPGAs (eFPGAs) are used to enable post-silicon soft-hardware programmable functions in SoCs or MCUs to make updates on accelerators or custom peripherals. As for the FPGA case, hardwired accelerators or peripherals outperform their eFPGA-based implementations, but lack flexibility and post-fabrication reconfigurability. The benefit of integrating eFPGAs into SoCs is the possibility to increase performance by specializing the SoCs for one particular domain that can change over time, increasing the product life-time and application span.

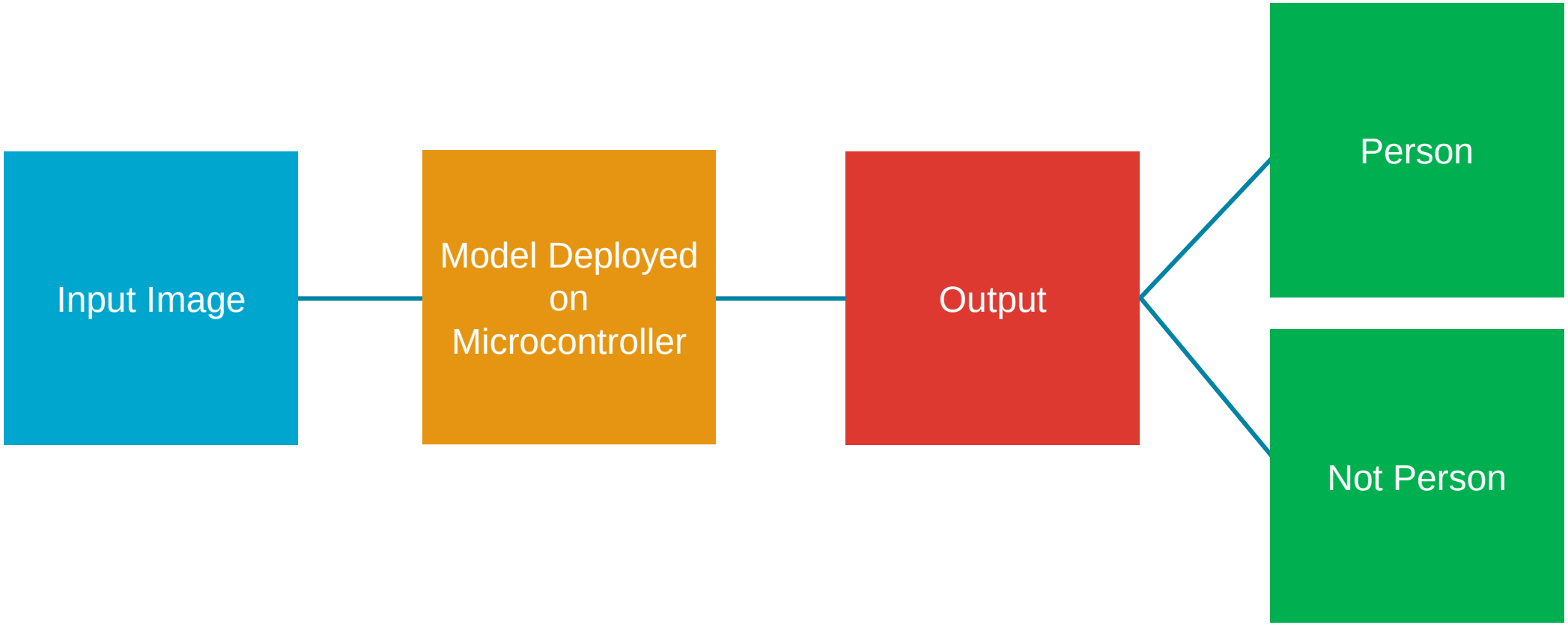
In this paper, we present *Arnold*: a RISC-V based MCU extended with an eFPGA, implemented in GF22FDX tech-

P. D. Schiavone, A. Di Mauro, F. Gürkaynak, and L. Benini are with the Integrated Systems Laboratory, D-ITET, ETH Zürich, 8092 Zürich, Switzerland. D. Rossi is with the Energy-Efficient Embedded Systems Laboratory, DEI, University of Bologna, 40126 Bologna, Italy. T. Saxe, M. Wang, and K.C. Yap are with the QuickLogic Corporation, 2220 Lundy Ave, San Jose, CA 95131, United States of America.

https://arxiv.org/pdf/2006.14256.pdf

Case Study – Human Presence Detection

- Presence detection (aka Visual Wake Words) application using TensorFlow Lite for Microcontrollers (TFLu): (https://arxiv.org/abs/1906.05721)



Arnold2 and TFLu Person Detection – eFPGA Use Case

- Basic facts:
  - TFLu model: 230KiB
  - TFLu arena: 95KiB
  - ROM seg: 308KiB
  - RAM seg: 116KiB
    Total: 424KiB
- Basic facts: inference
  - 7.072M MAC per inference
  - 134M clock cycles per inference
  - 3.66fps at 492MHz (11.2mW)

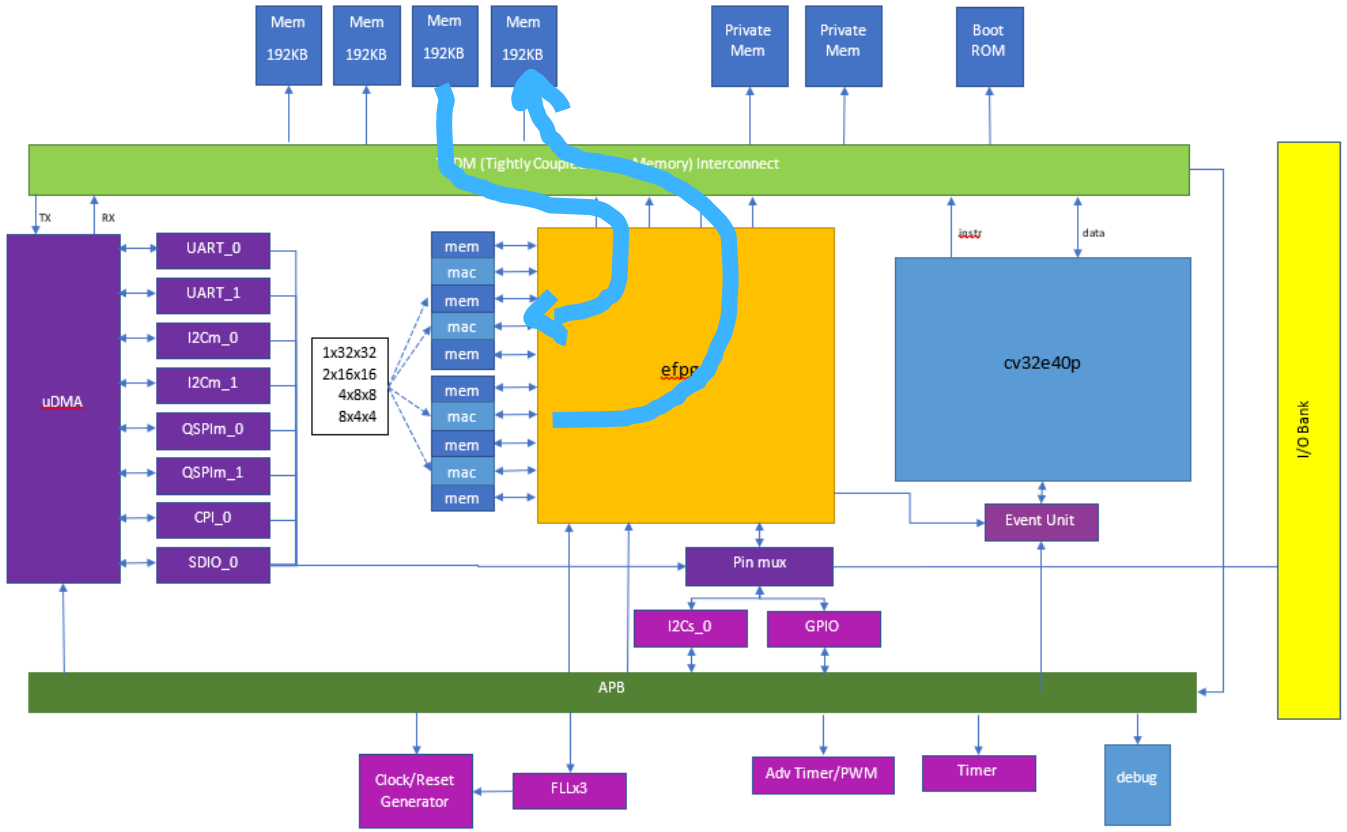
  ➔ 19 clock cycles per MAC

- Two main users of MACs
  - Conv_2d ➔ 6,193,902 MAC
  - Depthwise_conv_2d ➔ 878,616 MAC

Use eFPGA for offloaded conv_2d

- Three main benefits
  1. State machine handles loops and addressing
     → Saves 19 clocks per MAC
  2. Multiple MACs in parallel
     →8 in Arnold with possible 2X increase
     →Coefficients cached in local memory
     →Saves memory bandwidth and power

Arnold2



- Coefficients moved into local memory
- Activations streamed from main memory, thru MACs and back to main memory

- 13x speedup
- 31x better energy efficiency

Measured Value

Node	MACs	ConvSW					ConvHW		#clocks	MACs/CLKS	speedup
		mSec	mSec	mSec	mSec		mSec	mSec			
2	294,912	20.08	20.04	20.32	20.34		1.42	1.41	69,251	4.3	14.3
4	294,912	17.55	17.53	17.65	17.68		1.10	1.10	53,507	5.5	16.0
6	589,824	32.65	32.57	32.73	32.72		2.05	2.05	100,099	5.9	15.9
8	294,912	16.27	16.28	16.37	16.36		1.06	1.06	51,587	5.7	15.4
10	589,824	31.31	31.29	31.44	31.44		2.04	2.04	99,715	5.9	15.4
12	294,912	15.67	15.68	15.73	15.73		1.14	1.14	56,003	5.3	13.7
14	589,824	30.71	30.73	30.77	30.81		2.25	2.24	110,290	5.3	13.7
16	589,824	30.74	30.71	30.79	30.77		2.24	2.24	110,290	5.3	13.7
18	589,824	30.72	30.72	30.78	30.78		2.25	2.24	110,290	5.3	13.7
20	589,824	30.75	30.69	30.77	30.78		2.24	2.24	110,290	5.3	13.7
22	589,824	30.69	30.75	30.78	30.79		2.24	2.24	110,290	5.3	13.7
24	294,912	15.33	15.37	15.39	15.39		1.61	1.62	79,760	3.7	9.5
26	589,824	30.39	30.41	30.49	30.40		3.22	3.22	158,668	3.7	9.5
Total	6,193,152	332.86	332.77	334.01	333.99		24.85	24.84	1,220,040	5.08	
										63%	
		Software Conv2d @ 456 MHz for 13 convolutions = ~333 mS									
		Hardware Accel @ 50 MHz = 24.84mS				13.4	times improvement				

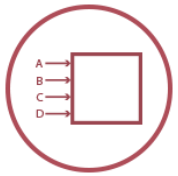
1/ expect to increase to 75% efficiency
2/ measured 75MHz➔ 20x speedup; most recent measured timing 88MHz

- S/W at 456 MHz
- FPGA at 50MHz
- Avg 5 MAC/clock
  → 63% efficiency[1]
- 13.4x speedup[2]
- S/W 10.4mW
  → 3,467μJ
- FPGA 4.5mW
  → 112μJ
- 31x energy efficiency

Creating MVP

Definition

Arch + Std Cells

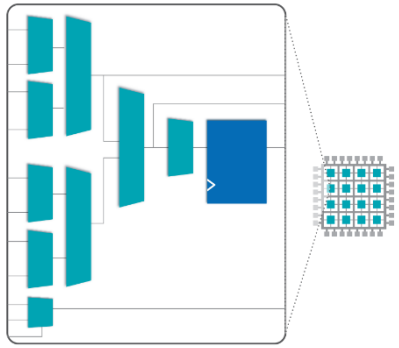


Arch

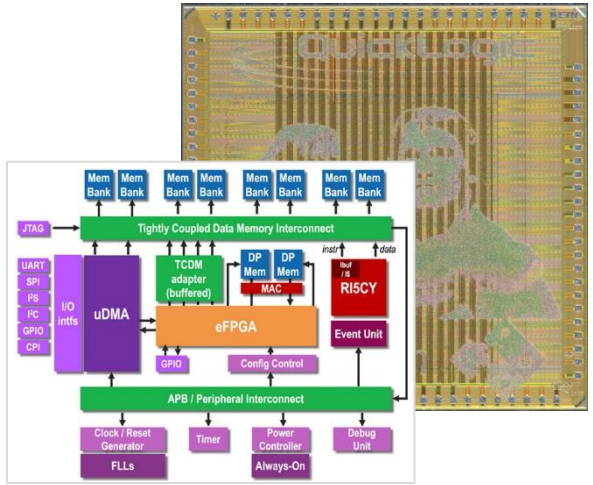
Test

Program

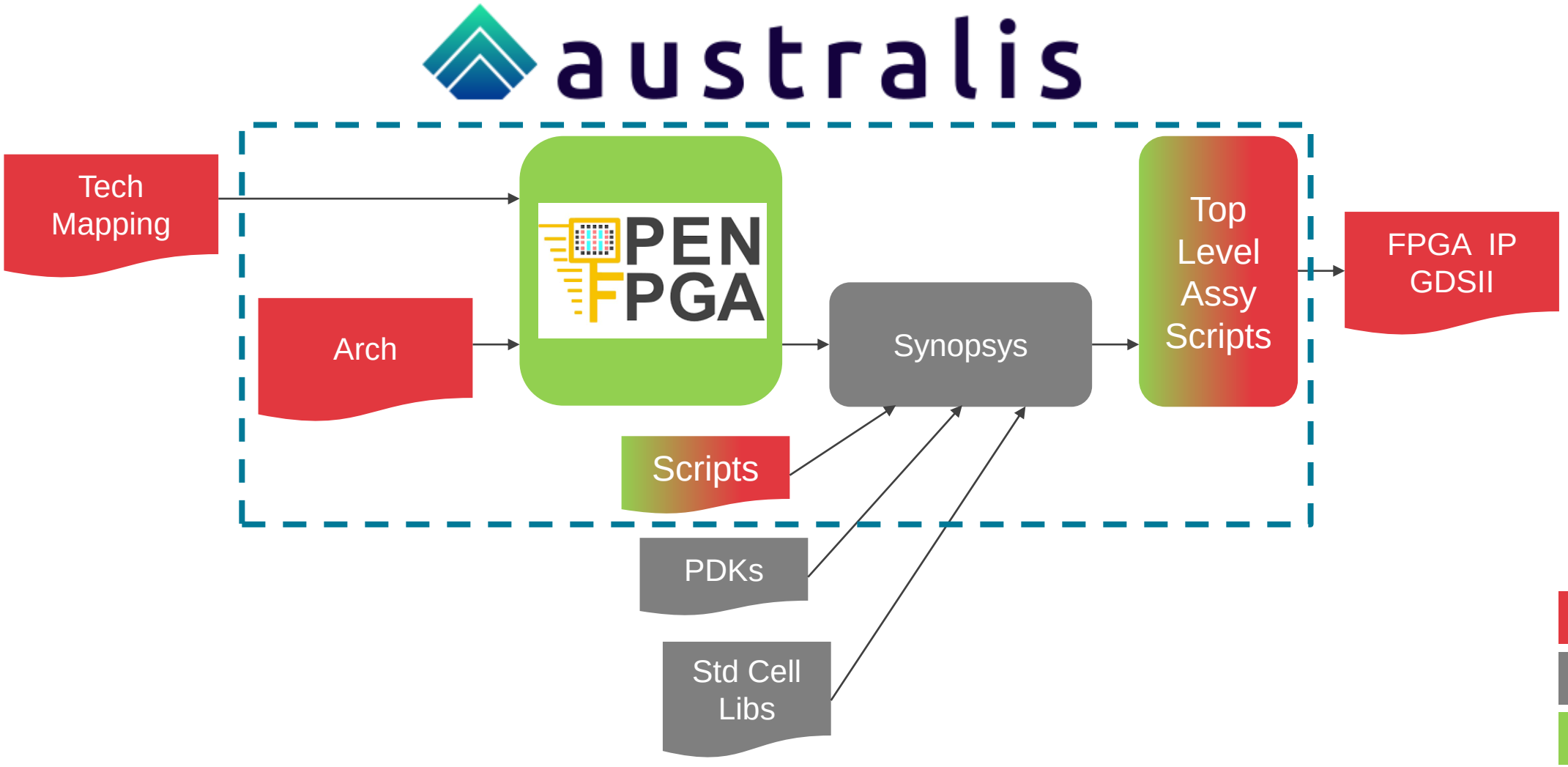
Implementation



Validation



QuickLogic eFPGA IP Development Workflow



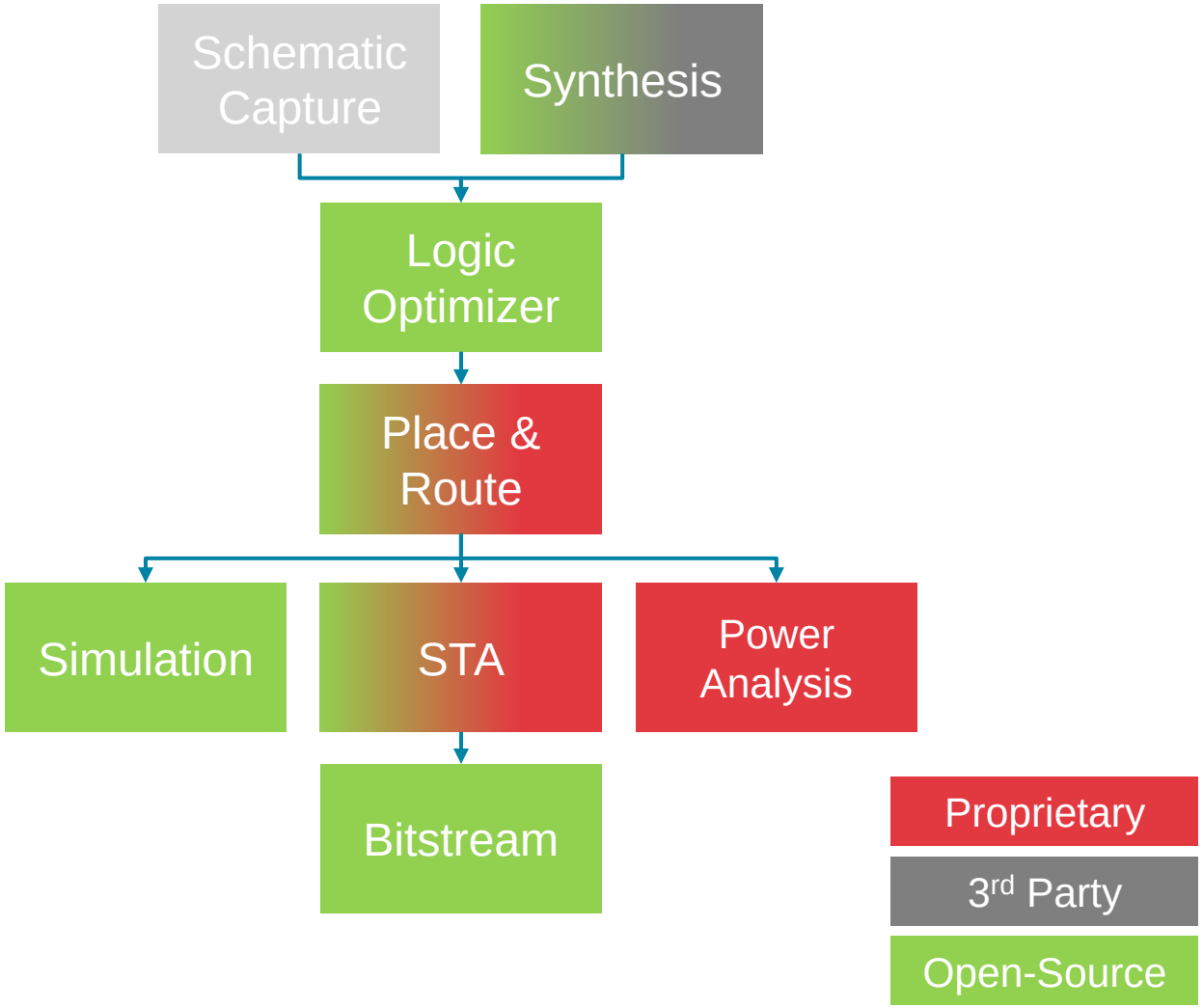
Proprietary

3rd Party

Open-Source

QuickLogic Aurora FPGA User Tools

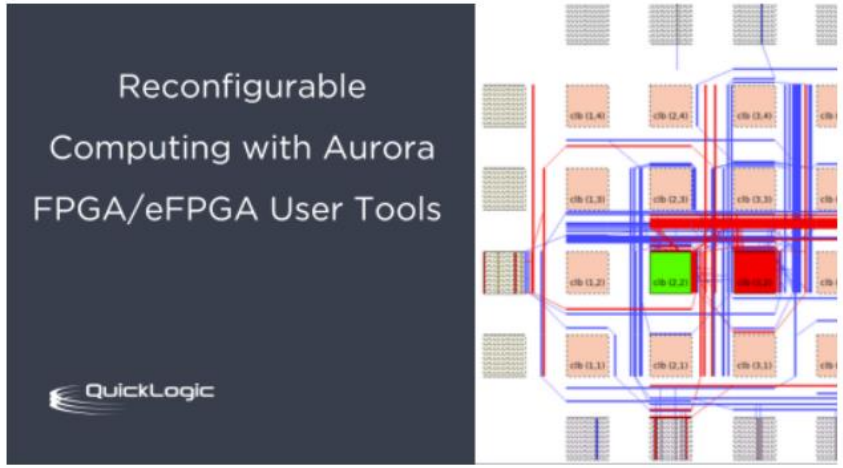
- Leveraging Open-Source Tools where possible
- Contribute to improvements in Open-Source Tools for greater community
- Enable compatibility with 3rd Party tools for broader use
- Adapt for new, domain-specific architectures

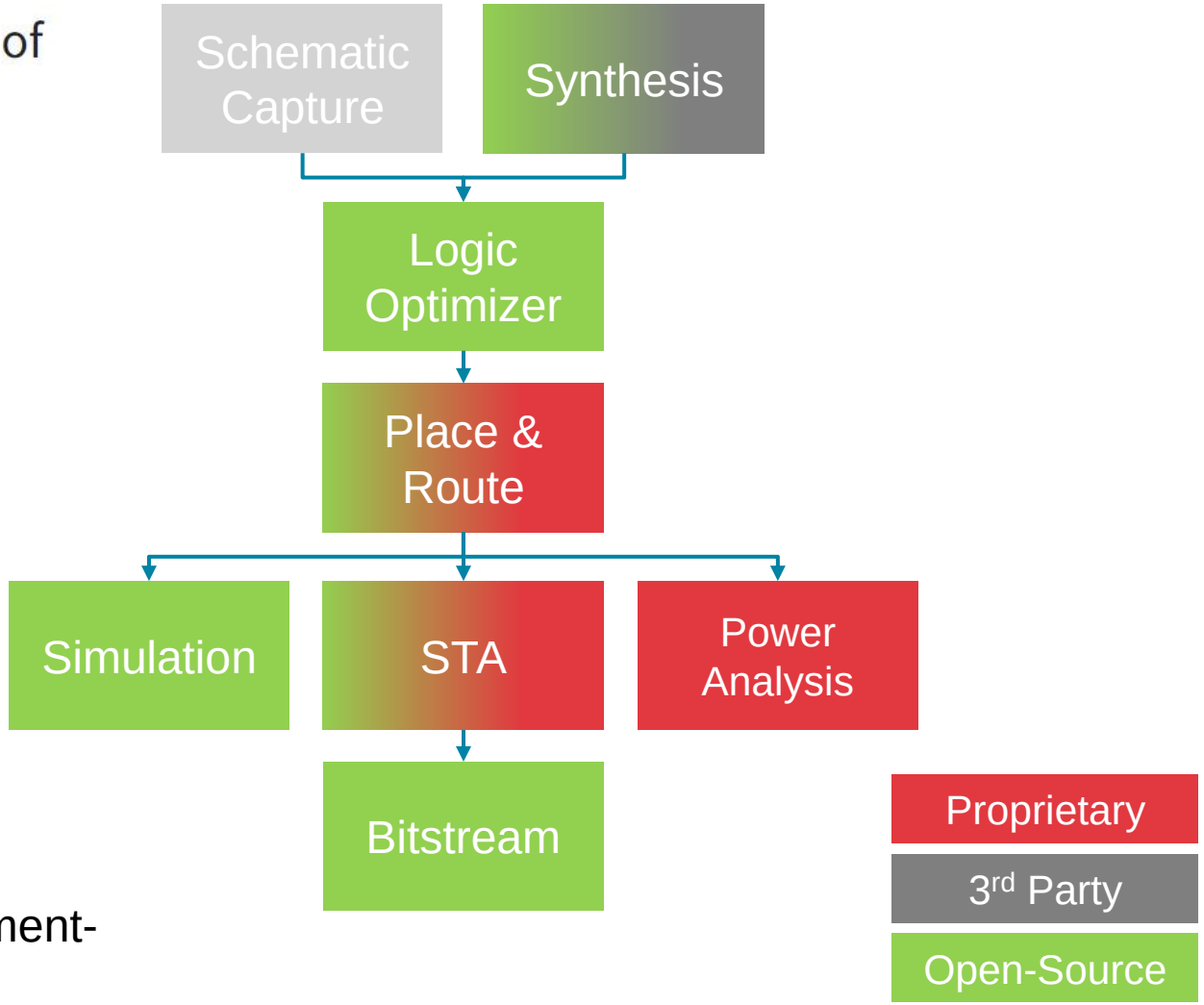


Thank you to U of Toronto, Amin Mohaghegh and Vaughn Betz

Aurora 2.4 Development Tool: Celebrating the Power of Open-Source Innovation

Posted on November 16, 2023 by Andrea Vedanayagam





https://www.quicklogic.com/2023/11/16/aurora-2-4-development-tool-celebrating-the-power-of-open-source-innovation/

Commitment to Fostering The Next Generation

UNIVERSITY OF TORONTO

SANTA CLARA UNIVERSITY 1851

UNIVERSITY of ROCHESTER

ROCHESTER INSTITUTE OF TECHNOLOGY R·I·T 1829

THE UNIVERSITY OF TEXAS AT AUSTIN

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

THE UNIVERSITY OF UTAH

What’s Next?

Domain-specific Variants

Customized eFPGA/FPGA for Operating Environments

Operating Environments

- High Reliability
- Ruggedized / Extreme Environments
- Often requires using custom ASIC cell librarires

Customized eFPGA/FPGA for Specific Workloads

Customizing for Specific Workloads

- New eFPGA/FPGA architecture features
- Optimizing the routing tracks for the workloads
- Optimizing the features ratios for the workloads
- May require different configuration memories
- May require different configuration architectures / tools

Conclusions

100% Open-Source Flow is Great…

But Commercialization Requires Additional Contributions…

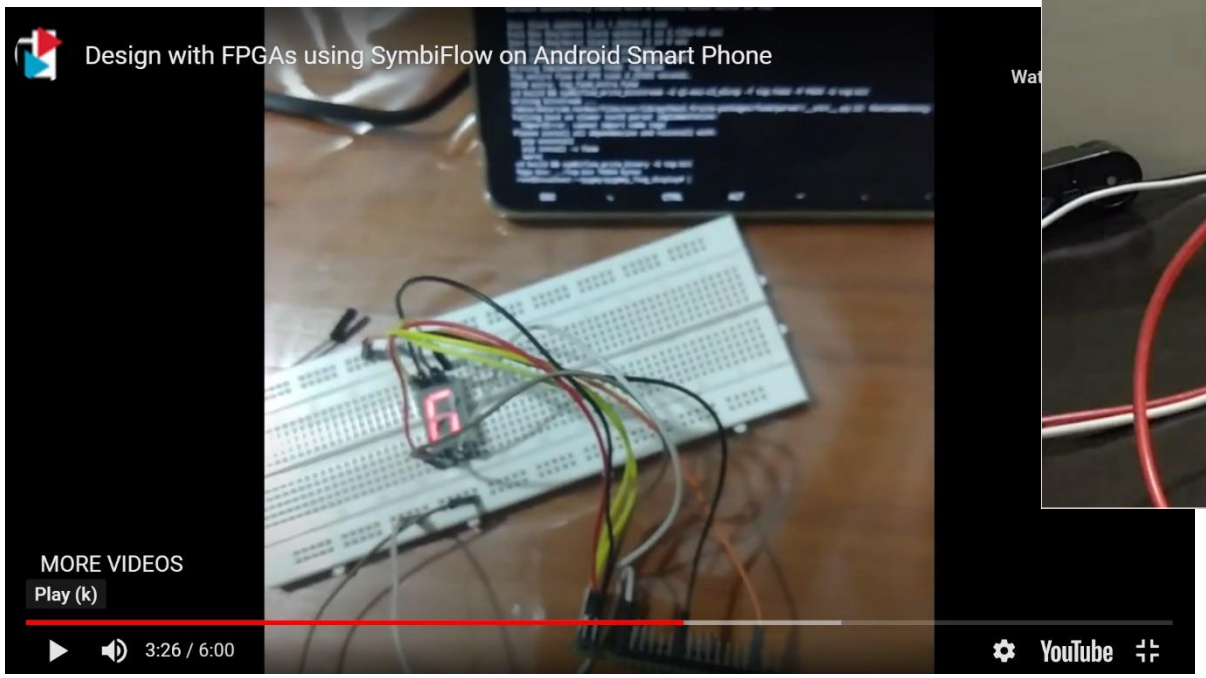
Recommendations

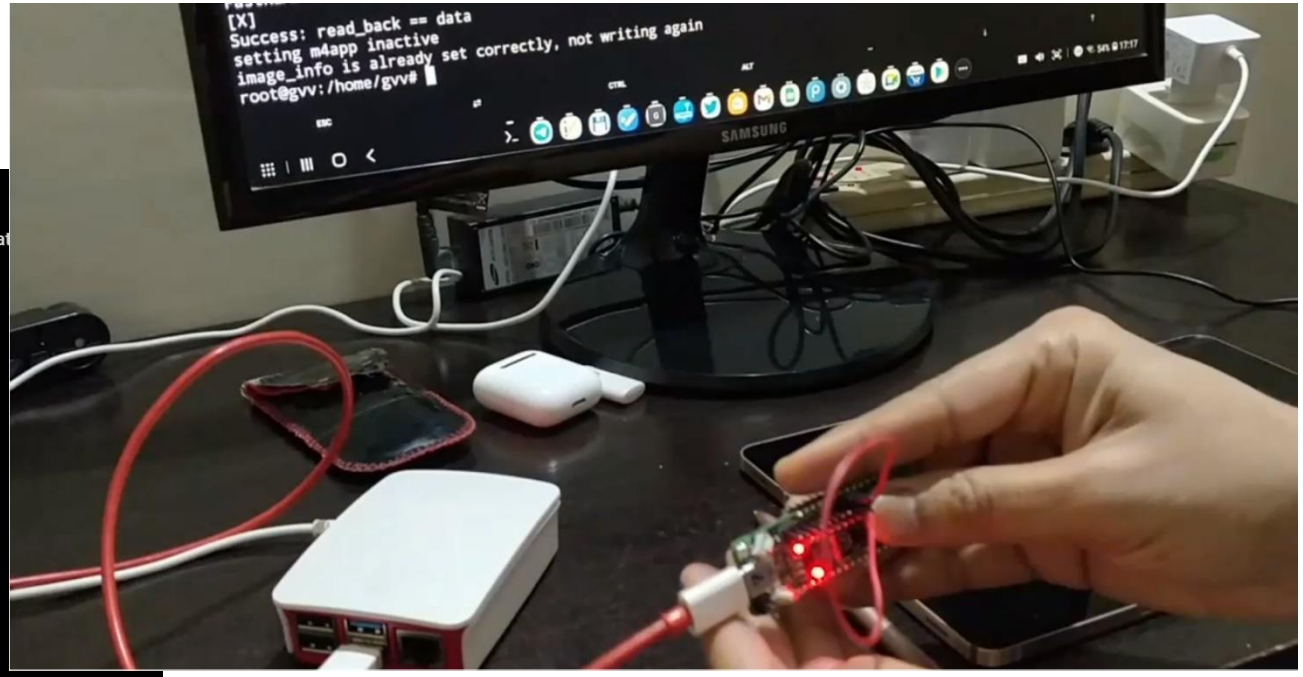
How do we overcome NIH?

Continue to improve QoR of User Tools?

A Glimpse Into the Future – Designing with your Smartphone

- https://www.youtube.com/watch?v=bRrJqL3NGlg





slido

How do you feel about open-source FPGA tools now?

ⓘ Start presenting to display the poll results on this slide.

Want to build FPGAs / eFPGA IPs?

Email me:
faith@quicklogic.com