



DEEP LEARNING INFERENCE WITH INTEL® FPGA

Class 1

Agenda

Refresher: Introduction to Deep Learning

Introduction to FPGAs for Software Developers

Why Intel® FPGAs for Inference

Objectives

You will be able to:

Define Artificial Intelligence, Machine Learning and Deep Learning

Differentiate Deep Learning inference and training

Explain the basic components of Deep Learning CNN models

Explain the components that make up an FPGA and how programs are mapped to them

Explain why FPGAs are good for inference application acceleration

AI is Transforming Industries



CONSUMER



HEALTH



FINANCE



RETAIL



GOVERNMENT



ENERGY



TRANSPORT



INDUSTRIAL



OTHER

EXAMPLES

Smart Assistants
Chatbots
Search
Personalization
Augmented Reality
Robots

Enhanced Diagnostics
Drug Discovery
Patient Care
Research
Sensory Aids

Algorithmic Trading
Fraud Detection
Research
Personal Finance
Risk Mitigation

Support Experience
Marketing
Merchandising
Loyalty
Supply Chain Security

Defense
Data Insights
Safety & Security
Resident Engagement
Smarter Cities

Oil & Gas Exploration
Smart Grid
Operational Improvement
Conservation

Automated Cars
Automated Trucking
Aerospace
Shipping
Search & Rescue

Factory Automation
Predictive Maintenance
Precision Agriculture
Field Automation

Advertising
Education
Gaming
Professional & IT Services
Telco/Media
Sports

The Flood of Data

BY 2020



The average internet user will generate

~1.5 GB OF TRAFFIC PER DAY



Smart hospitals will be generating over

3,000 GB PER DAY



Self driving cars will be generating over

4,000 GB PER DAY... EACH



A connected plane will be generating over

40,000 GB PER DAY



A connected factory will be generating over

1,000,000 GB PER DAY



RADAR ~10-100 KB PER SECOND

SONAR ~10-100 KB PER SECOND

GPS ~50 KB PER SECOND

LIDAR ~10-70 MB PER SECOND

CAMERAS ~20-40 MB PER SECOND

1 CAR 5 EXAFLOPS PER HOUR

All numbers are approximated.
<http://www.asco.com/c/en/us/solutions/service-provider/vni-network-traffic-forecast/infographic.html>
http://www.asco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html
<https://datafloq.com/read/self-driving-cars-create-2-petabytes-data-annually/172>
http://www.asco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html
http://www.asco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.html

Fast Evolution of Technology

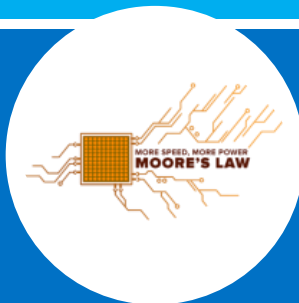
We have the compute capability to solve these problems today

Bigger Data



Image: 1000 KB / picture
Audio: 5000 KB / song
Video: 5,000,000 KB / movie

Better Hardware



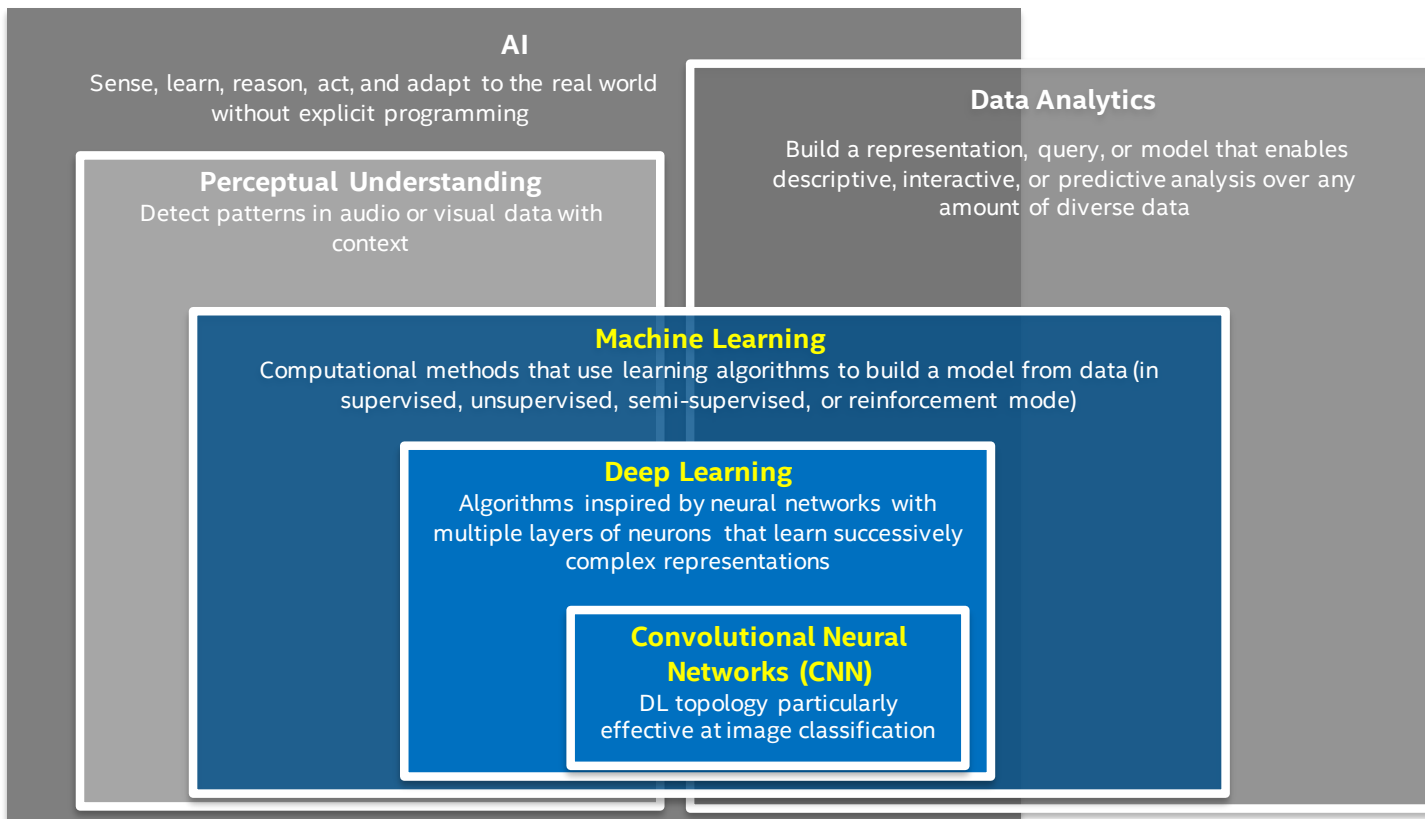
Transistor density doubles
every 18 months
Cost / GB in 1995: \$1000.00
Cost / GB in 2015: \$0.03

Smarter Algorithms

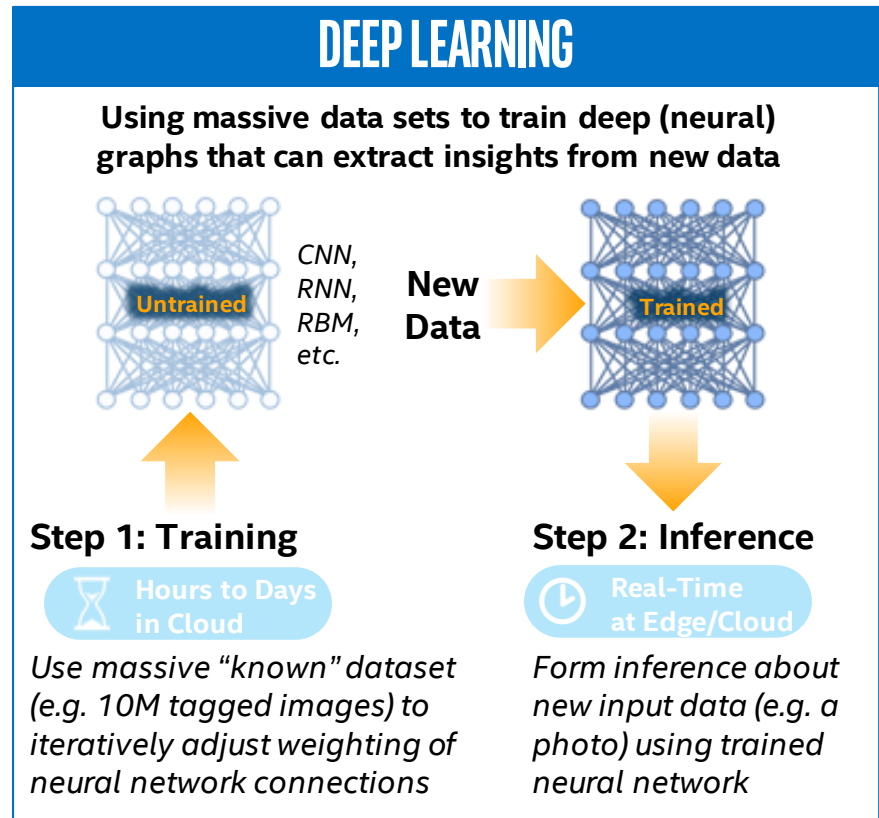
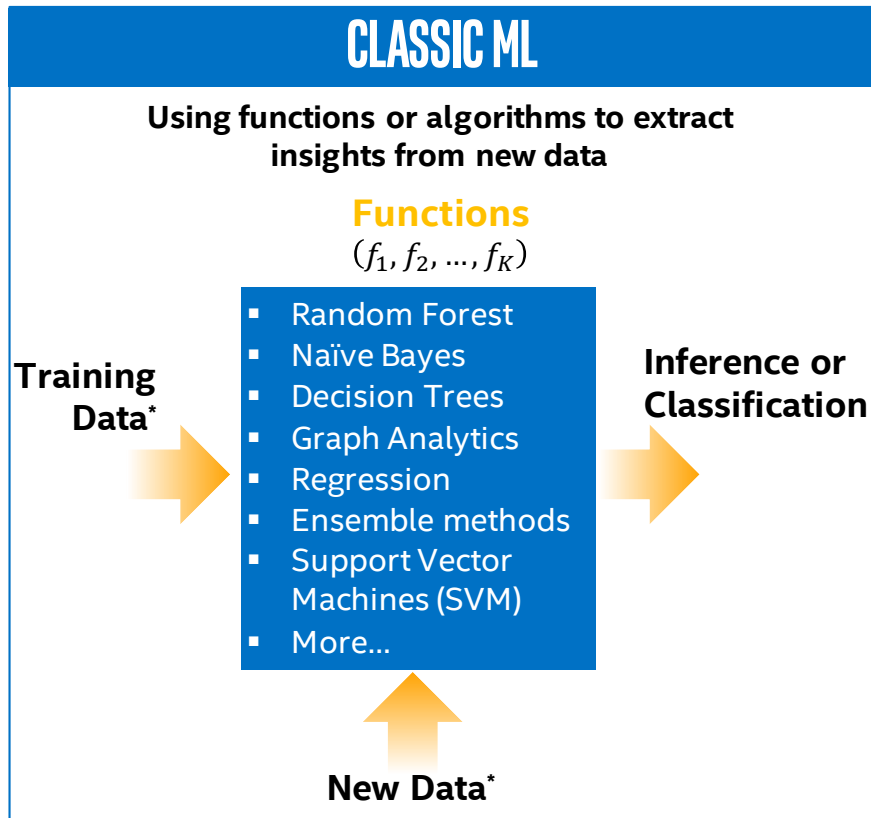


Advances in neural
networks leading to better
accuracy in training models

Taxonomic Foundations



Classical Machine Learning vs Deep Learning



Training vs Inference

Step 1: Training

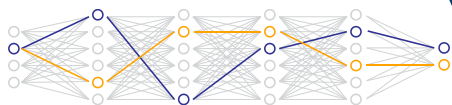
(In Data Center – Over Hours/Days/Weeks)

Lots of labeled
input data



Person

Create “Deep
neural net”
math model



Trained
Model

Output
Classification

90% person
8% traffic light

Also known as Learning

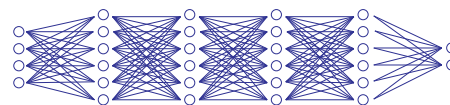
Step 2: Inference

(End point or Data Center - Instantaneous)



New input from
camera and
sensors

Trained neural
network model



97%
person

Output
Classification

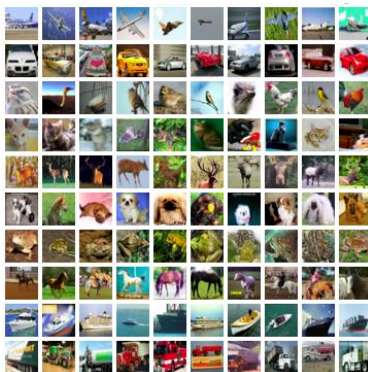
Also known as Classification, Scoring, Execution

Common Data Sets



MNIST

- 28 x 28 greyscale images
- 10 categories
- 60,000 training images
- 10,00 validation images



CIFAR-10

- 32 x 32 colour images
- 10 categories
- 50,000 training images
- 10,000 validation images



ImageNet

- 224 x 224 colour images
- 1000 categories
- 1.2M training images
- 50,000 validation images
- 100,000 test images

ImageNet Classification Competition

Recent winners all Deep Neural Nets!

- 2012 Winner: **AlexNet (University of Toronto)**, top-5 error rate of 15.3%, 5 convolution layers
- 2014 Winner: **GoogLeNet**, top-5 error rate of 6.67%, 22 layers in total
- 2015 Winner: **Microsoft (ResNet)** with 3.6% , 152 layers
- 2016 Winner: **3rd Research Institute of Ministry of Public Security, China** with 2.991%
- 2017 Winner: **WMW (Researchers from Momenta and Oxford)** with 2.251%

CPU's & FPGAs process thousands of frames of images per second

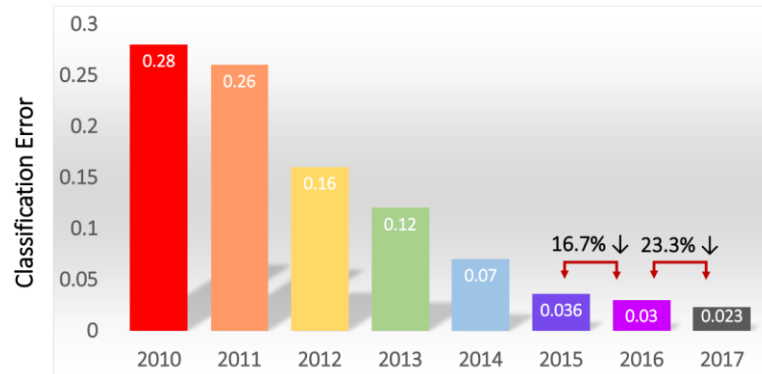
Trained human result: 5.1% Top-5 Error Rate @ 1 min/image



(a) Siberian husky



(b) Eskimo dog



Deep Learning Terminology

DL Network

- Layers and other hyperparameters
- e.g. AlexNet, GoogLeNet, VGG, ResNet, etc..

DL Frameworks

- High-level tools make it easier to build deep learning algorithms
- Provides definitions and tools to define, train, and deploy your network
- e.g. Caffe, TensorFlow™, MxNet, Caffe2, Theano, Torch, etc

DL Primitives Libraries

- Low-level accelerator specific libraries
- e.g. clDNN, MKL, DLA Suite , cuDNN, etc

A large, full-canopied tree with vibrant green leaves stands on a flat, green grassy field. The sky above is a clear blue with scattered, soft white clouds. The tree's trunk is dark and thick, branching out into a wide, rounded crown of leaves. The grass is a uniform green, and the overall scene is bright and clear.

Sky

Clouds

Tree

Grass

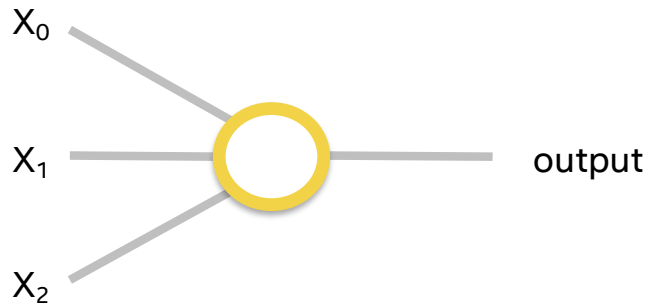
Perceptron

Simple model of a neuron

Developed in the 1950s and 1960s

Arbitrary number of inputs, single output

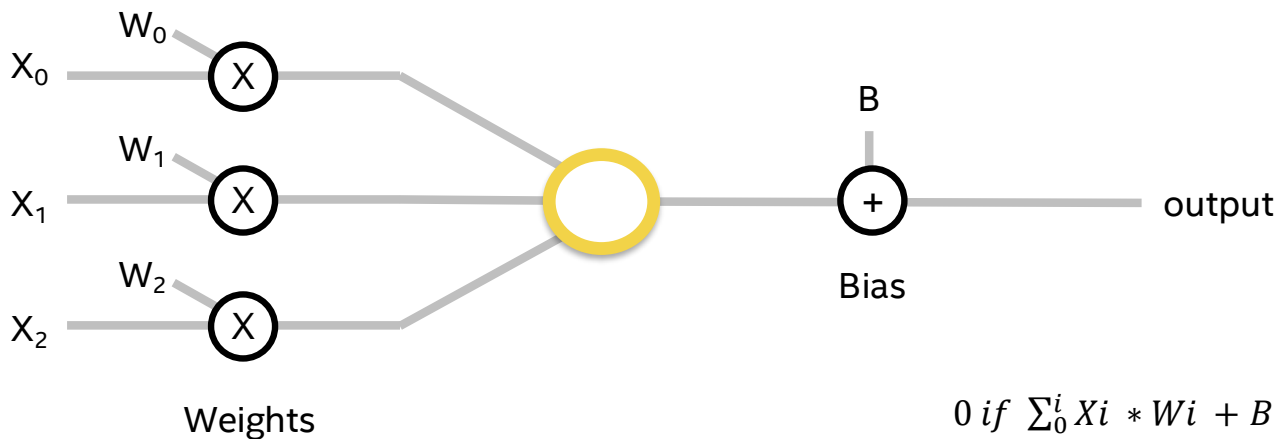
Binary inputs and output



Weights and Bias

Weights determine the level of influence for a given input

Bias impacts the likelihood of activation

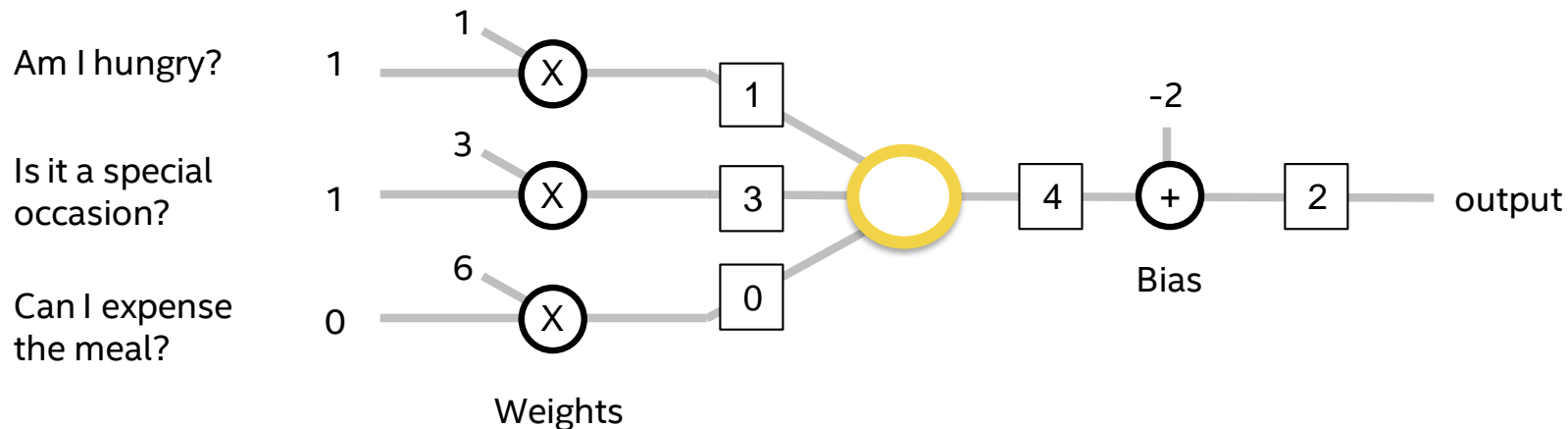


$$0 \text{ if } \sum_0^i X_i * W_i + B \leq 0$$

$$1 \text{ if } \sum_0^i X_i * W_i + B > 0$$

Example

Shall I eat out tonight?



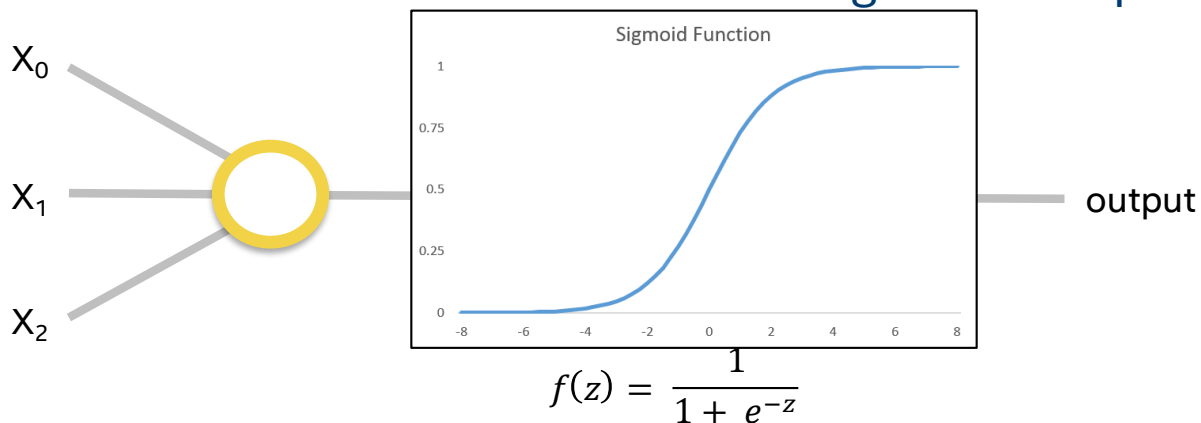
Sigmoid Neuron

Inputs and outputs range from 0.0 to 1.0

Same concept of weights and bias as a Perceptron

Sigmoid function applied to the output

- Neurons become less sensitive to small changes at the input and easier to train



Neural Networks

Network of interconnected Neurons

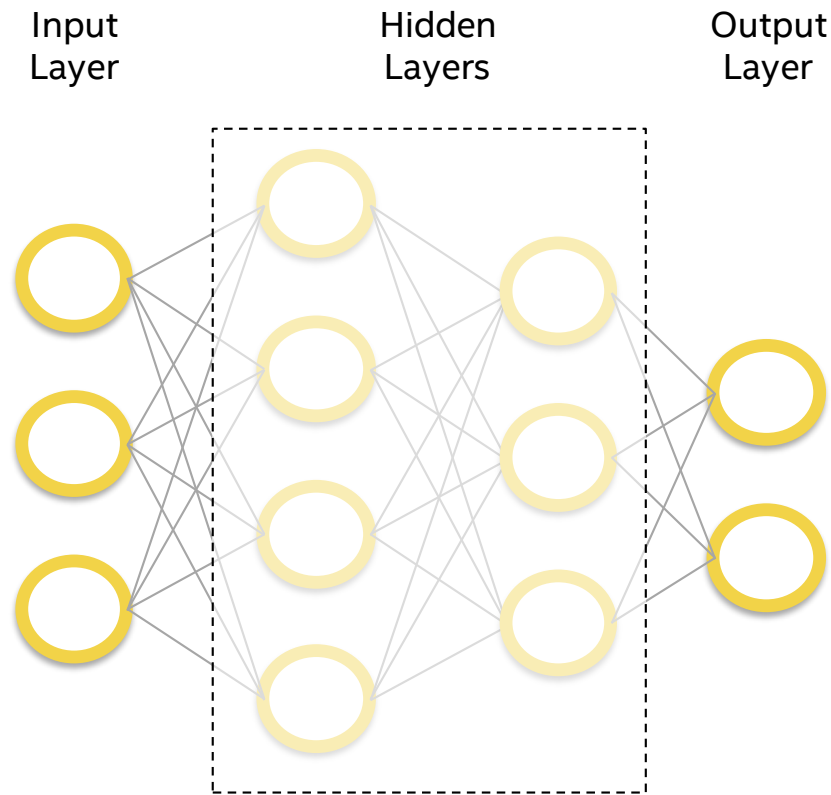
Each neuron in the input layer relates to a piece of input data

- E.g.: pixel for image classification

Each neuron in the output relates to a piece of output data

- E.g.: confidence of a car for image classification

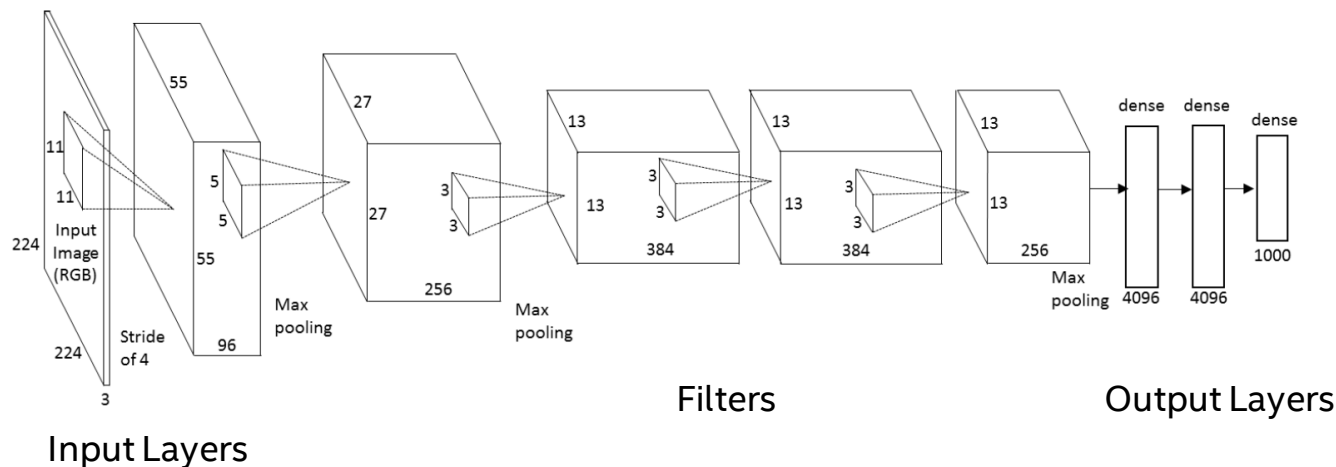
Diagram shows a Fully Connected Network



AlexNet

Common benchmark used by the industry

Based upon Convolutional Neural Network – CNN



*Image depicts the AlexNet **Graph** that describes the **Topology** of the network based upon the **Hyper-Parameters** that define the layer types, dimensions, filter dimensions, etc*

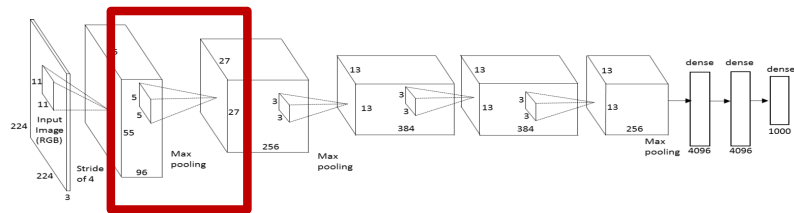
Why Convolution?

Reduces the data and number of calculations

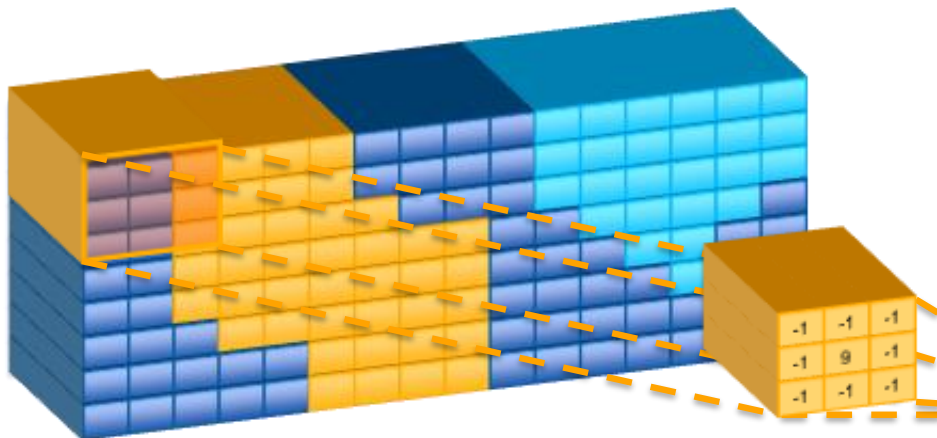
Take first two layers of AlexNet

Layer 1 neurons	Layer 2 neurons	Fully Connected Calculations	Convolutional Calculations
$224 \times 224 \times 3 = 150,528$	$55 \times 55 \times 96 = 290,400$	43 Billion	97 Million

Convolution Layers



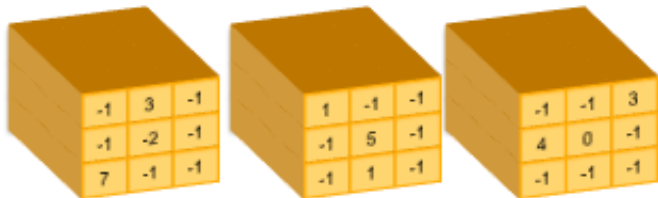
$$I_{\text{new}}[x][y] = \sum_{x'=-1}^1 \sum_{y'=-1}^1 I_{\text{old}}[x+x'][y+y'] \times F[x'][y']$$



Input Feature Map
(Set of 2D Images)

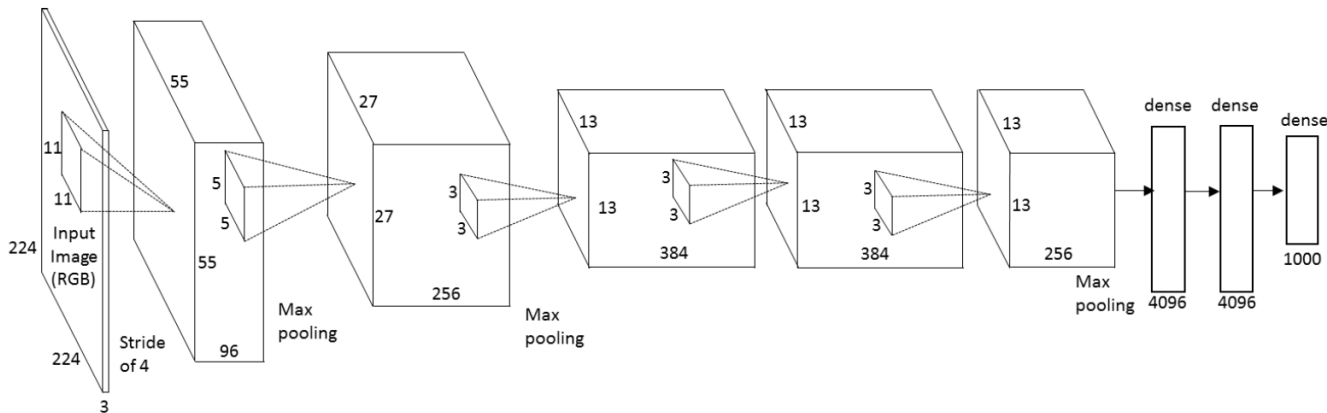
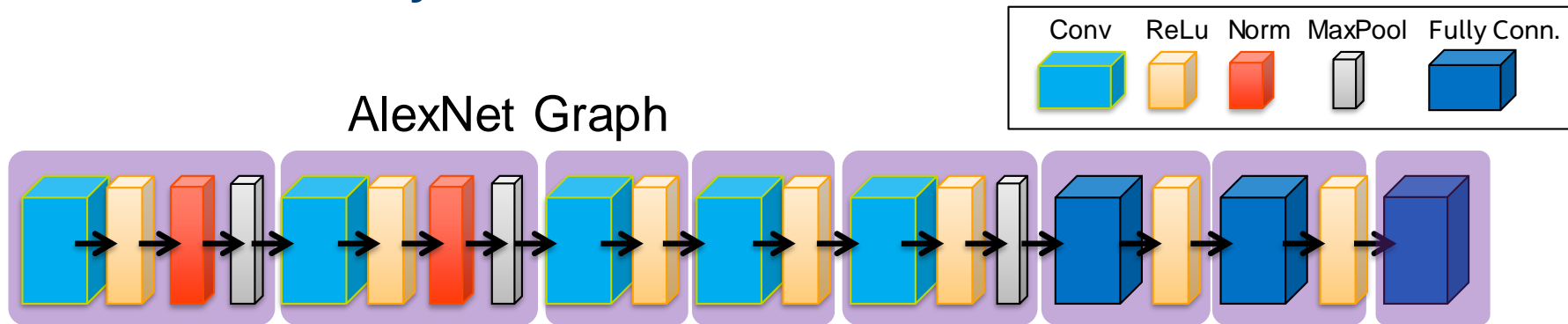
Filter
(3D Space)

Output Feature Map



**Repeat for Multiple Filters to Create
Multiple “Layers” of Output Feature Map**

Additional Layers Used in AlexNet

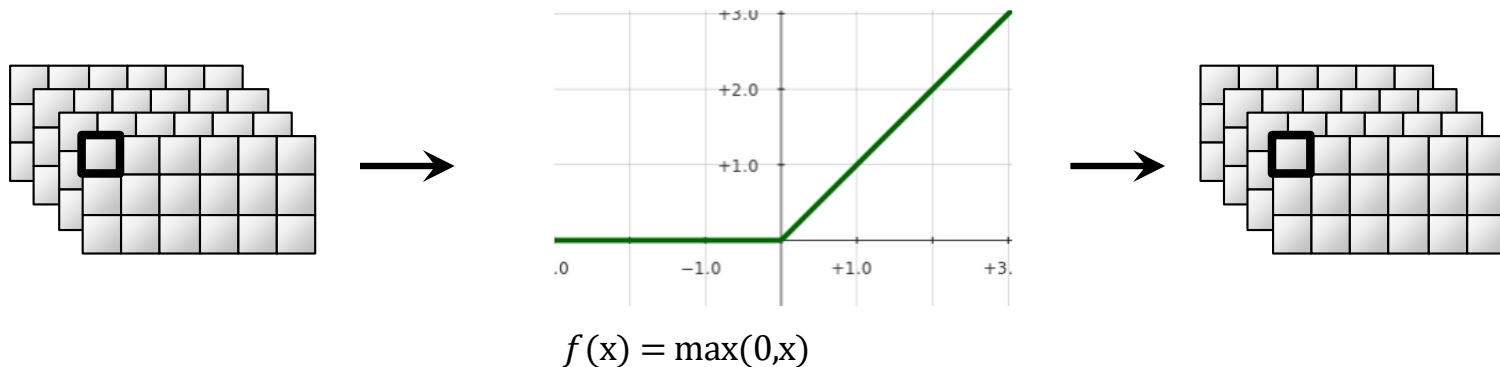


Rectified Linear Unit (ReLU)

Activation Layer

Similar function to Sigmoid function

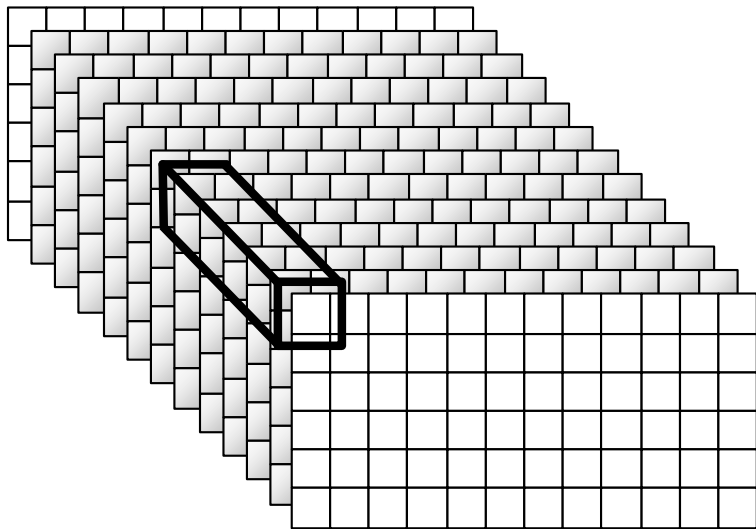
Applied to each neuron independently during the ReLU layer



Normalization (Local Response Normalization)

Smoothing function applied through the depth of the image layers

- Reduces the relative peaks in neighbouring filters
- Padding used to preserve output layer depth



$$b_{x,y}^i = a_{x,y}^i / \left(k + \alpha \sum_{j=\max(0, i-n/2)}^{\min(N-1, i+n/2)} (a_{x,y}^j)^2 \right)^\beta$$

AlexNet normalization function

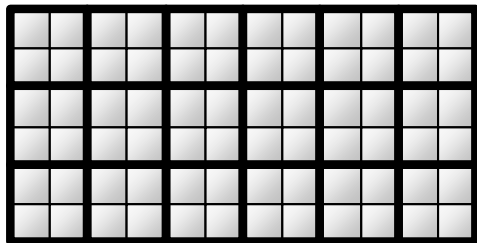
Pooling

Data reduction technique

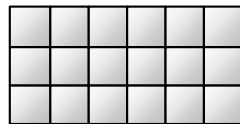
Performed on each layer

2 dimensional – reduces the width and height but not the depth

Common techniques are max pool and average pool



Input Layer



Output Layer



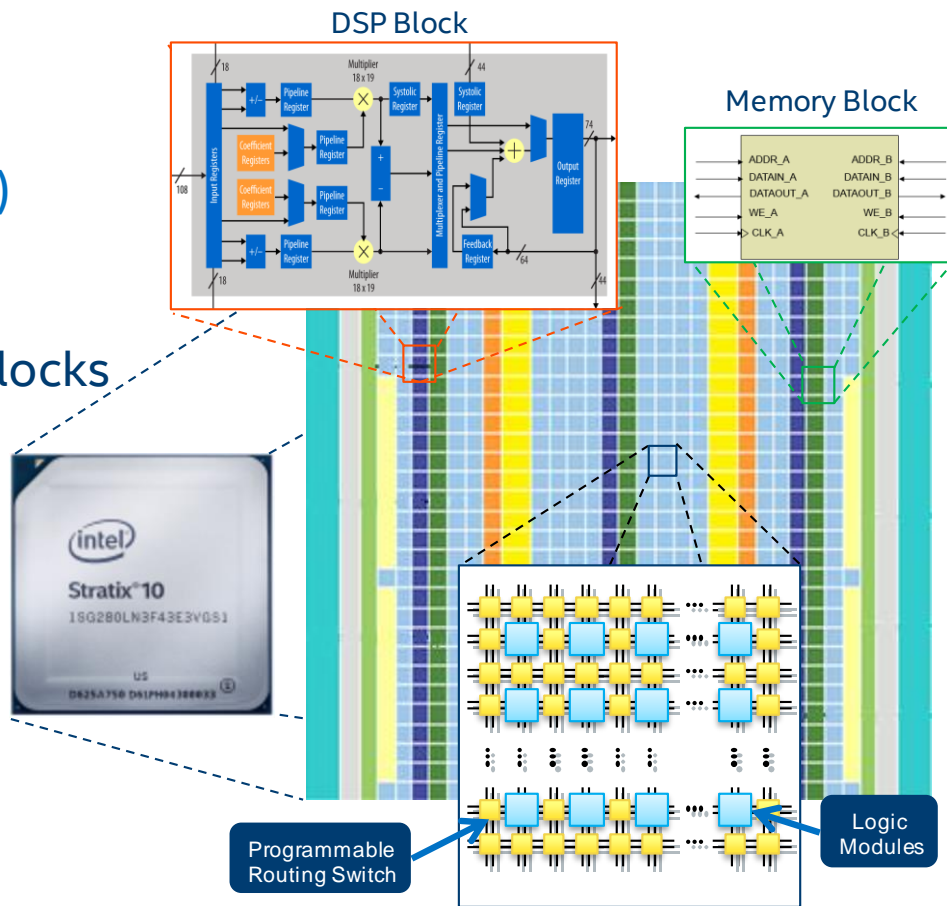
WHAT ARE FPGAS?

FPGA Architecture

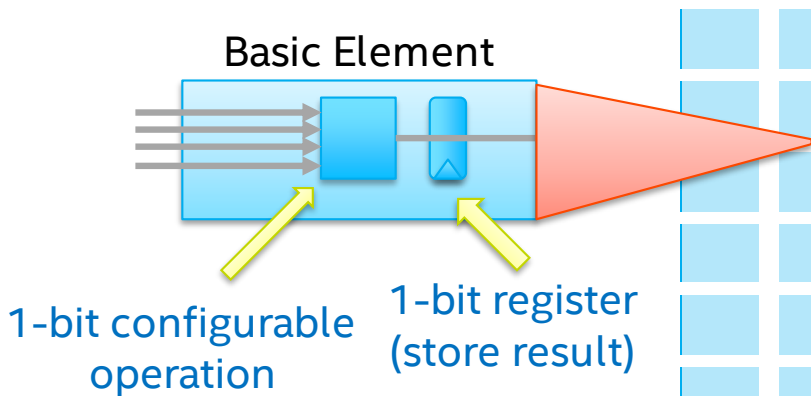
Field Programmable Gate Array (FPGA)

- Millions of logic elements
- Thousands of embedded memory blocks
- Thousands of DSP blocks
- Programmable interconnect
- High speed transceivers
- Various built-in hardened IP

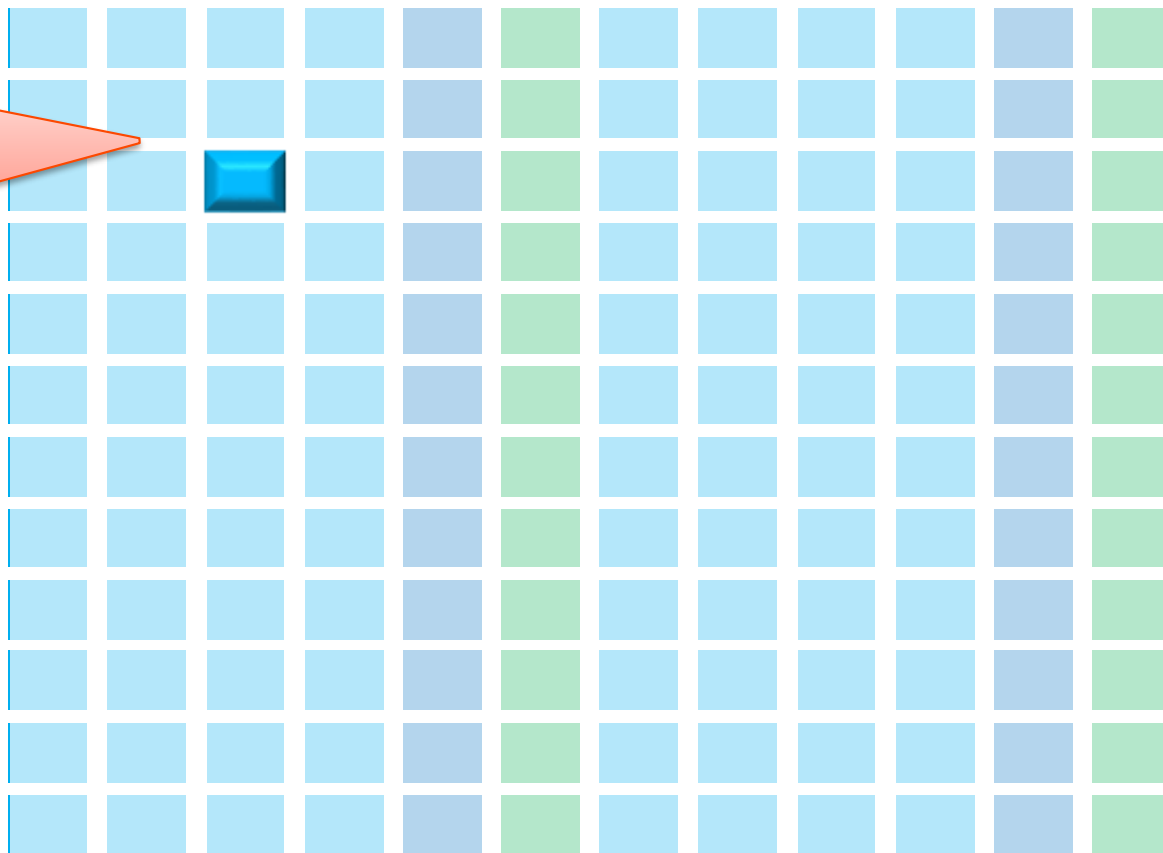
Used to create **Custom Hardware!**



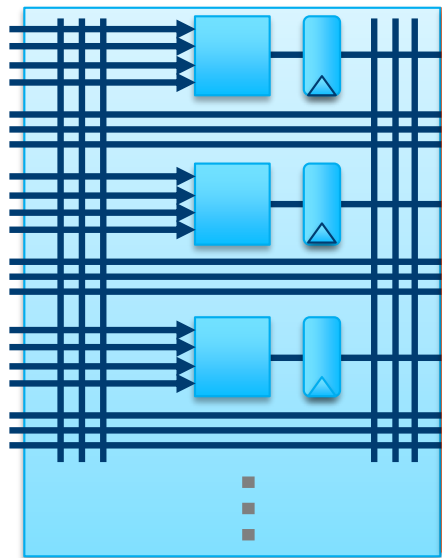
FPGA Architecture: Basic Elements



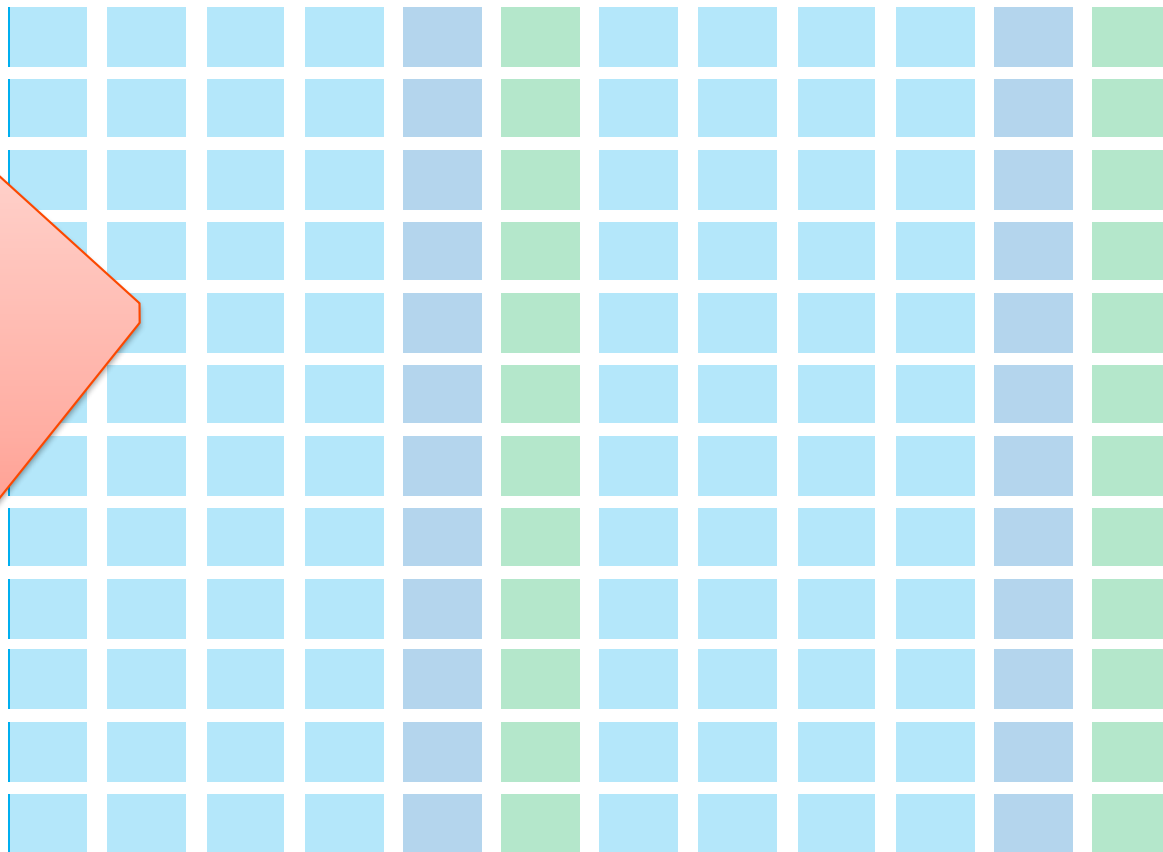
Configured to perform any
1-bit operation:
AND, OR, NOT, ADD, SUB



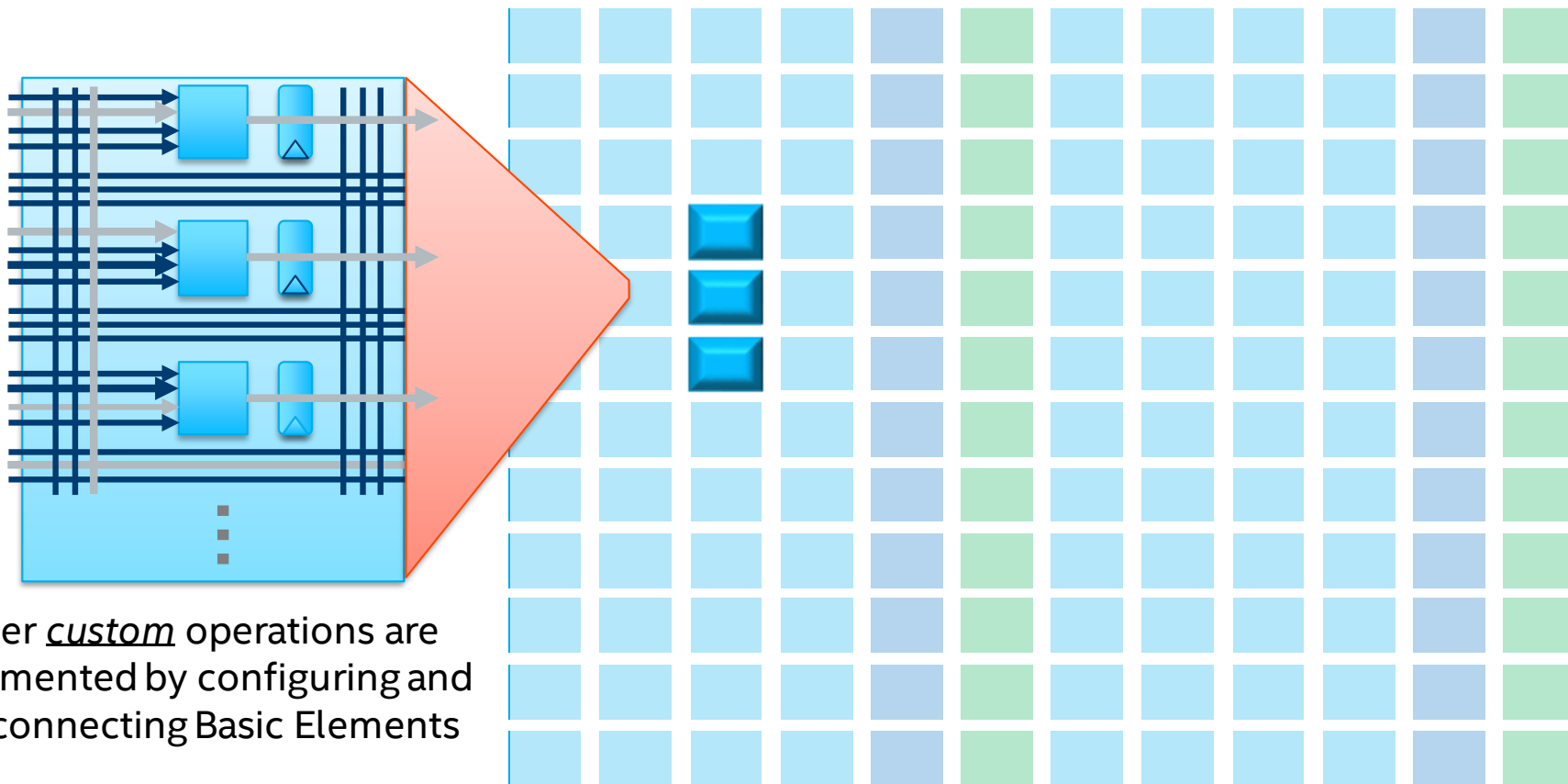
FPGA Architecture: Flexible Interconnect



Basic Elements are surrounded with a flexible interconnect

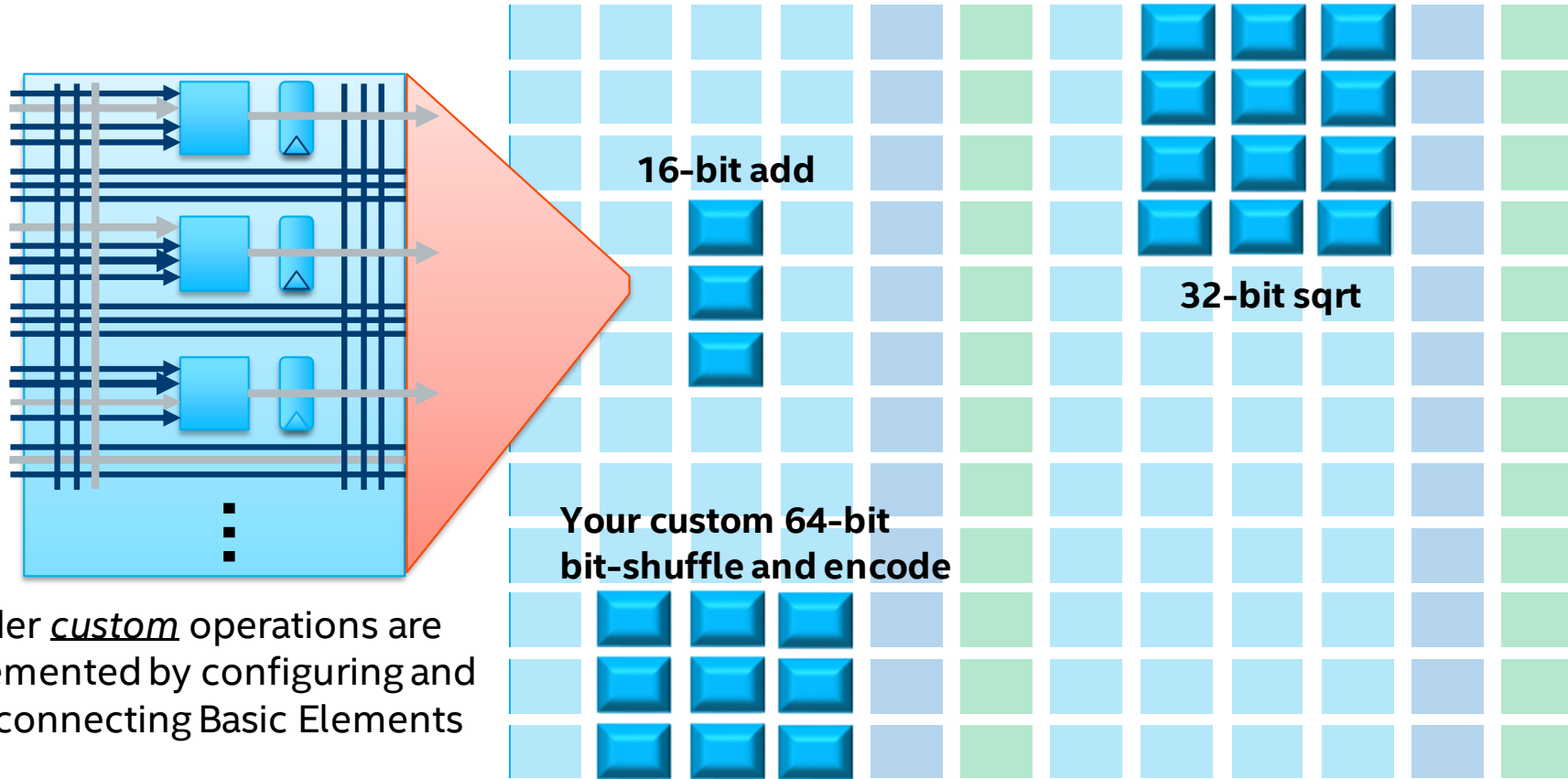


FPGA Architecture: Flexible Interconnect

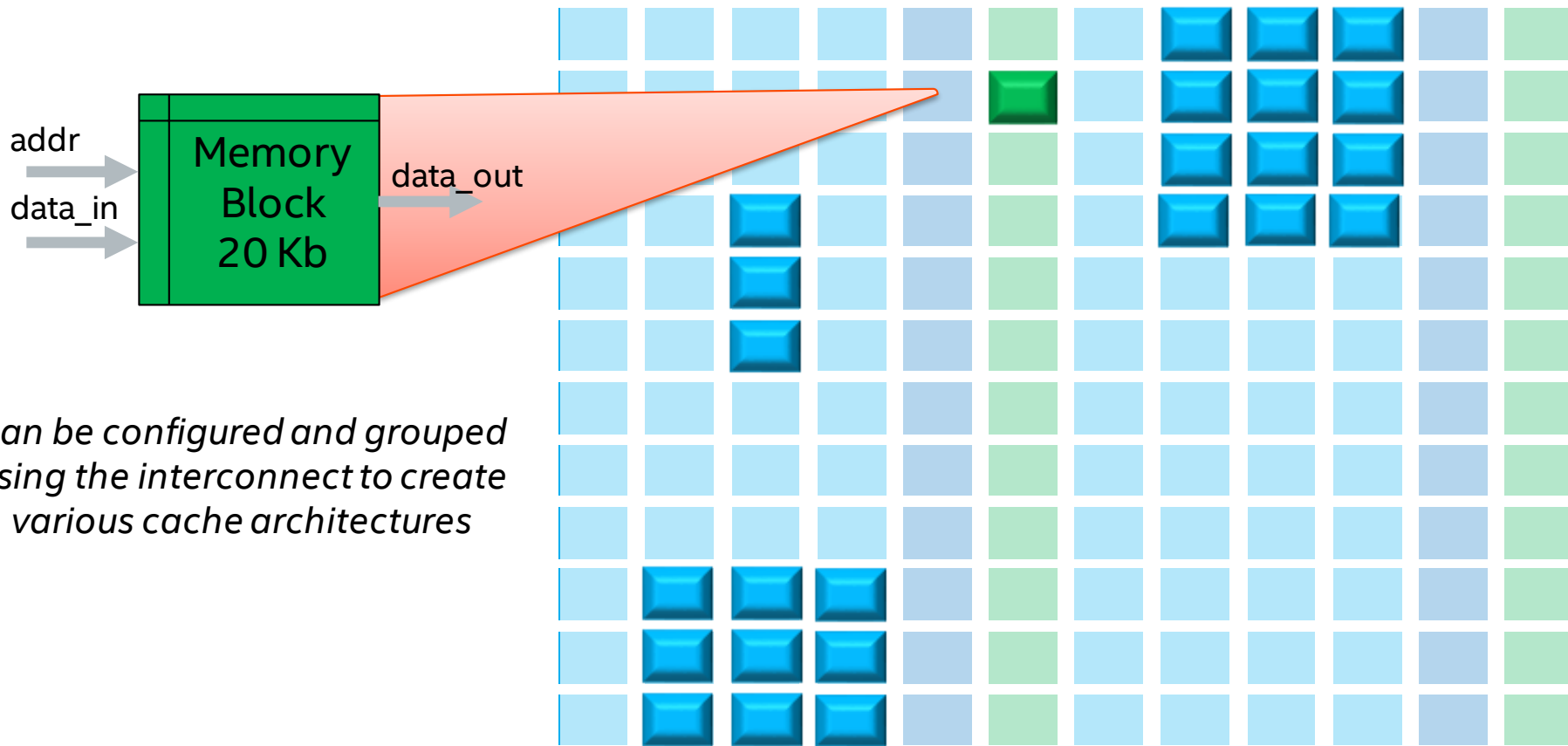


Wider custom operations are implemented by configuring and interconnecting Basic Elements

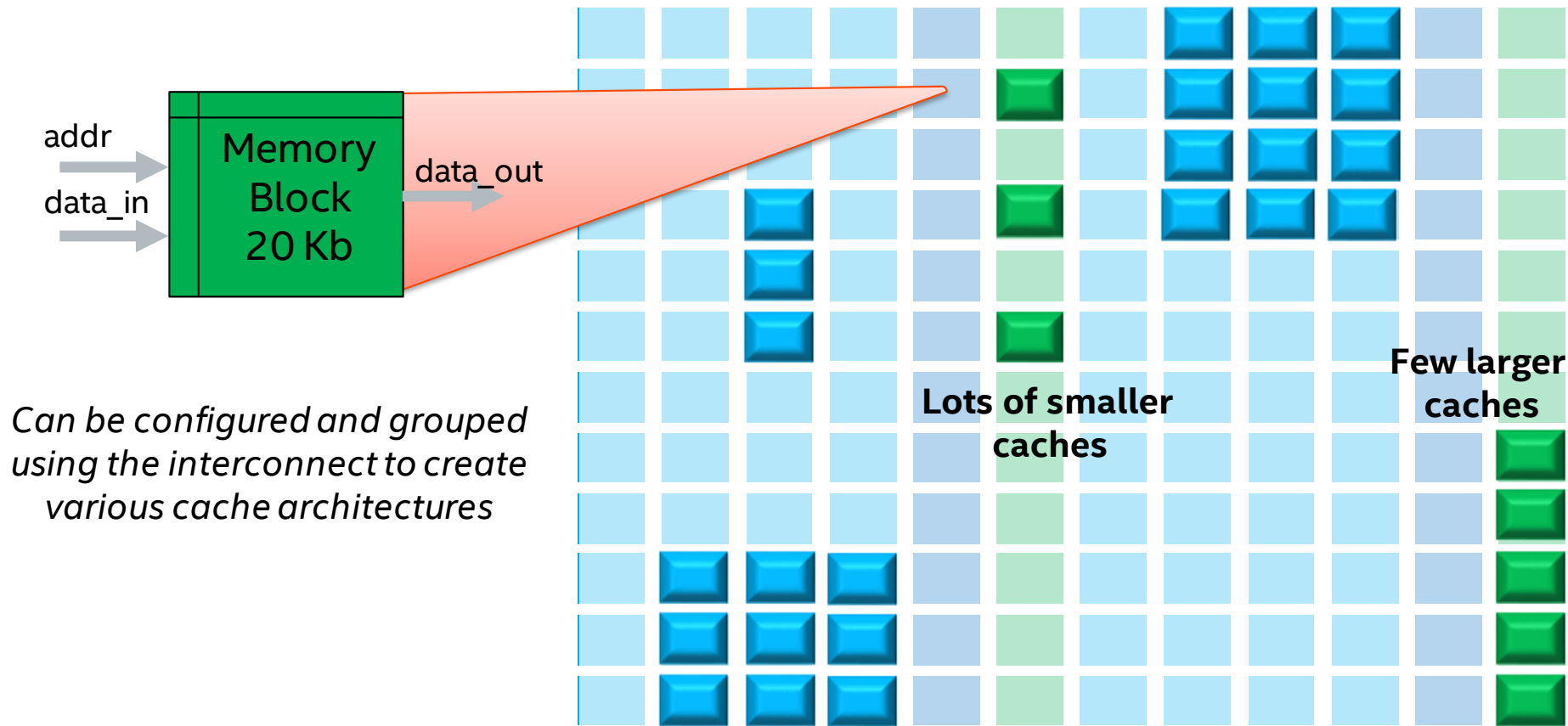
FPGA Architecture: Custom Operations Using Basic Elements



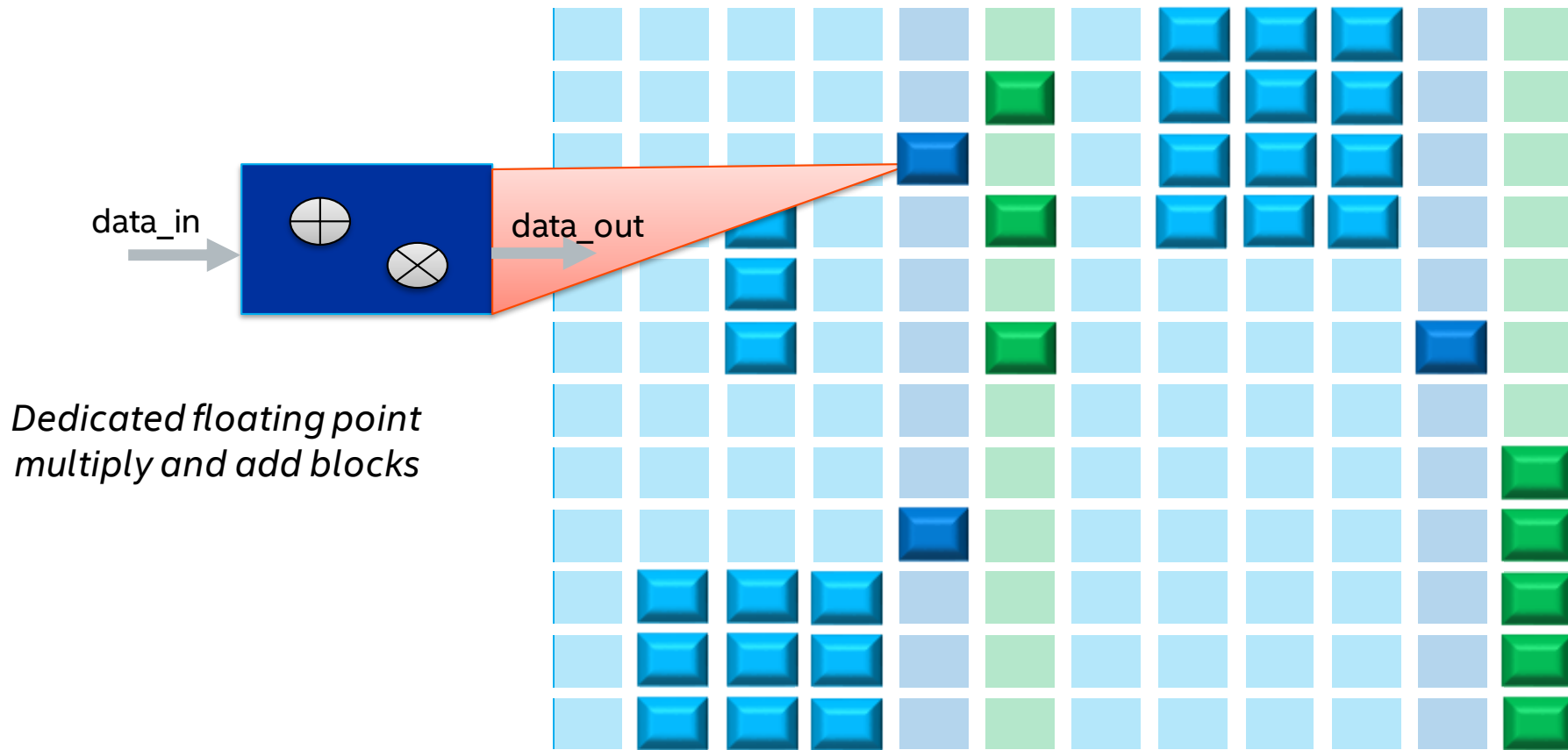
FPGA Architecture: Memory Blocks



FPGA Architecture: Memory Blocks



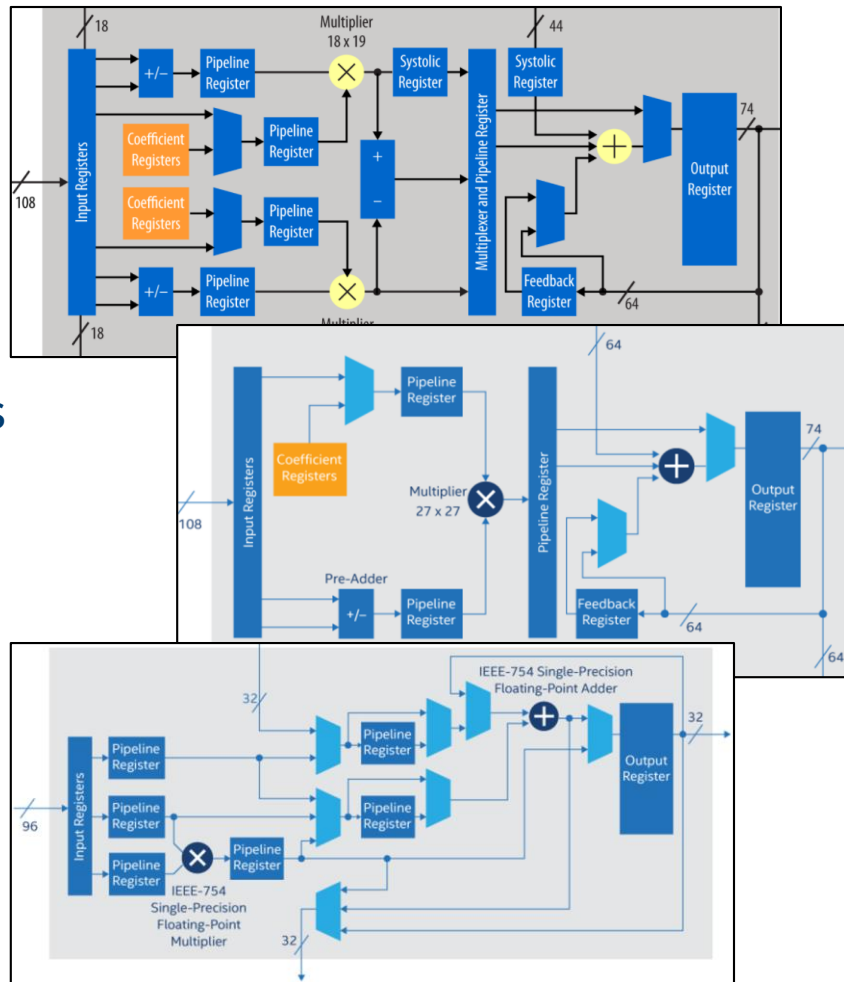
FPGA Architecture: Floating Point Multiplier/Adder Blocks



DSP Blocks

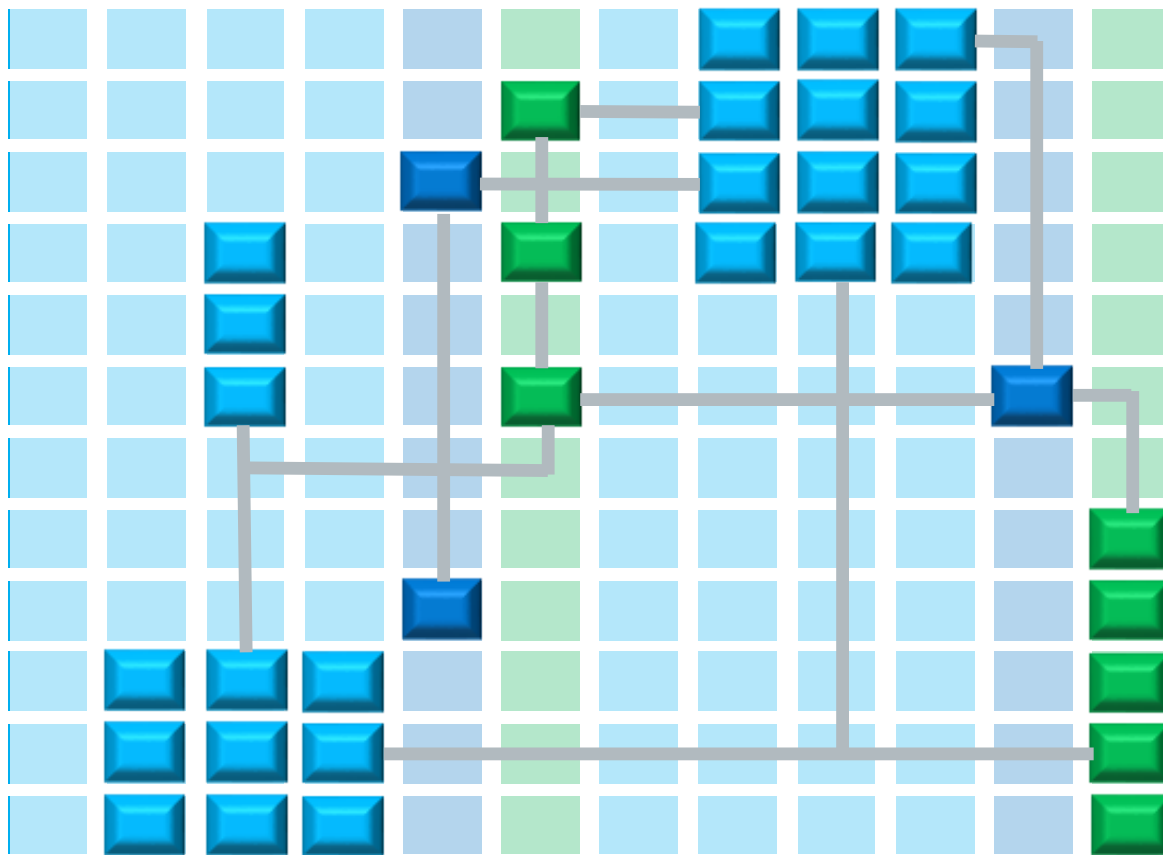
Thousands Digital Signal Processing (DSP) Blocks in Modern FPGAs

- Configurable to support multiple features
 - Variable precision fixed-point multipliers
 - Adders with accumulation register
 - Internal coefficient register bank
 - Rounding
 - Pre-adder to form tap-delay line for filters
 - Single precision floating point multiplication, addition, accumulation



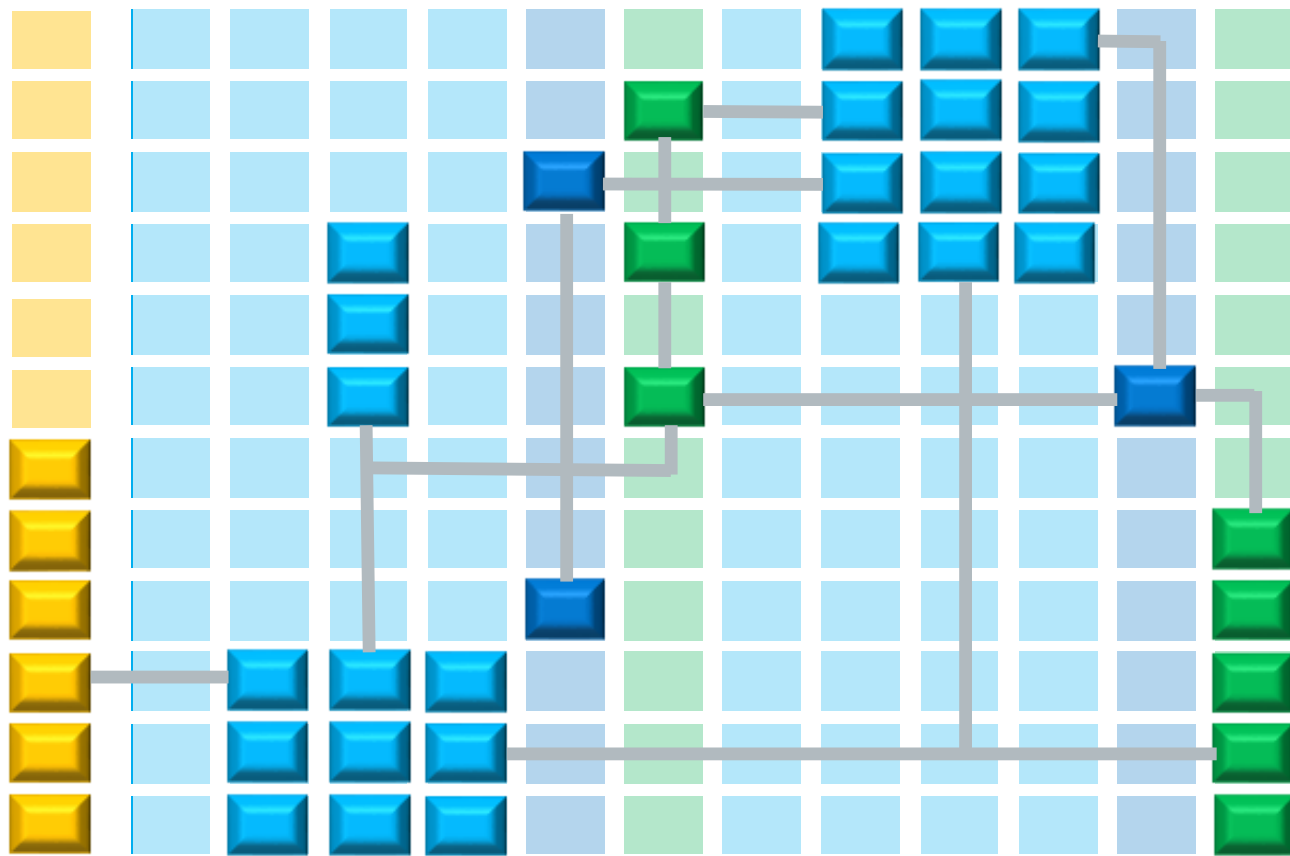
FPGA Architecture: Configurable Routing

Blocks are connected into a **custom data-path** that matches your application.



FPGA Architecture: Configurable IO

The **Custom data-path** can be connected directly to **custom or standard IO interfaces** for inline data processing



FPGA I/Os and Interfaces

Hardened Memory Controllers

- Available interfaces to off-chip memory such as HBM, HMC, DDR SDRAM, QDR SRAM, etc.

High-Speed Transceivers

- Provide any variety of protocols for moving data in and out of the FPGA

Hard IP for PCI Express standard

Phase Lock Loops (PLLs)

Intel® FPGA Product Portfolio

Wide range of FPGA products for a wide range of applications



- Products features differs across families
 - Logic density, embedded memory, DSP blocks, transceiver speeds, IP features, process technology, etc.

Mapping a Simple Program to an FPGA

Mem[100] += 42 * Mem[101]



CPU instructions

R0 ← Load Mem[100]

R1 ← Load Mem[101]

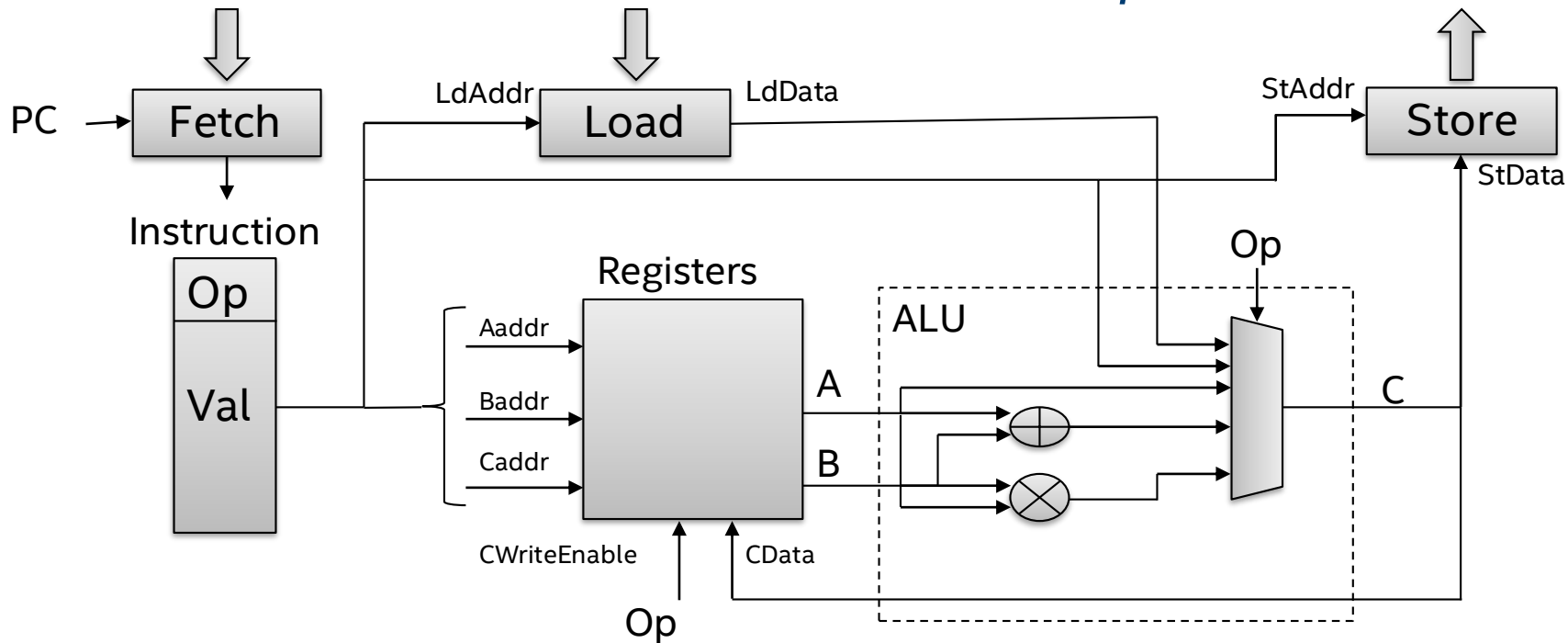
R2 ← Load #42

R2 ← Mul R1, R2

R0 ← Add R2, R0

Store R0 → Mem[100]

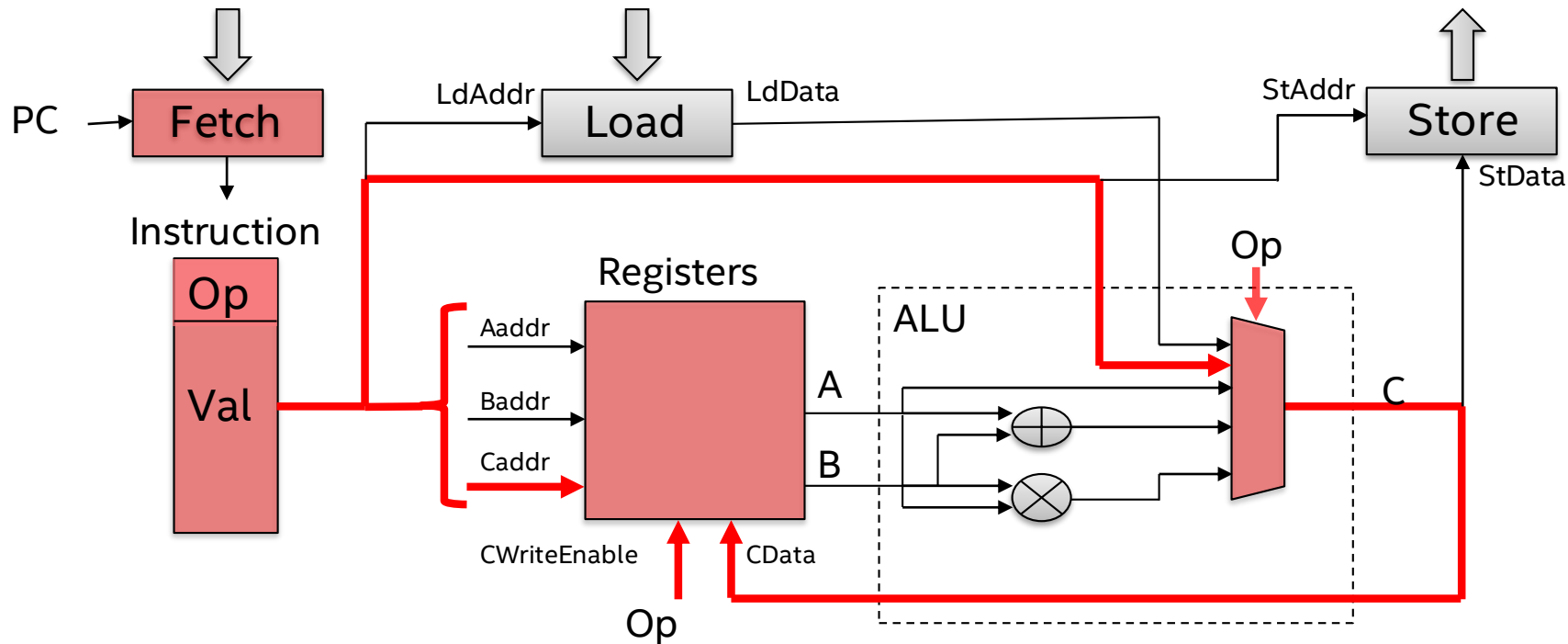
First let's take a look at execution on a simple CPU



**Fixed and general
architecture:**

- General “cover-all-cases” data-paths
- Fixed data-widths
- Fixed operations

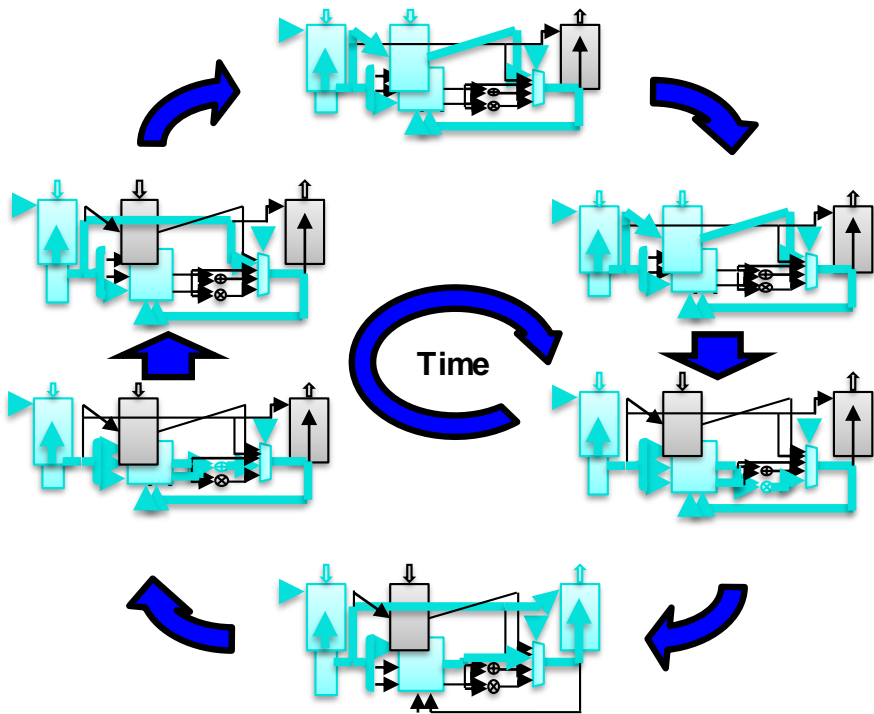
Looking at a Single Instruction



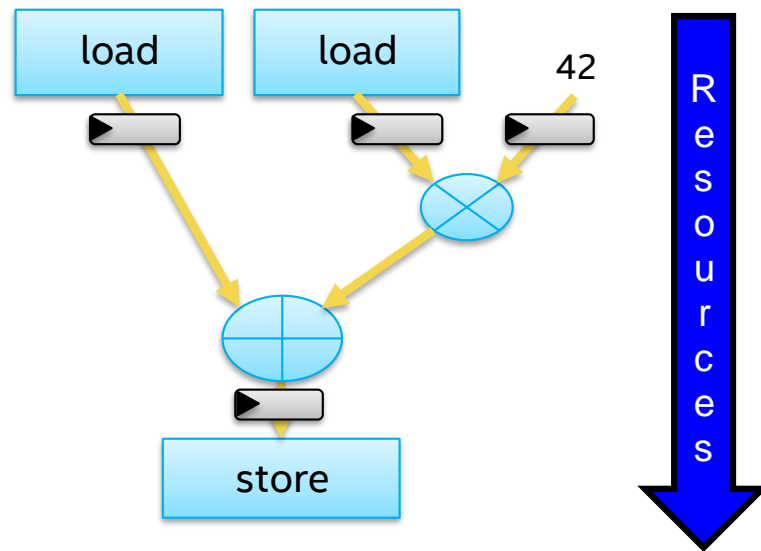
Very inefficient use of hardware!

Sequential Architecture vs. Dataflow Architecture

Sequential CPU Architecture



FPGA Dataflow Architecture

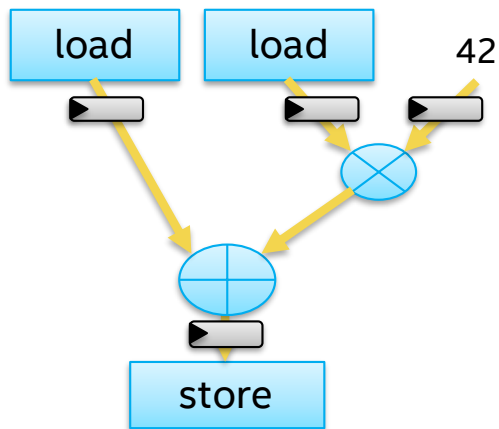


Custom Data-Path on the FPGA Matches Your Algorithm!

High-level code

```
Mem[100] += 42 * Mem[101]
```

Custom data-path



Build exactly what you need:

Operations

Data widths

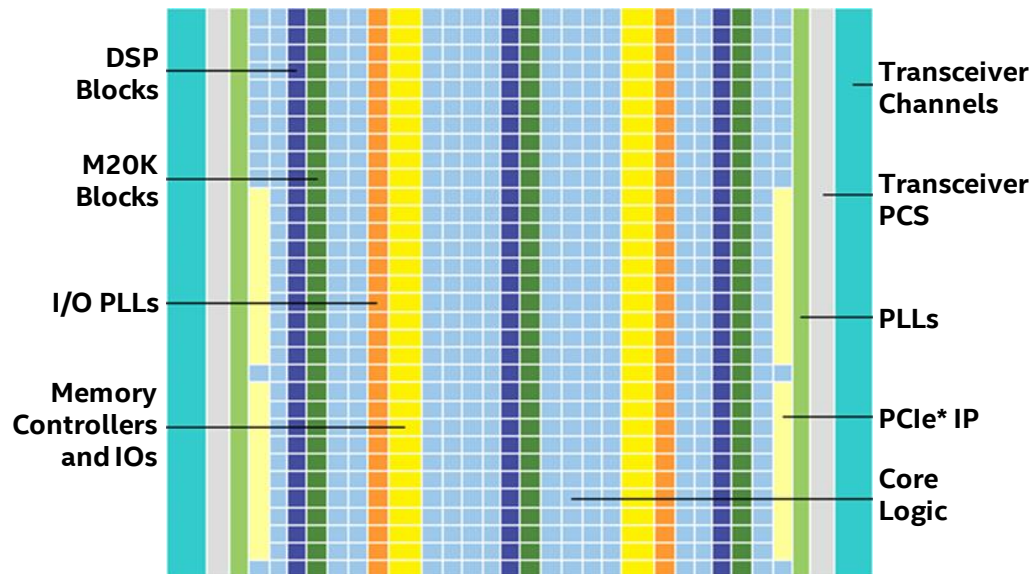
Memory size & configuration

Efficiency:

Throughput / Latency / Power

Advantages of Custom Hardware with FPGAs

- **Custom hardware!**
- Efficient processing
- Fine-grained parallelism
- Low power
- Flexible silicon
- Ability to reconfigure
- Fast time-to-market
- Many available I/O standards





WHY FPGAS FOR DL INFERENCE

Solving Challenges with FPGA



EASE-OF-USE

SOFTWARE ABSTRACTION,
PLATFORMS & LIBRARIES

Intel FPGA solutions enable software-defined programming of customized machine learning accelerator libraries.



REAL-TIME

DETERMINISTIC
LOW LATENCY

Intel FPGA hardware implements a deterministic low latency data path unlike any other competing compute device.



FLEXIBILITY

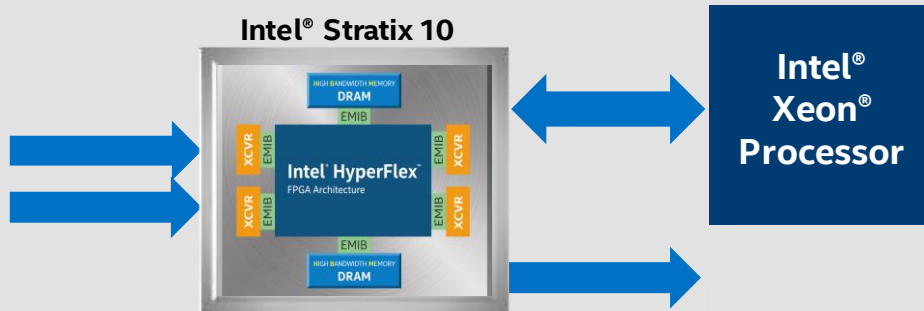
CUSTOMIZABLE HARDWARE
FOR NEXT GEN DNN ARCHITECTURES

Intel FPGAs can be customized to enable advances in machine learning algorithms.

FPGAs Provide Flexibility to Control the Data path

Compute Acceleration/Offload

- Workload agnostic compute
- FPGAaaS
- Virtualization



Inline Data Flow Processing

- Machine learning
- Object detection and recognition
- Advanced driver assistance system (ADAS)
- Gesture recognition
- Face detection

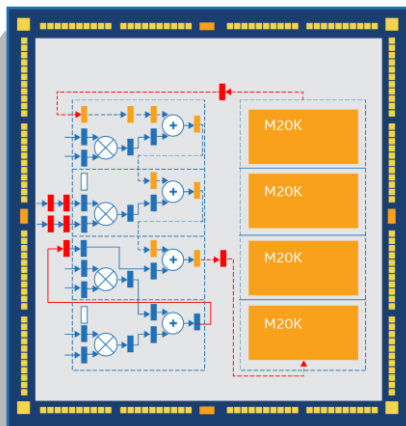


Storage Acceleration

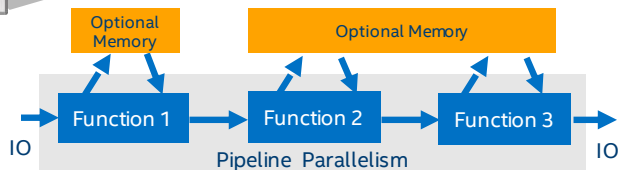
- Machine learning
- Cryptography
- Compression
- Indexing

Why Intel® FPGAs for Machine Learning?

Convolutional Neural Networks are Compute Intensive



**Fine-grained & low latency
between compute and memory**

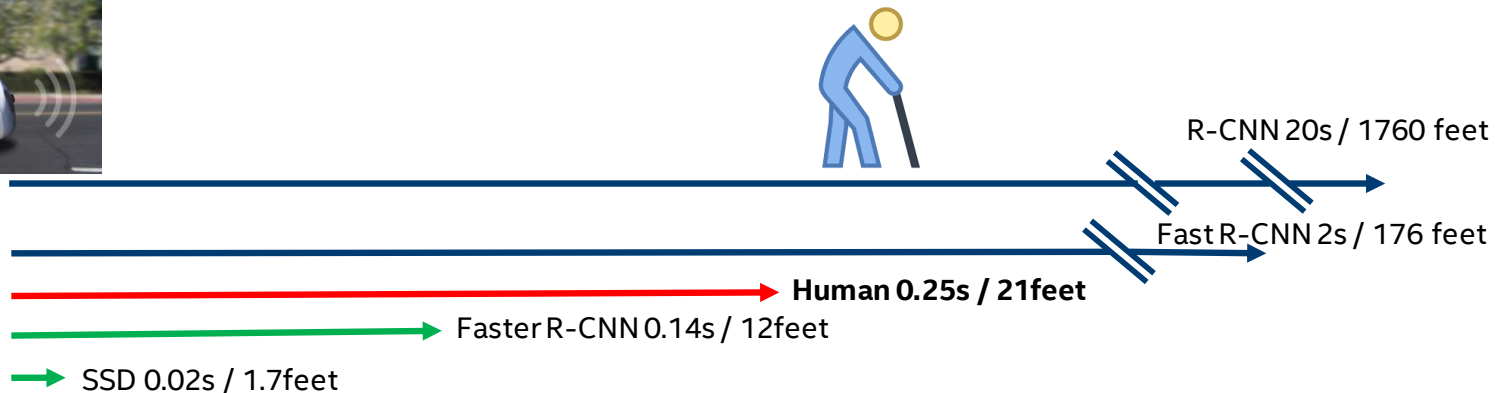


Feature	Benefit
Highly parallel architecture	Facilitates efficient low-batch video stream processing and reduces latency
Configurable Distributed Floating Point DSP Blocks	FP32 9Tflops, FP16, FP11 Accelerates computation by tuning compute performance
Tightly coupled high-bandwidth memory	>50TB/s on chip SRAM bandwidth, random access, reduces latency, minimizes external memory access
Programmable Data Path	Reduces unnecessary data movement, improving latency and efficiency
Configurability	Support for variable precision (trade-off throughput and accuracy). Future proof designs, and system connectivity

Deterministic Latency Matters for Inference

Automotive example:

- Latency impacts response time and distance
- Factors that impact latency – batch size / IO latency
- Need to perform better than human

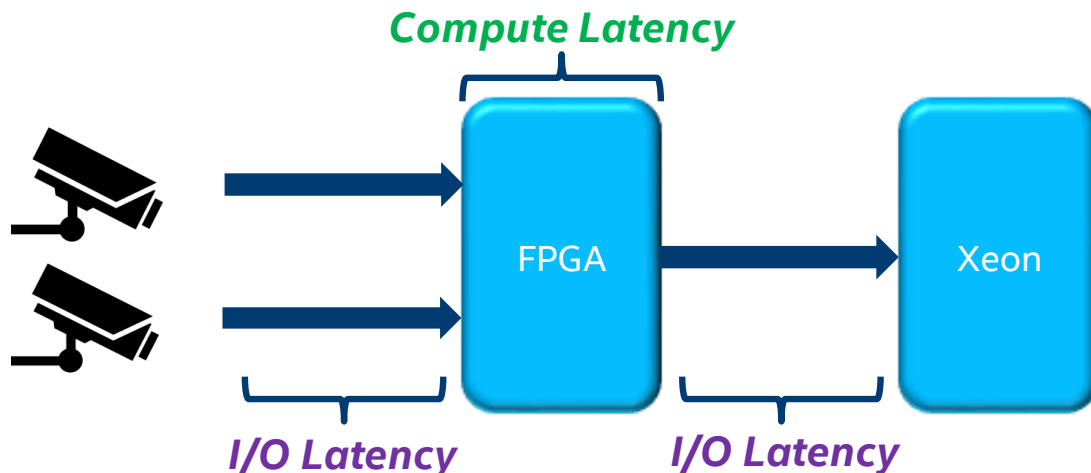


FPGAs Provide Deterministic System Latency

FPGAs leverages parallelism across the entire chip to reduce compute latency

FPGAs has flexible and customizable IOs with low & deterministic I/O latency

$$\text{System Latency} = \text{I/O Latency} + \text{Compute Latency}$$

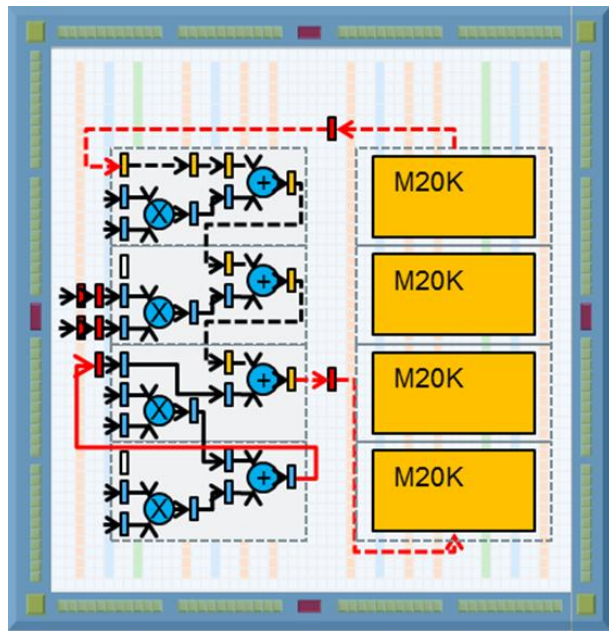
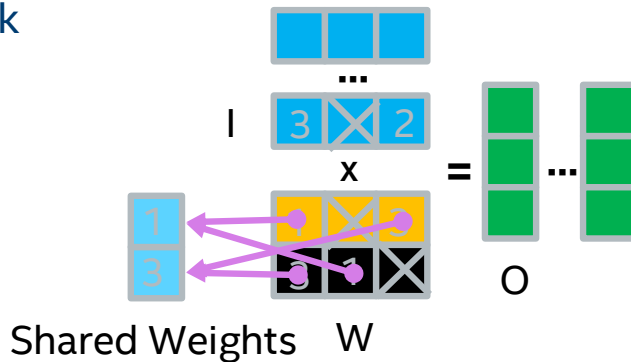
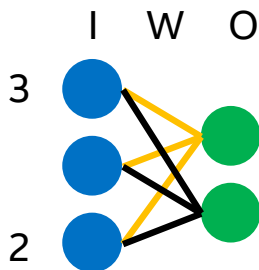


FPGA Flexibility Supports Arbitrary Architectures

Many efforts to improve efficiency in network development around limitations of GPU

- Batching
- Reduce bit width
- Sparse weights
- Sparse activations
- Weight sharing
- Compact network

LeNet [IEEE]	AlexNet [ILSVRC'12]	VGG [ILSVRC'14]	GoogLeNet [ILSVRC'14]	ResNet [ILSVRC'15]	XNORNet
BinaryConnect [NIPS'15]	TernaryConnect [ICLR'16]				
Spatially SparseCNN [CIFAR-10 winner '14]	Pruning [NIPS'15]	SparseCNN [CVPR'15]			
DeepComp [ICLR'16]	HashedNets [ICML'15]				
SqueezeNet					



CNN Inference Implementation Requirements

High throughput, feed forward data flow



Many floating point multiplies and accumulate operations

>8 TFLOP performance in Stratix 10



High bandwidth local storage for filter data and partial sums

>58 TB/s internal memory bandwidth in Stratix 10



Flexibility for different topologies and different problems



Summary

Deep Learning is a type of machine learning for extracting patterns from data using neural networks

DL neural networks are built and trained using frameworks and combining various layers

FPGAs are made up of a variety of building blocks that using FPGA development tools will translate code into custom hardware

FPGAs provide a flexible, deterministic low-latency, high-throughput, and energy-efficient solution for accelerating the constantly changing networks and precisions for DL inference

Legal Disclaimers/Acknowledgements

Features and benefits of Intel technologies depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. Check with your system manufacturer or retailer or learn more at www.intel.com.

Intel, the Intel logo, Intel Inside, the Intel Inside logo, MAX, Stratix, Cyclone, Arria, Quartus, HyperFlex, Intel Atom, Intel Xeon and Enpirion are trademarks of Intel Corporation or its subsidiaries in the U.S. and/or other countries.

OpenCL is the trademark of Apple Inc. used by permission by Khronos

*Other names and brands may be claimed as the property of others

© Intel Corporation

