
XILINX
ALL PROGRAMMABLE™

# FPGA加速机器学习应用

罗霖
*Andy.luo@Xilinx.com*

2017年6月20日

# Xilinx – The All Programmable Company

XILINX
ALL PROGRAMMABLE™

XILINX - Founded 1984

- Headquarters
- Research and Development
- Sales and Support
- Manufacturing

**$2.21B** FY16 revenue

**>57%** market segment share

**3,500+** employees worldwide

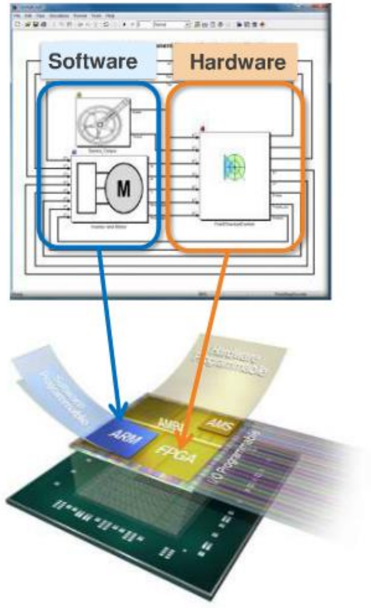
**20,000** customers worldwide

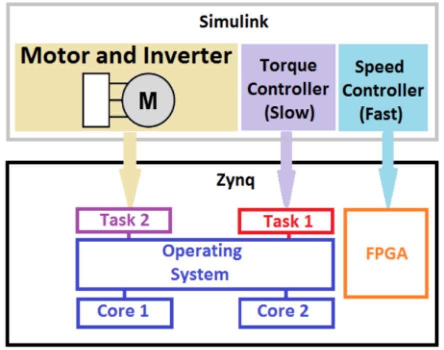
**3,500+** patents

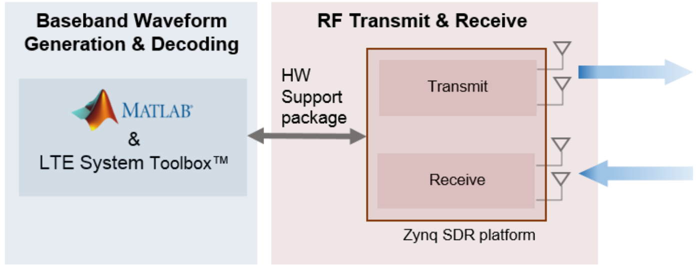
**60** industry firsts

# Great Partnership between Mathworks & Xilinx

XILINX SYSTEM GENERATOR™ For DSP




## Xilinx Zynq Support from Computer Vision System Toolbox

*Design and prototype vision systems using Xilinx Zynq-based hardware*

# What is FPGA

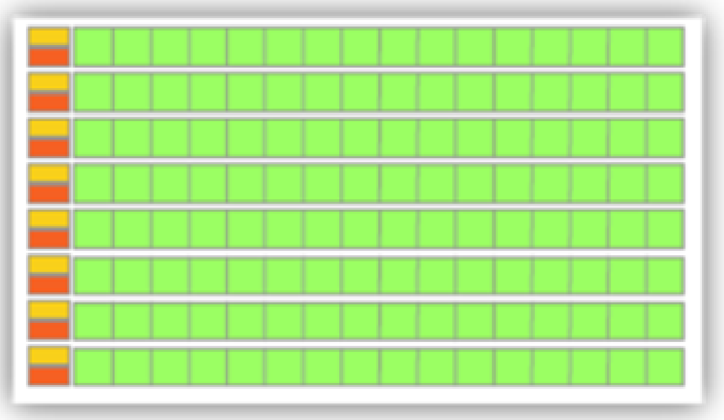
- A field-programmable gate array (FPGA) is an integrated circuit that can be programmed in the field after manufacture.

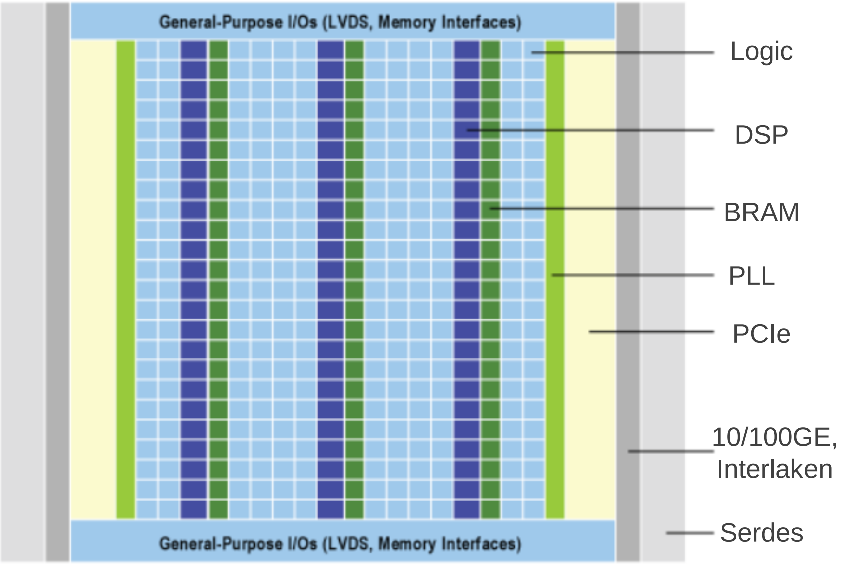

- FPGA contain an array of programmable logic blocks and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together“.

- Usually programmed with HDL (VHDL/Verilog) and now supports C/C++/OpenCL and model-based tool (Matlab, Labview…)

- A very wide range of applications including wired&wireless communication, date center, aerospace&defense, industrial, medical, automotive, test&measurement, audio&video, even consumer…

# CPU vs GPU vs FPGA



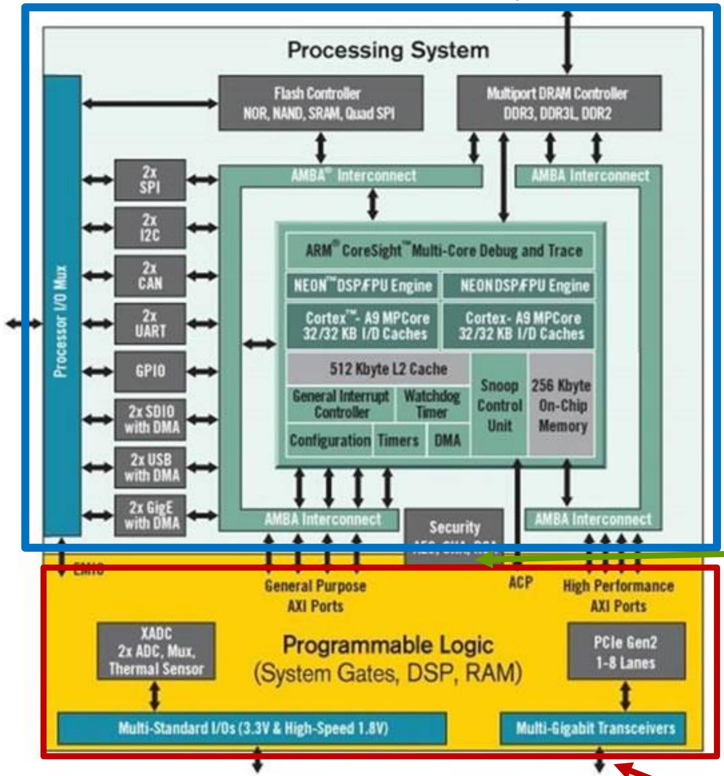

- Complex control logic
- Large caches
- Optimized for serial operations

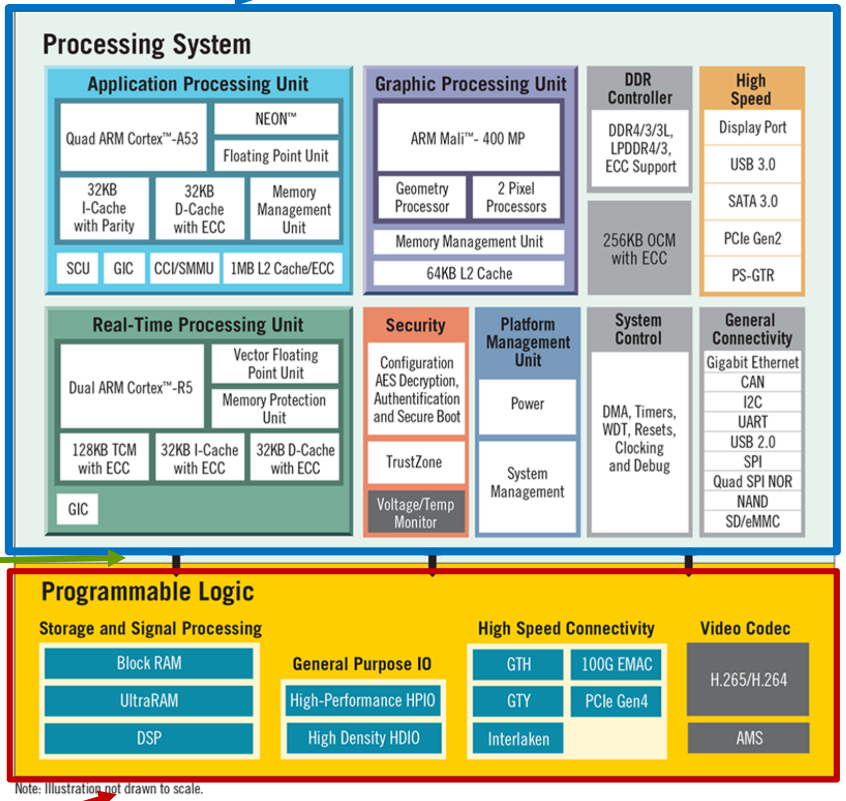
- Limited control function
- High throughput
- Built for parallel operations

- Many programmable I/O
- Large internal memory
- Customized for complex control & parallel computation

# SOC with FPGA

ARM Subsystem



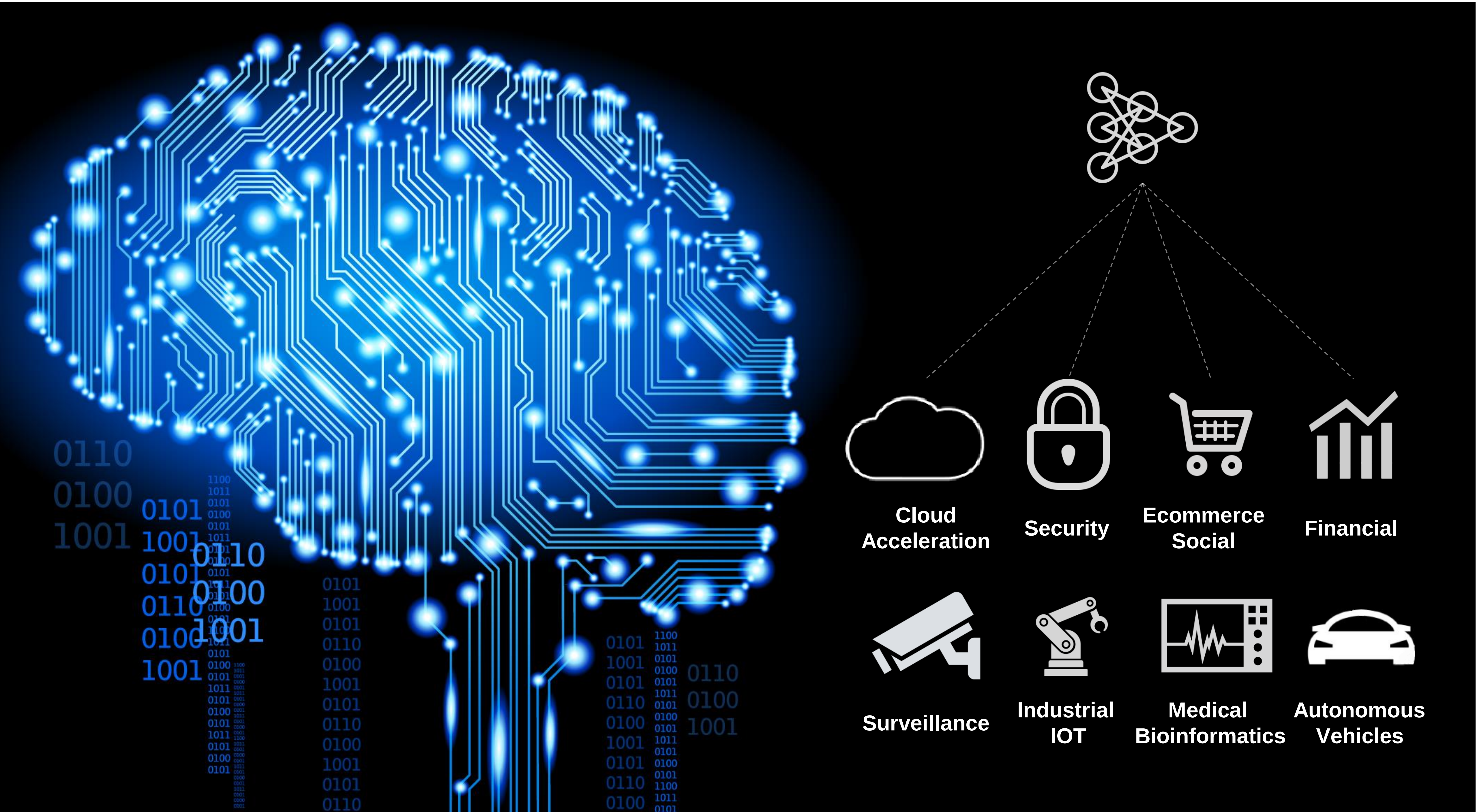




High-speed on-chip bus

28nm SoC

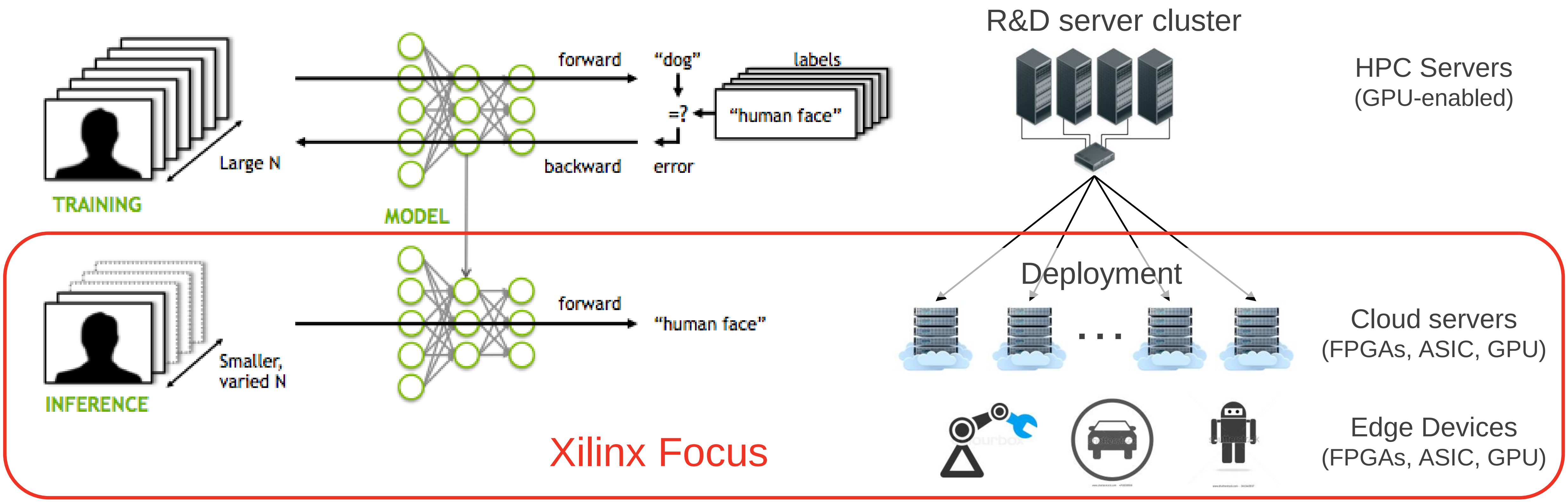

FPGA Subsystem

16nm SoC


Cloud Acceleration
Security
Ecommerce Social
Financial
Surveillance
Industrial IOT
Medical Bioinformatics
Autonomous Vehicles

**Training**: Process for machine to “learn” and optimize a model from data

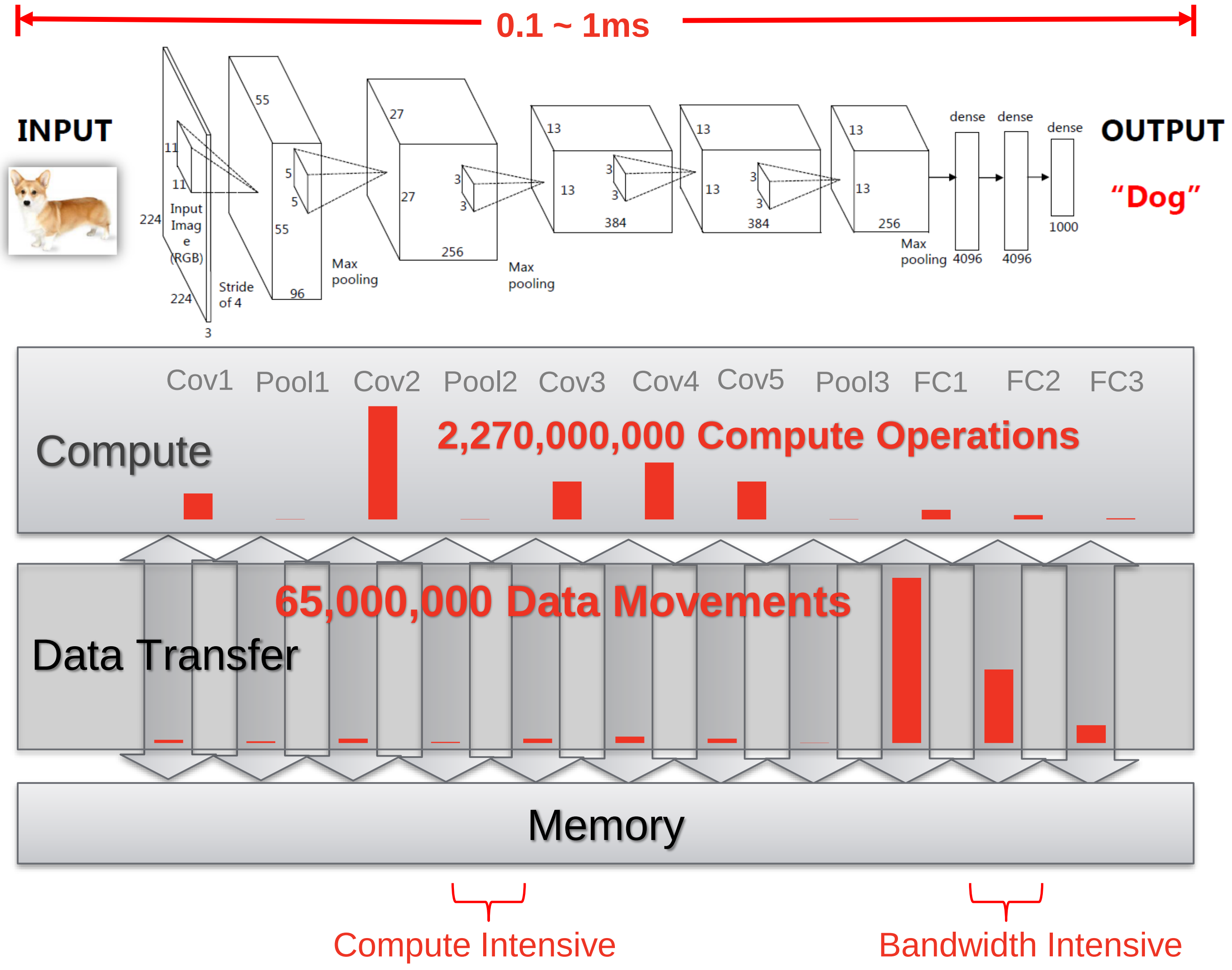
**Inference**: Using trained model to predict/estimate outcomes from new observations



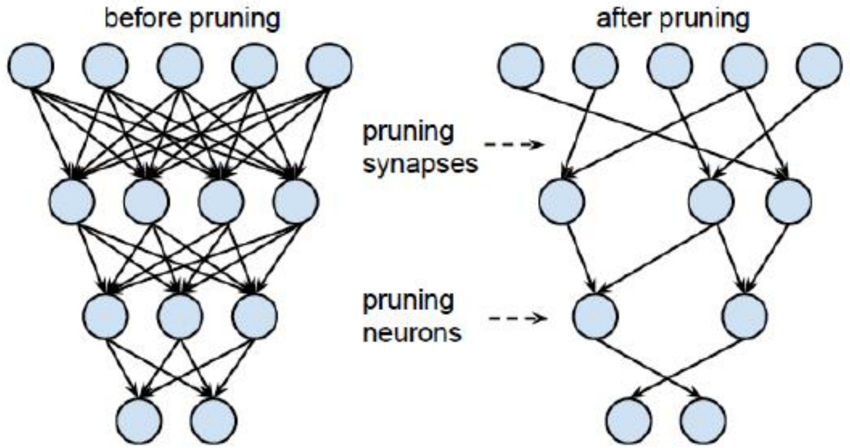
# Deep Learning Technical Challenges

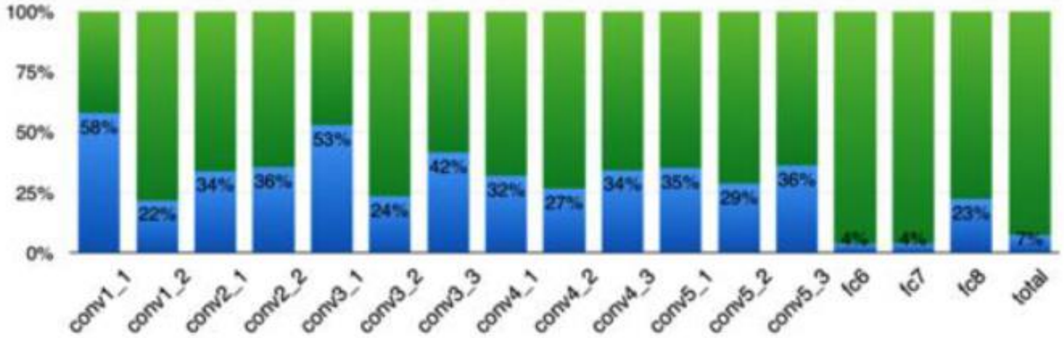
- Computational Intensive
- Memory Bandwidth Intensive
- Deployment Power Efficiency

Example – Deep Learning Inference: Image Classification (AlexNet)



# Deep Compression




30x – 50x compression rate without hurting accuracy

- **Small DNN models are critical.**


| Network | Original Size | Compressed Size | Compression Ratio | Original Accuracy | Compressed Accuracy |
|---|---|---|---|---|---|
| AlexNet | 240MB ⟶ | 6.9MB | **35x** | 80.27% ⟶ | 80.30% |
| VGGNet | 550MB ⟶ | 11.3MB | **49x** | 88.68% ⟶ | 89.09% |
| GoogleNet | 28MB ⟶ | 2.8MB | **10x** | 88.90% ⟶ | 88.92% |
| SqueezeNet | 4.8MB ⟶ | 0.47MB | **10x** | 80.32% ⟶ | 80.35% |

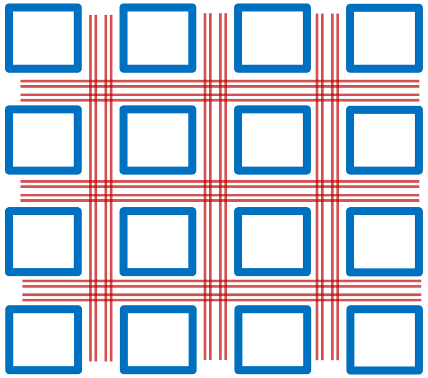
# Machine Learning Moving towards Lower Precession

## Activation Quantization: 8 Bits Are Enough

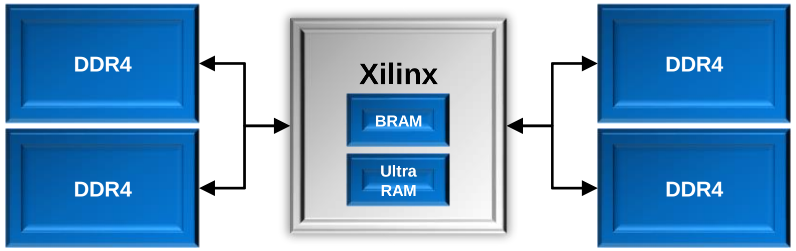
- Inference with Integer Quantization
  - Fixed-Point sufficient for Deployment (INT16, INT8)
  - No Significant Loss in Accuracy (< 1%)
  - >10x Energy Efficiency OPs/J (INT8 vs FP32)
  - 4x Memory Energy Efficiency Tx/J (INT8 vs FP32)

| | | FP32 | FIXED-16 | FIXED-8 |
|---|---|---|---|---|
| VGG16 | Top-1 | 65.77% | 65.78% | 65.58% |
| | Top-5 | 86.64% | 86.65% | 86.38% |
| GoogLeNet | Top-1 | 68.60% | 68.70% | 62.75% |
| | Top-5 | 88.65% | 88.45% | 85.70% |
| SqueezeNet | Top-1 | 58.69% | 58.69% | 57.27% |
| | Top-5 | 81.37% | 81.35% | 80.32% |

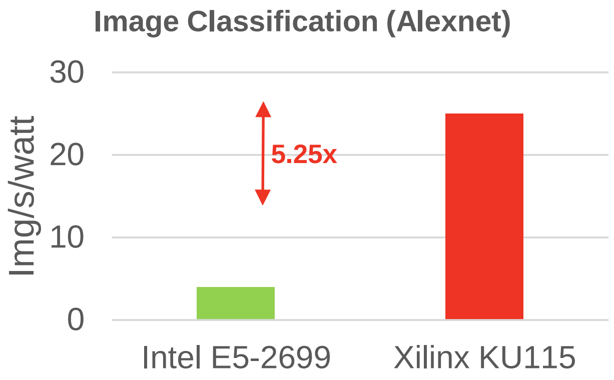
# FPGA Advantages in Deep Learning



Customizable Massive Parallel Compute Power




Fine-grained Memory Hierarchy Reduce Memory Bottlenecks

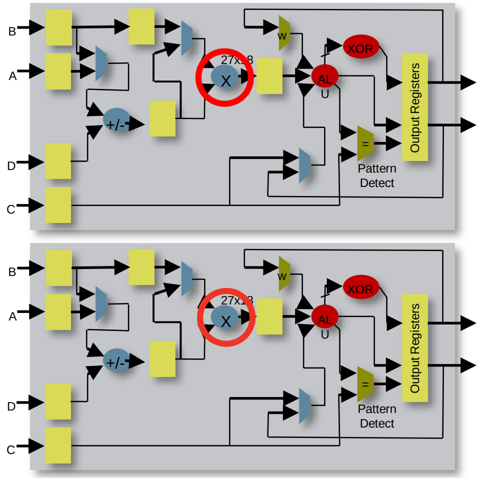



Power Efficient

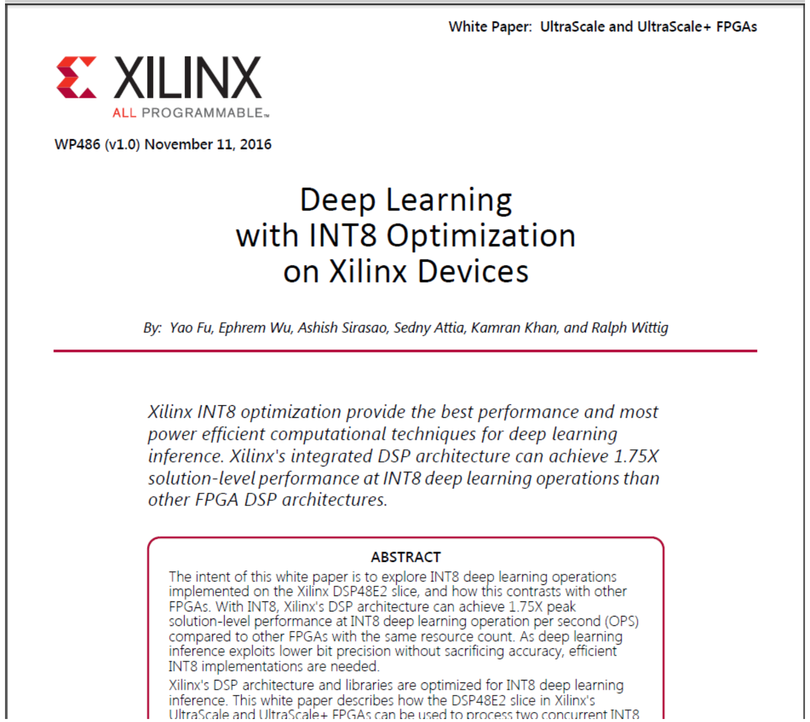
# Xilinx is more Efficient at Int8 Inference

## Scalable MACC with reduced precision

- Xilinx supports up to 27x18 bits in a single multiplier vs. 18x18 in Arria/Stratix 10 DSP Block
- Enough bit-width to perform two separate MACCs with one shared factors for 8-bit computes on single DSP




Xilinx DSP48E2



White Paper: UltraScale and UltraScale+ FPGAs

XILINX
ALL PROGRAMMABLE.

WP486 (v1.0) November 11, 2016

Deep Learning
with INT8 Optimization
on Xilinx Devices

By: Yao Fu, Ephrem Wu, Ashish Sirasao, Sedny Attia, Kamran Khan, and Ralph Wittig

*Xilinx INT8 optimization provide the best performance and most power efficient computational techniques for deep learning inference. Xilinx's integrated DSP architecture can achieve 1.75X solution-level performance at INT8 deep learning operations than other FPGA DSP architectures.*

ABSTRACT

The intent of this white paper is to explore INT8 deep learning operations implemented on the Xilinx DSP48E2 slice, and how this contrasts with other FPGAs. With INT8, Xilinx's DSP architecture can achieve 1.75X peak solution-level performance at INT8 deep learning operation per second (OPS) compared to other FPGAs with the same resource count. As deep learning inference exploits lower bit precision without sacrificing accuracy, efficient INT8 implementations are needed.

Xilinx's DSP architecture and libraries are optimized for INT8 deep learning inference. This white paper describes how the DSP48E2 slice in Xilinx's UltraScale and UltraScale+ FPGAs can be used to process two concurrent INT8

# ML Performance Comparison with Nvidia Tegra Devices

**Nvidia Tegra K1/X1 SoC**

- 28 nm / 20nm
- 192 / 256 CUDA Cores
- Caffe with latest CuDNN

**Xilinx Zynq 7020/ZU2CG (Projection)**

- 28nm / 16nm
- 85k/103k logic cells
- 220/240 DSP
- 4.9/5.3Mb BRAM

## Benchmark


VGG16
Image classification

30.68 Gop 13 Conv layers

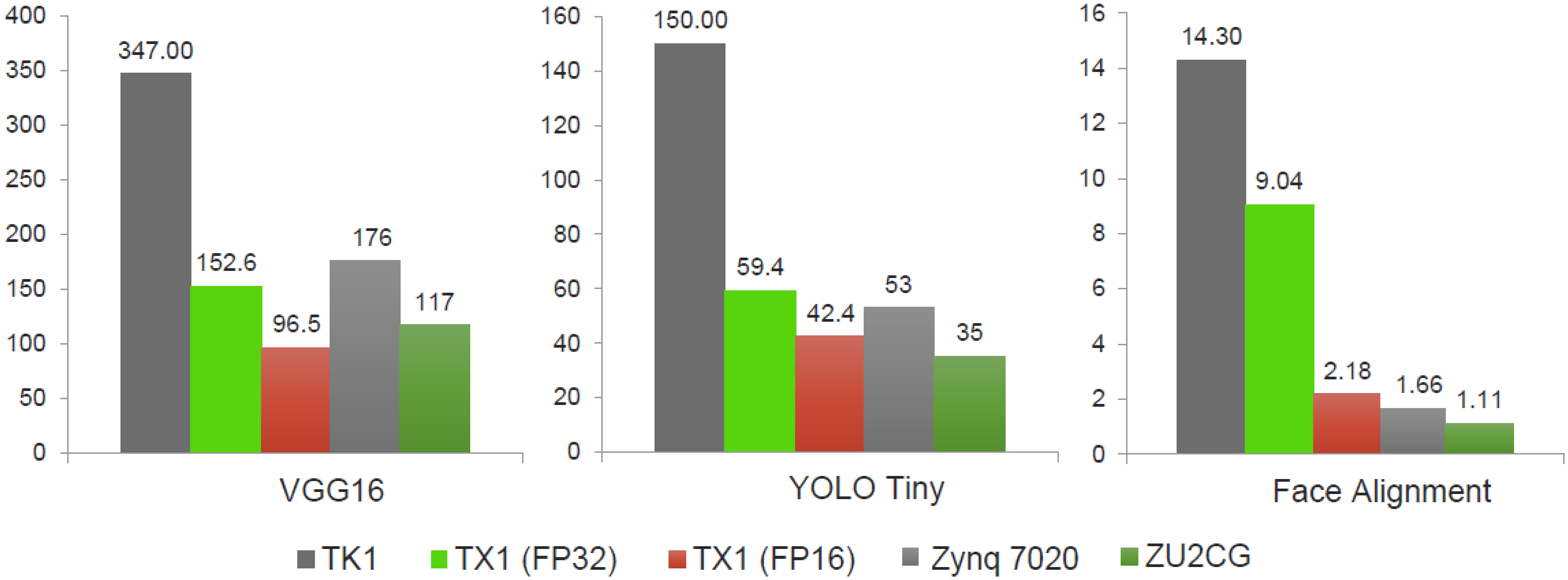
YOLO Tiny
General object detection

5.54 Gop, 9 Conv layers

Customized Network
Face alignment

104.6 Mop, 9 Conv layers

# ML Performance Comparison with Nvidia Tegra Devices (Cont.)




Source: Deephi
Zynq7020 PL @ 200MHz, ZU2CG PL @ 300MHz

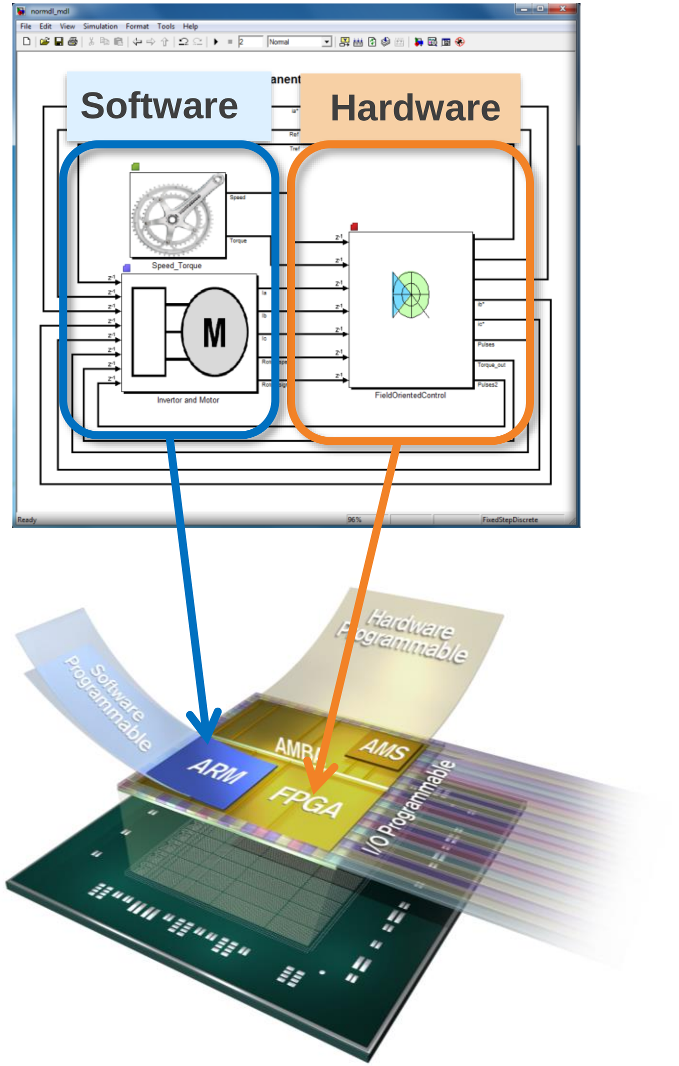
# Different User Personas

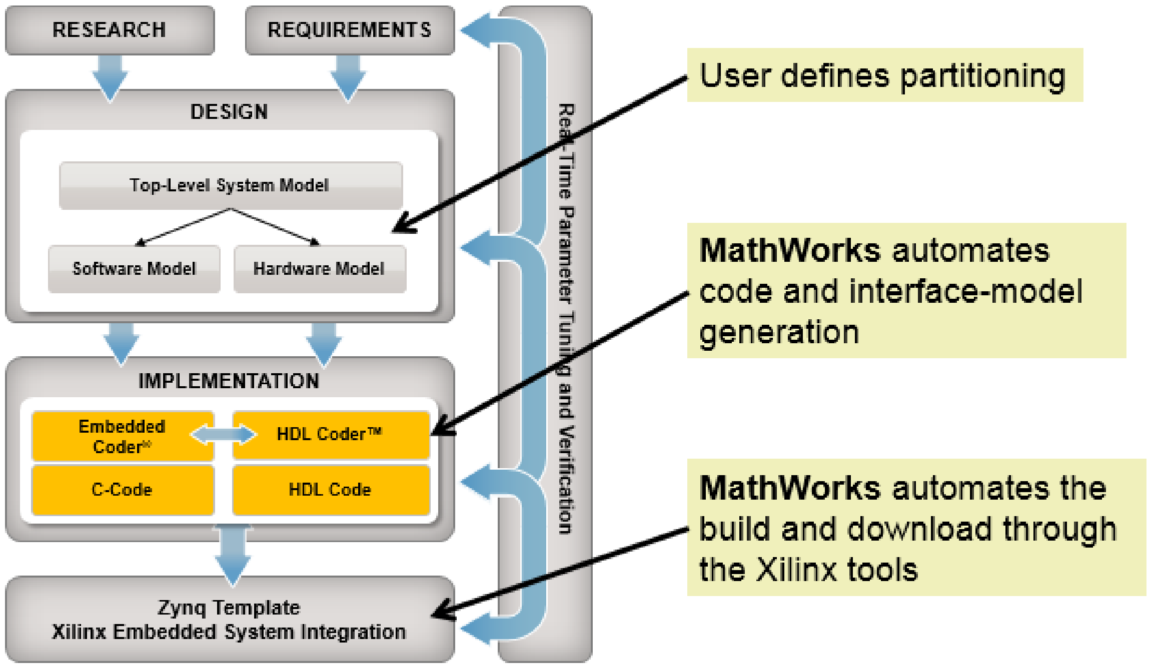
Hardware Engineer

Algorithm / DSP Engineer

Software Engineer

# MathWorks Guided Workflow for Zynq




- **From requirements, to model, to rapid prototype**
- **A guided workflow for hardware and software development**
  - **HDL Coder: Programmable Logic bitstream generation**
  - **Embedded Coder: Software build file generation / Drivers**

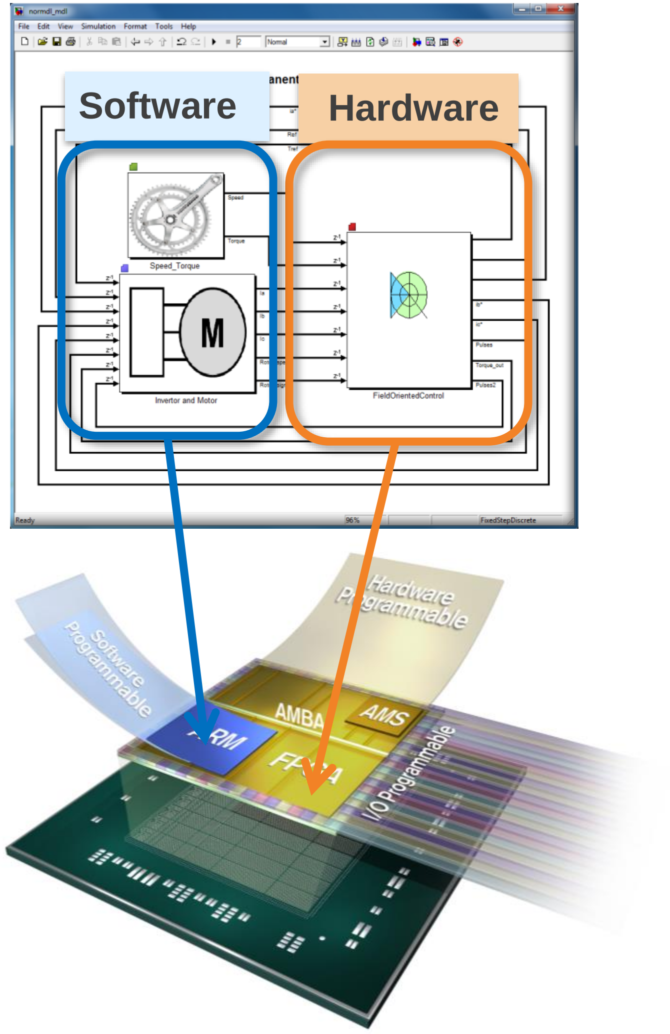
# Accelerate Deep Learning Prototype on Zynq with Matlab/Simulink

Algorithm Development

Neural Network Toolbox

MatConvNet

http://www.vlfeat.org/matconvnet/

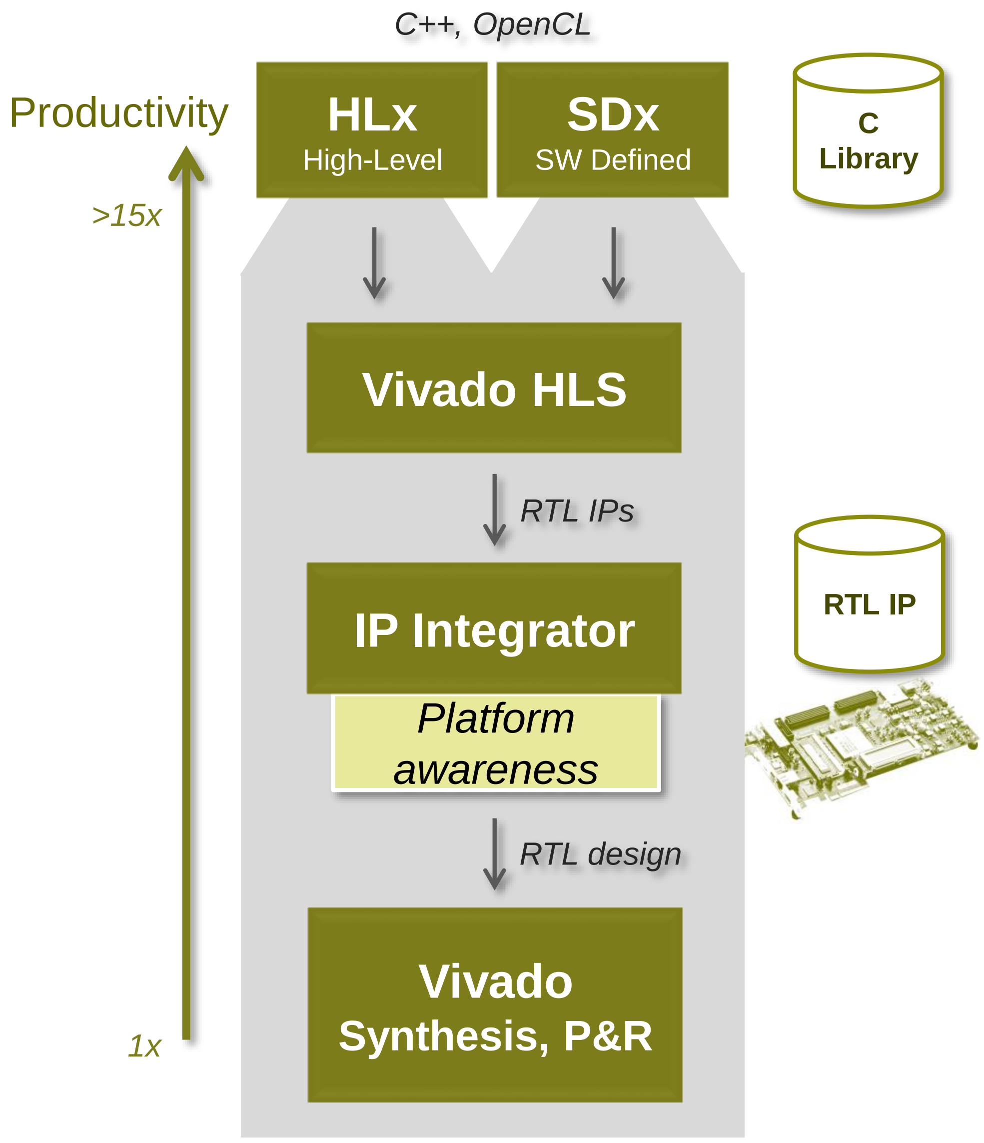



Automated Code Generation

Target HW Deployment

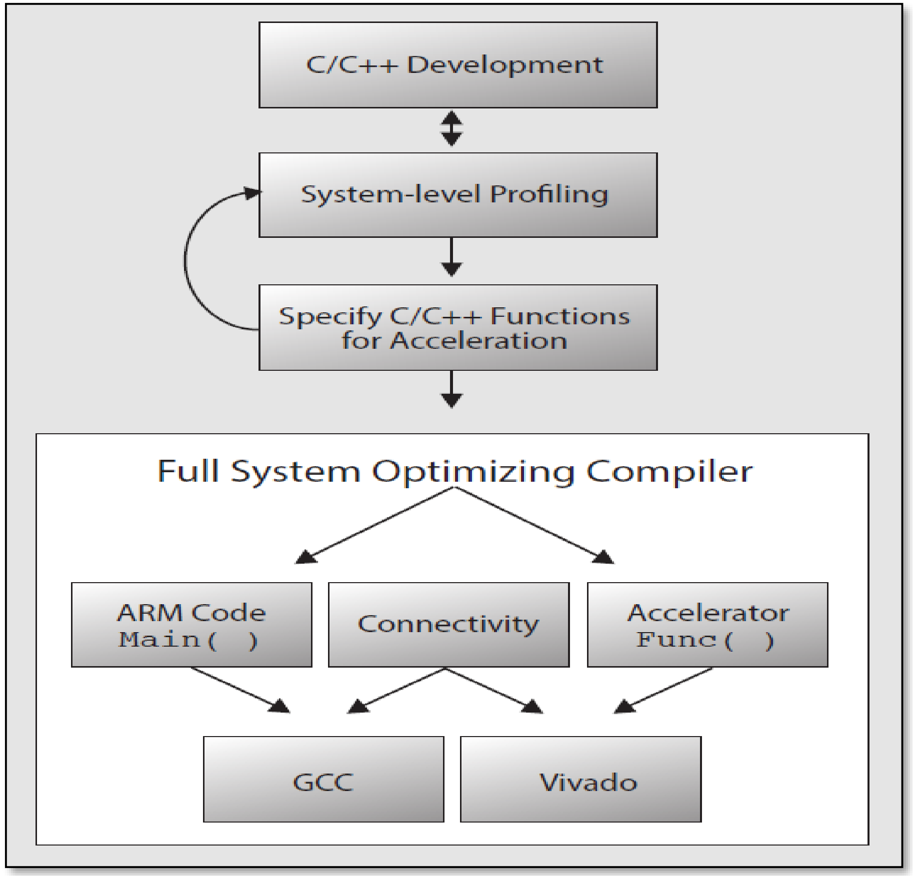
# HLx Summary – Accelerating Design Productivity

- **Separate platform design from differentiated logic**
  - Let application designers focus on the differentiated logic
- **Spend less time on the standard connectivity**
  - **IPI**: configure & generate a platform on a custom board
  - Use of Partial Reconfiguration to guarantee performance
- **Spend more time on the differentiated logic**
  - **HLS**: enabling core technology: C/C++/OpenCL synthesis
    - Exhaustive simulation, architecture exploration, code portability
  - HLx: Accelerates HW design: IP design (HLS/SysGen) + connectivity platform integration (IP Integrator)
  - SDx: Brings SW programmability to FPGA based platform

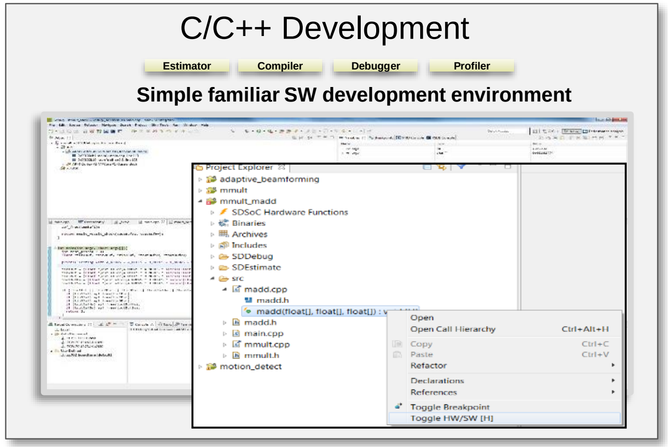


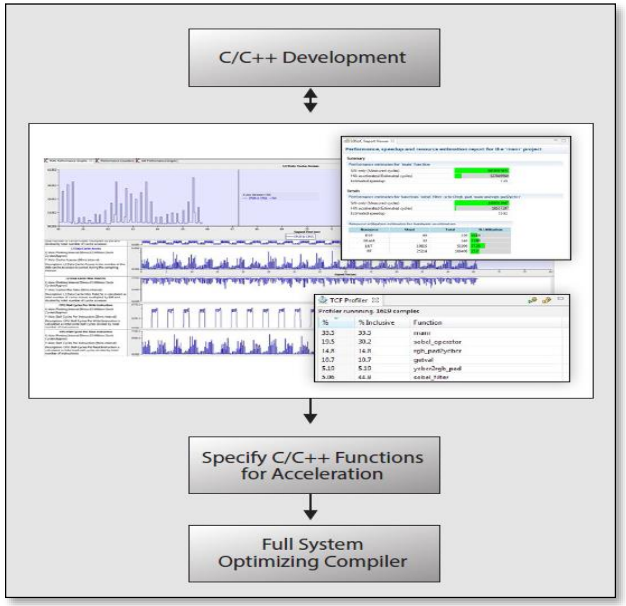
# The SDSoC Development Environment

SDSoC Environment




- ASSP-like programming experience
- System-level profiling
- Full system optimizing compiler
- Expert use model for platform developers and system architects



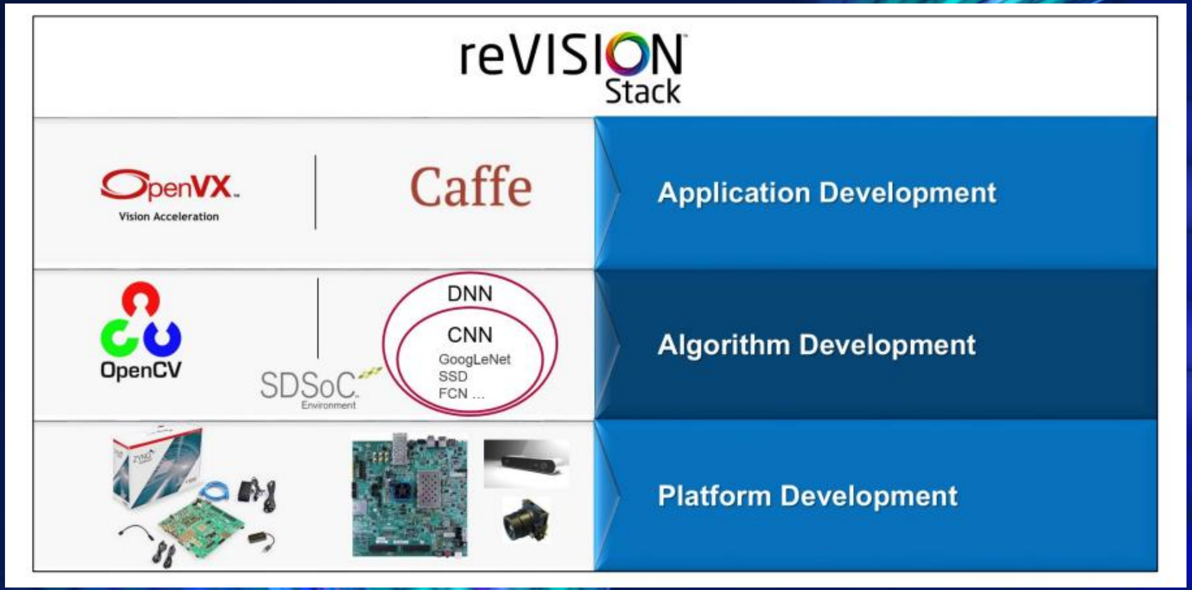

ZYNQ

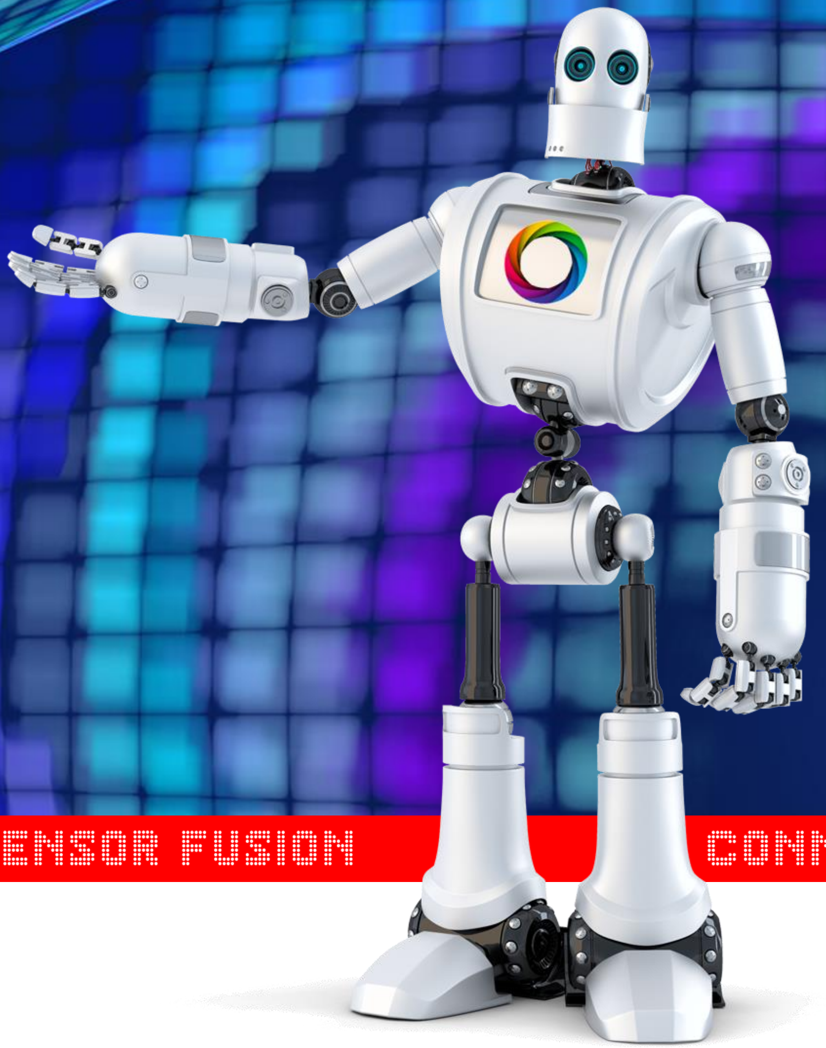
ZYNQ UltraSCALE+

XILINX ALL PROGRAMMABLE.

# reVISION™

Responsive and Reconfigurable Vision Systems



MACHINE LEARNING | COMPUTER VISION | SENSOR FUSION | CONNECTIVITY

# Xilinx Deep Learning Stack

1 Import .prototxt and trained weights

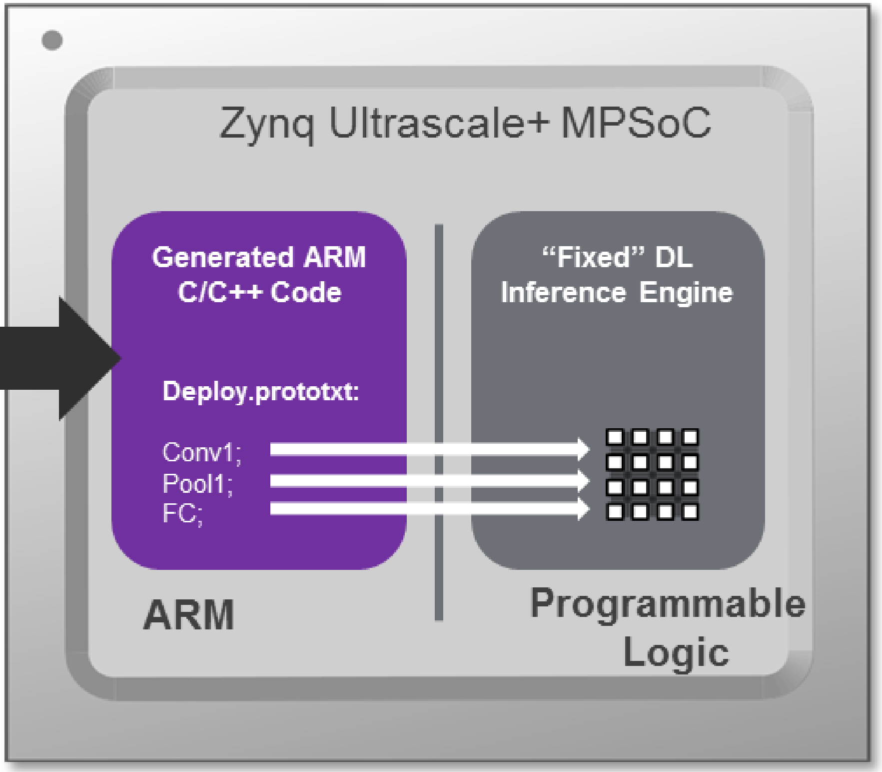
2 Call prototxt runtime API in your application

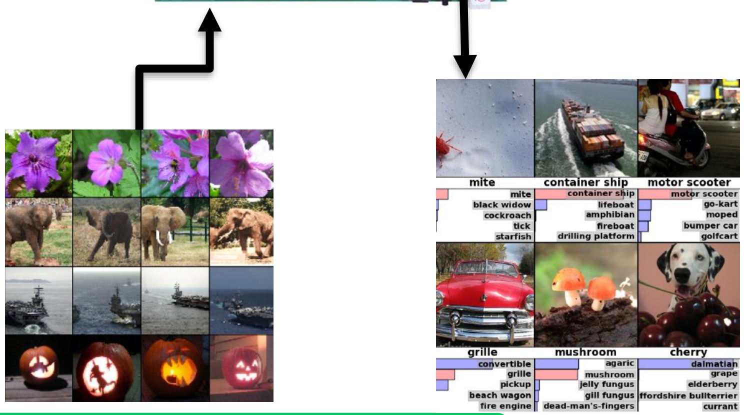
3 Cross-compile for Cortex-A53 and run on a board

Caffe Prototxt

-1 0 1
Filter Weights

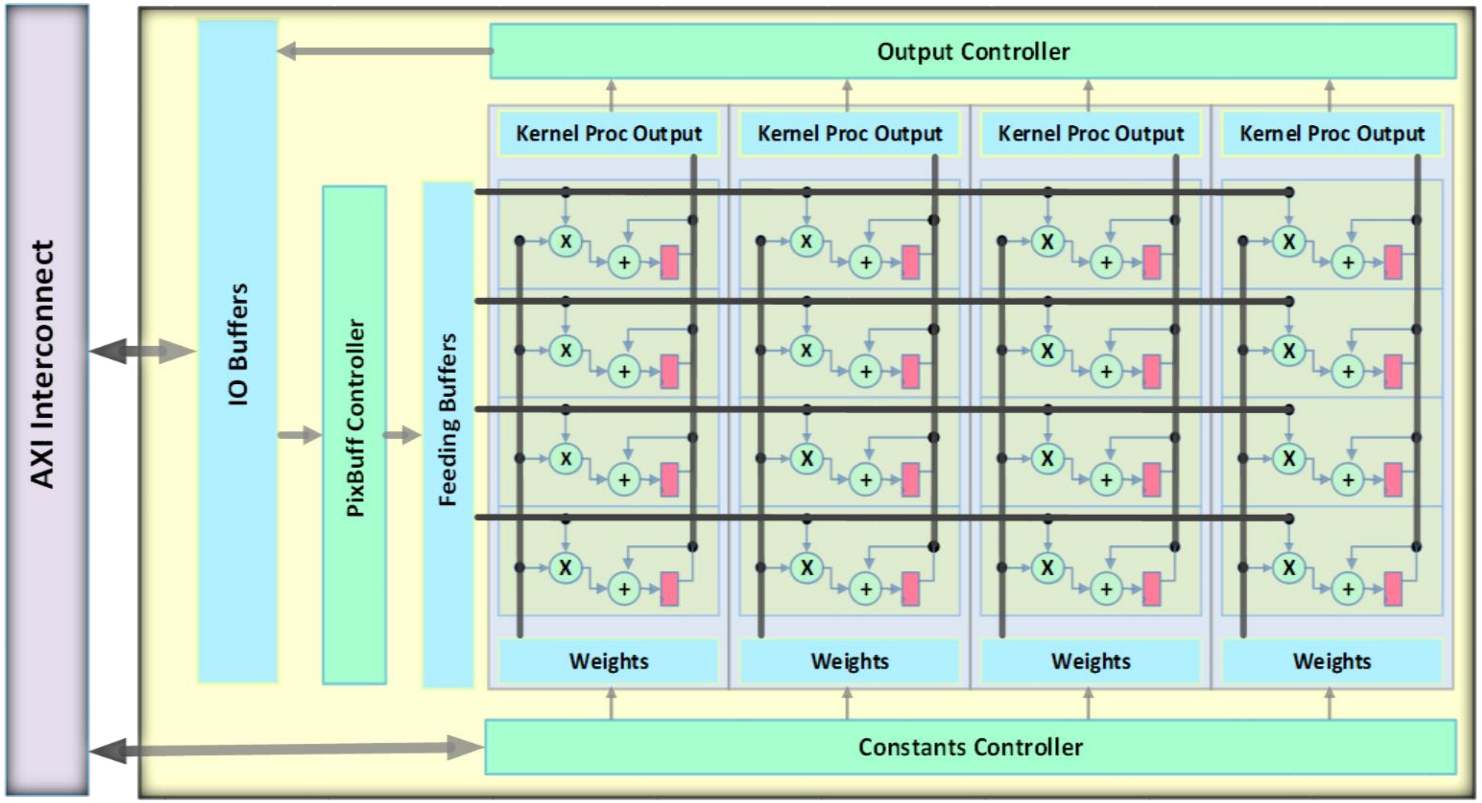
Target Data



**Compiles only ARM software code in minutes. No hardware compilation**

# DeepX: Deep Learning Inference Processor



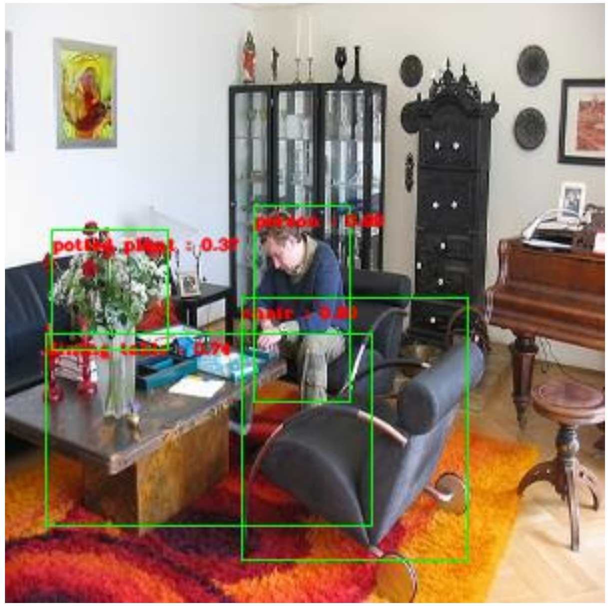

- Parameterized design. Scalable.
- Rich Instruction Set with 30+ opcodes.
  - Support for all popular networks.
- Mixed Precision support (16b, 8b).
- Simple Usage Model.

# CNN Functions in different networks

| Function\CNN | AlexNet | AlexNet FCN | VGG | GoogleNet | SqueezeNet | PVANet | ResNet | SSD |
|---|---|---|---|---|---|---|---|---|
| Convolution (2D) | Y | Y | Y | Y | Y | Y | Y | Y |
| ReLU activation | Y | Y | Y | Y | Y | Y | Y | Y |
| CReLU | N | N | N | N | N | Y | N | N |
| Fully connected | Y | N | Y | Y | N | Y | Y | Y |
| SoftMax | Y | N | Y | Y | Y | N | Y | Y |
| Deconv | N | Y | N | N | N | Y | N | N |
| Dilation | N | N | N | N | N | N | N | Y |
| NMS | N | N | N | N | N | N | N | Y |
| Permute | N | N | N | N | N | N | N | Y |
| Maxpool | Y | Y | Y | Y | Y | Y | Y | Y |
| Avg Pool | N | N | N | Y | Y | N | Y | Y |
| Concat | N | N | N | Y | Y | Y | N | Y |
| Eltwise | N | N | N | N | N | Y | Y | N |
| LRN Norm | N | N | N | Y | N | N | N | N |
| L2 Norm | N | N | N | N | N | N | N | Y |
| Batch Norm | Y | N | N | N | N | Y | Y | N |

# Deep Learning Design Examples

| | | May 2017 | Roadmap |
|---|---|---|---|
| GoogLeNet @ batch = 1 3.2 Gops/img | Images/s | **121** | **370** |
| | Power (W) | 6.0 | 7.0 |
| | Images/s/watt | **20.2** | **52.9** |
| SSD @ batch = 1 62.4 Gops/img | Images/s | **6.3** | |
| | Power (W) | 6.0 | |
| | Images/s/watt | **1.1** | |
| FCN-AlexNet @ batch = 1 42.0 Gops/img | Images/s | **7.0** | |
| | Power (W) | 6.0 | |
| | Images/s/watt | **1.2** | |
| VGG-16 @ batch = 1 30.9 Gops/img | Images/s | **14.5** | |
| | Power (W) | 6.0 | |
| | Images/s/watt | **2.4** | |
| AlexNet @ batch = 1 1.4 Gops/img | Images/s | **92** | |
| | Power (W) | 6.0 | |
| | Images/s/watt | **15.3** | |



- Programmable Logic running at 300 MHz, Input size: GoogLeNet, AlexNet, VGG-16 = 224x224, SSD = 300x300, FCN=480x480

# Development Kits

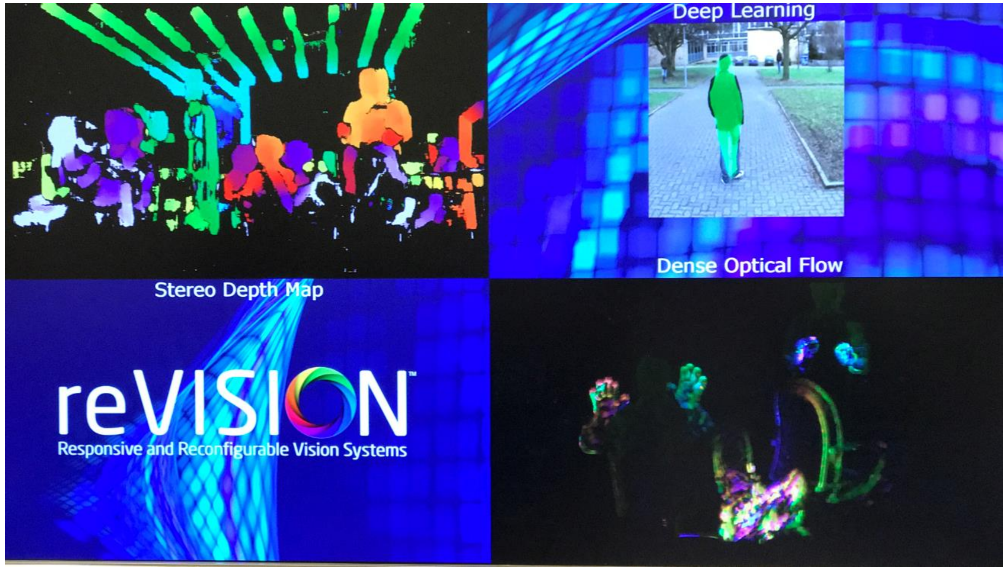
| Base Zynq Board | ZCU102 | ZCU104 | ZC702 | ZC706 |
|---|---|---|---|---|
| Device | ZU9 (16nm) | ZU7 (16nm) | Z7020 (28nm) | Z7045 (28nm) |
| CPU | Quad Cortex A53 up to 1.5GHz | | Dual Cortex A9 up to 1.0GHz | |
| Peak GOPS @ INT8 | 7857 | 5386 | 571 | 2331 |
| On-chip RAM (Mbits) | 32.1 | 38.0 | 4.9 | 19.1 |
| Inputs | USB3, MIPI, HDMI | USB3, MIPI, HDMI | HDMI* | HDMI* |
| Outputs | HDMI, DisplayPort | HDMI, DisplayPort | HDMI | HDMI |
| Video Codec Units | No | 4K60 Encode/Decode | No | No |
| reVISION Support | xFopencv, xFdnn | xFopencv, xFdnn | xFopencv, xFdnn | xFopencv, xFdnn |

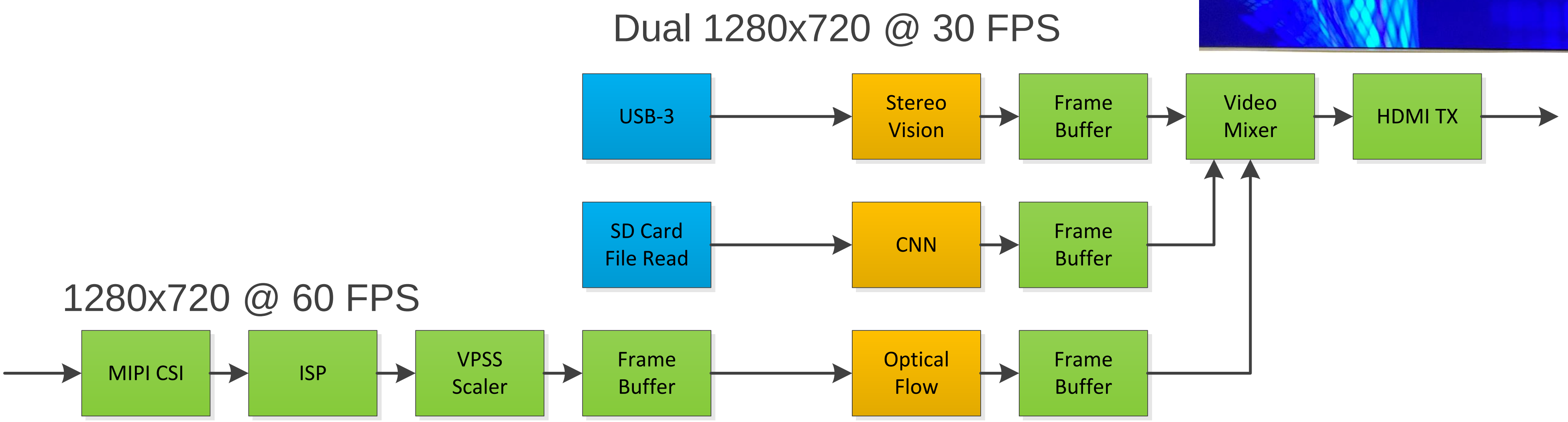
| Sensor Inputs | Sony IMX274 | Quad OnSemi AR0231 | StereoLab Zed Stereo | eCon camera |
|---|---|---|---|---|
| Spec | 3840x2160 @ 60 FPS | 1920x1080 @ 30 FPS | 3840x1080 @ 30 FPS | 1920x1080 @ 60 FPS |
| Interface | MIPI via FMC | MIPI via FMC | USB3 | USB3 |




* Requires an HDMI IO FMC card

# Optical Flow + Stereo Vision + Pedestrian Detection with Multiple Sensors

# Summary

- Machine learning inference poses great challenges for embedded system in computation and memory bandwidth

- FPGA is very suitable for machine learning inference

- Model-based design and optimized libraries accelerate customer design for machine learning applications

# Resources

- Deep Learning with INT8 Optimization on Xilinx Devices
  - https://www.xilinx.com/support/documentation/white_papers/wp486-deep-learning-int8.pdf
- Reduce Power and Cost by Converting from Floating Point to Fixed Point
  - https://www.xilinx.com/support/documentation/white_papers/wp491-floating-to-fixed-point.pdf
- Xilinx reVISION developer zone
  - https://www.xilinx.com/products/design-tools/embedded-vision-zone.html
- Xilinx Reconfigurable Acceleration Stack Accelerates Mainstream Adoption of Xilinx FPGAs in Hyperscale Data Centers
  - https://www.xilinx.com/support/documentation/backgrounders/acceleration-backgrounder.pdf
- WP477 UltraRAM: Breakthrough Embedded Memory Integration on UltraScale+ Devices
  - https://www.xilinx.com/support/documentation/white_papers/wp477-ultraram.pdf
- Virtex UltraScale+ FPGAs with HBM Technology
  - https://www.xilinx.com/video/fpga/virtex-ultrascale-plus-hbm-devices.html