OPENVINO ON FPGA

Maple Chou

FAE, Programmable Solutions Group

AI@IA HARDWARE

Multi-purpose to purpose-built
AI compute from cloud to device

MAINSTREAM

INTENSIVE

DEEP LEARNING

TRAINING

+

INFERENCE

+

MOST OTHER AI

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

ONE SIZE DOES NOT FIT ALL

OPENVINO™ TOOLKIT

Cross-Platform Tool to Accelerate Computer Vision & Deep Learning Inference Performance

Intel® Deep Learning Deployment Toolkit (DLDK)

Model Optimizer
Convert & Optimize

IR

Inference Engine
Optimized Inference

Code Samples & 10 Pre-trained Models

IR = Intermediate Representation file

Traditional Computer Vision Tools & Libraries

Optimized Libraries

OpenCV* | OpenVX* | Photography Vision

Code Samples

For Intel® CPU & CPU with integrated graphics

Increase Media/Video/Graphics Performance

Intel® Media SDK
Open Source version

OpenCL™ Drivers & Runtimes

For CPU with integrated graphics

Optimize Intel® FPGA

FPGA RunTime Environment
(from Intel® FPGA SDK for OpenCL™)

Bitstreams

FPGA – Linux* only

Intel® Architecture-Based Platforms Support

CELERON inside | ATOM inside | CORE inside | XEON inside | ARRIA 10 inside | MOVIDIUS inside | IRIS Pro GRAPHICS

OS Support CentOS* 7.4 (64 bit) Ubuntu* 16.04.3 LTS (64 bit) Microsoft Windows* 10 (64 bit) Yocto Project* version Poky Jethro v2.0.3 (64 bit)

All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

SOFTWARE.INTEL.COM/OPENVINO-TOOLKIT

DLDK

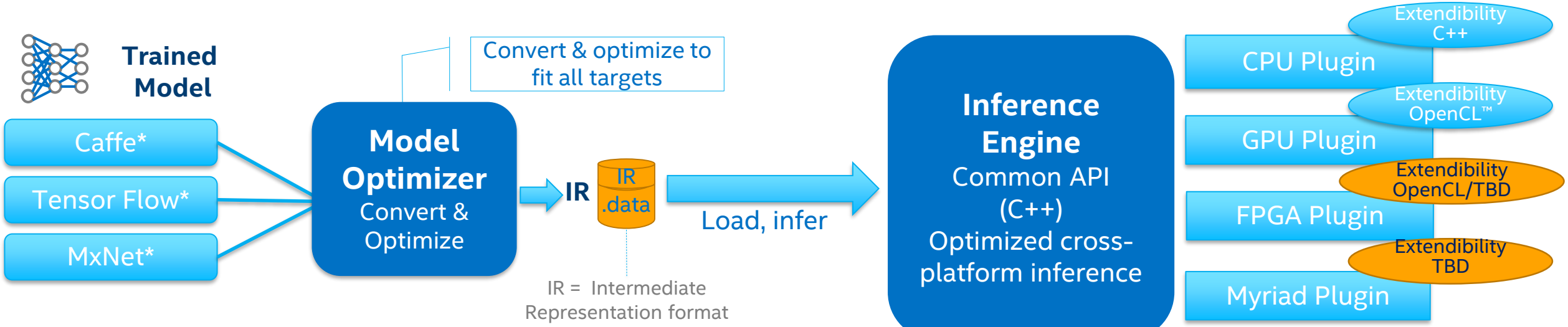
Take Full Advantage of the Power of Intel® Architecture for Deep Learning

Model Optimizer

- **What it is**: Preparation step -> imports trained models
- **Why important**: Optimizes for performance/space with conservative topology transformations; biggest boost is from conversion to data types matching hardware.

Inference Engine

- **What it is**: High-level inference API
- **Why important**: Interface is implemented as dynamically loaded plugins for each hardware type. Delivers best performance for each type without requiring users to implement and maintain multiple code pathways.

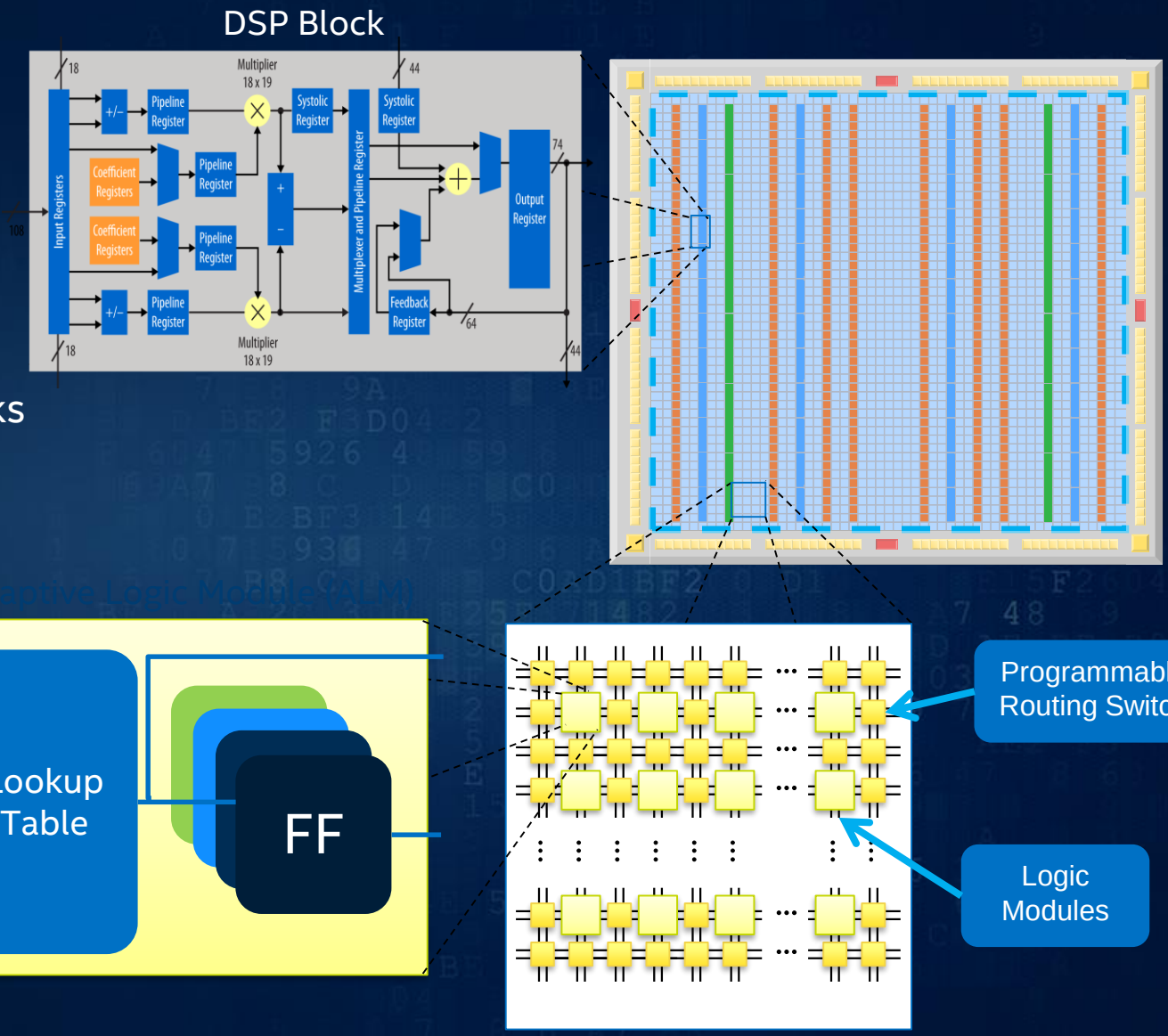


GPU = Intel CPU with integrated graphics processing unit/Intel® Processor Graphics

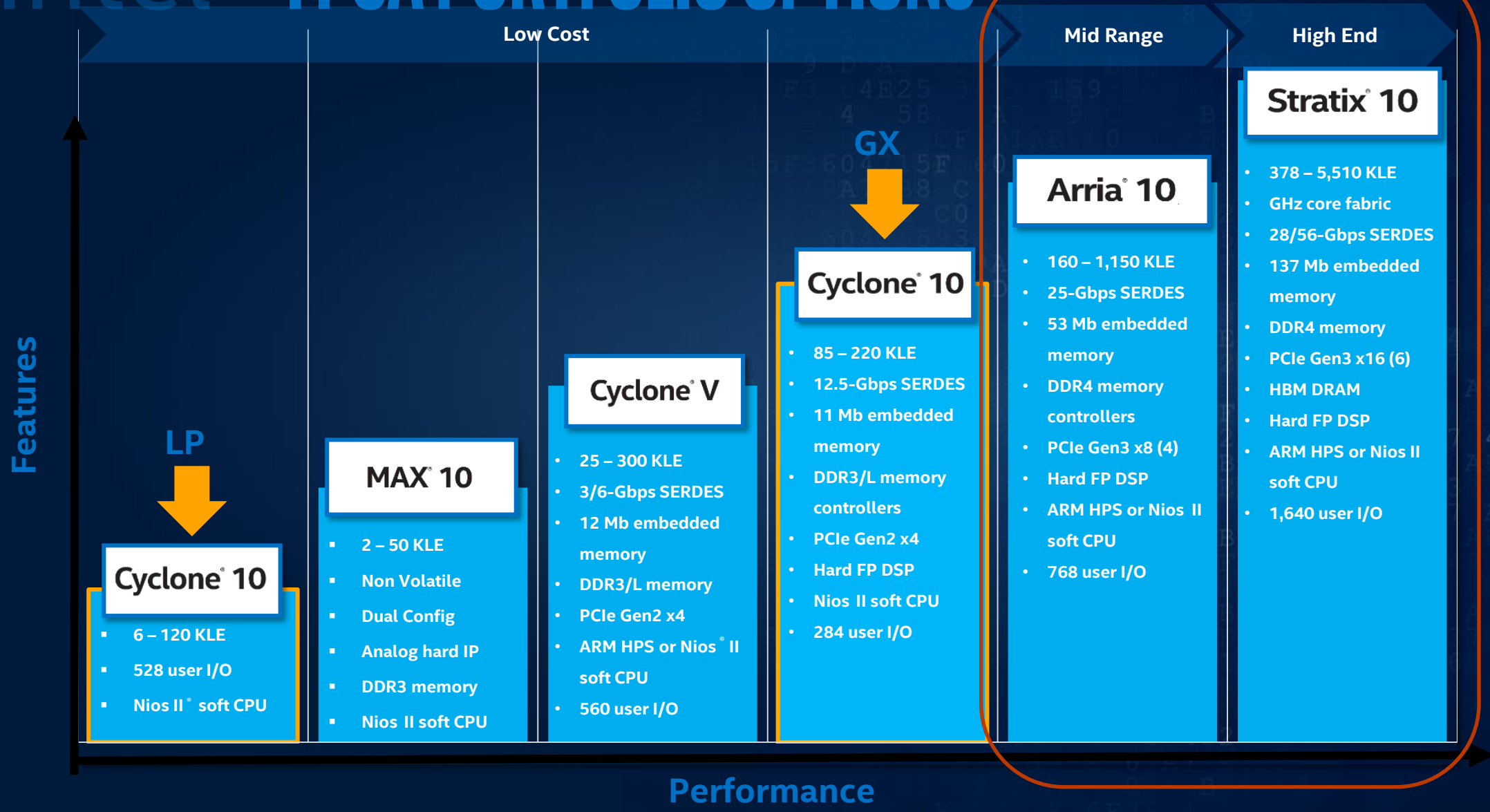
All products, computer systems, dates, and figures are preliminary based on current expectations, and are subject to change without notice.

FPGA ARCHITECTURE

- Massive Parallelism
  - Millions of logic elements
  - Thousands of embedded memory blocks
  - Thousands of Variable Precision DSP blocks
  - Programmable routing
  - Dozens of High-speed transceivers
  - Various built-in hardened IP
- FPGA Advantages
  - **Custom hardware!**
  - Efficient processing
  - Low power
  - Ability to reconfigure
  - Fast time-to-market

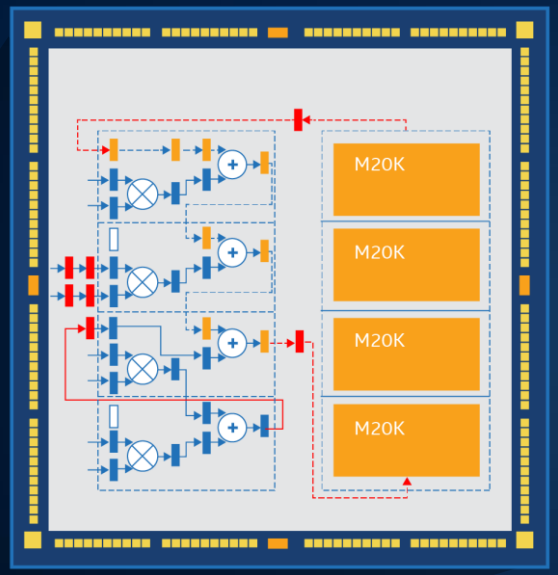


Intel® FPGA PORTFOLIO OPTIONS

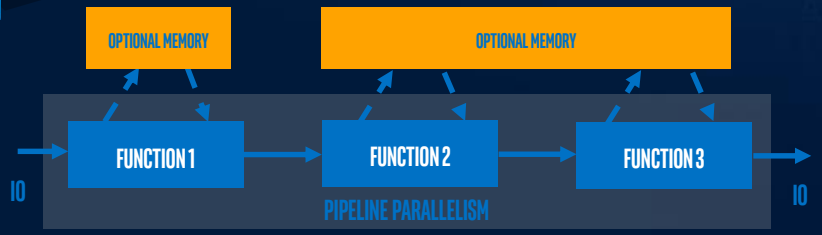


Why Intel® FPGAs for Machine Learning?

Convolutional Neural Networks are Compute Intensive



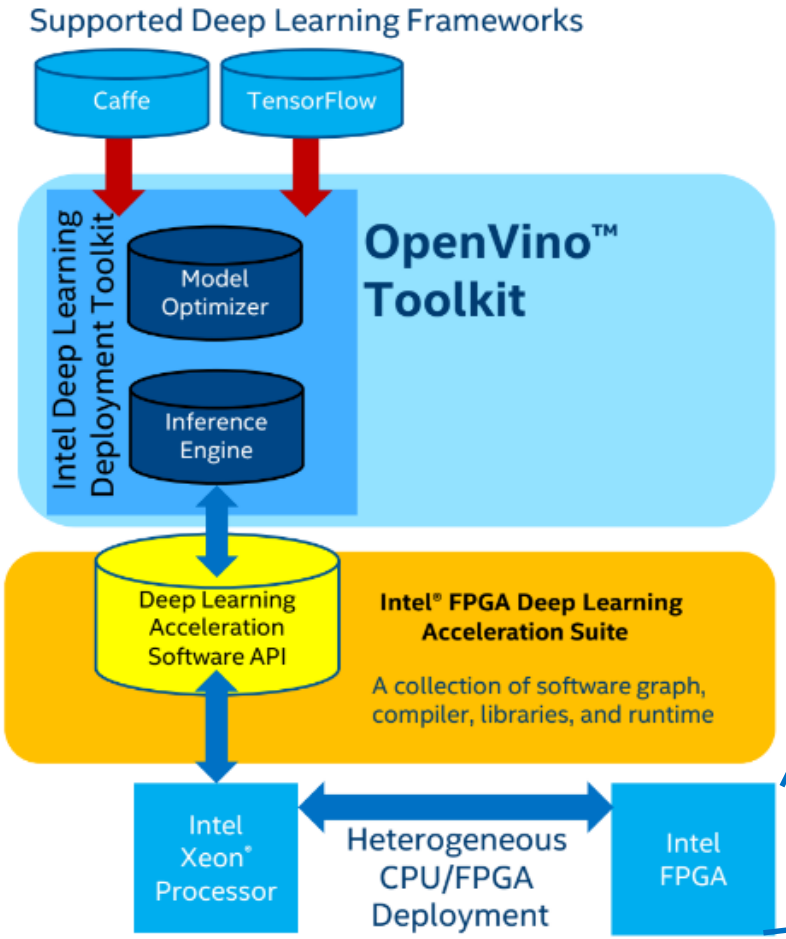
Fine-grained & low latency between compute and memory

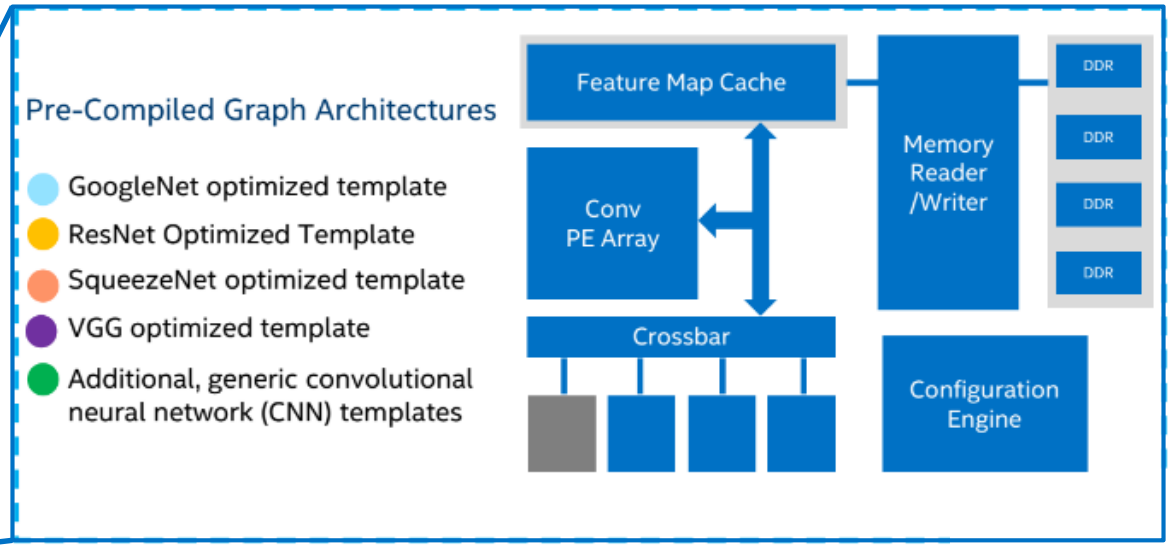


Feature	Benefit
Highly parallel architecture	Facilitates efficient low-batch video stream processing and reduces latency
Configurable Distributed Floating Point DSP Blocks	FP32 9Tflops, FP16, FP11 Accelerates computation by tuning compute performance
Tightly coupled high-bandwidth memory	>50TB/s on chip 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

INTEL® FPGA DLA SUITE

Enables transparent functional calling from high layer software to pre-compiled FPGA DL accelerators

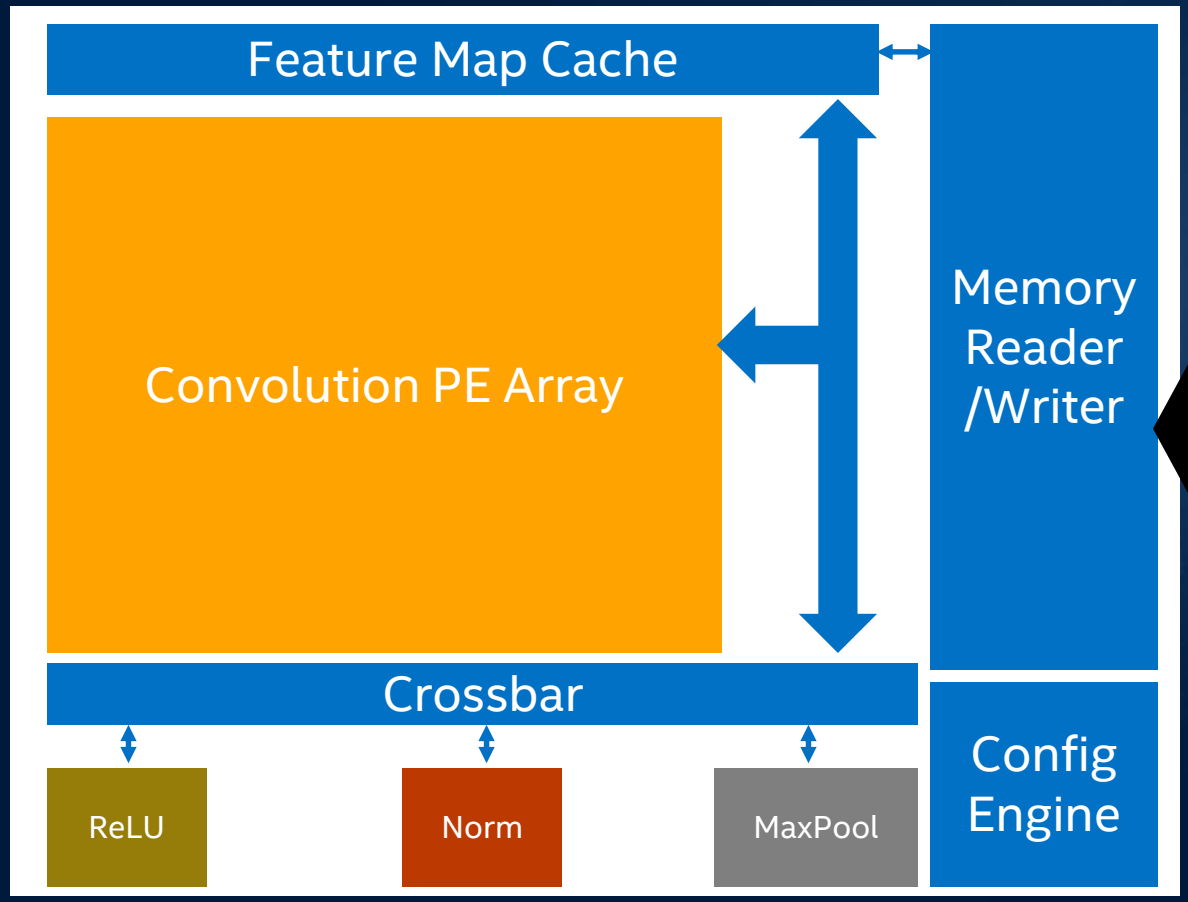




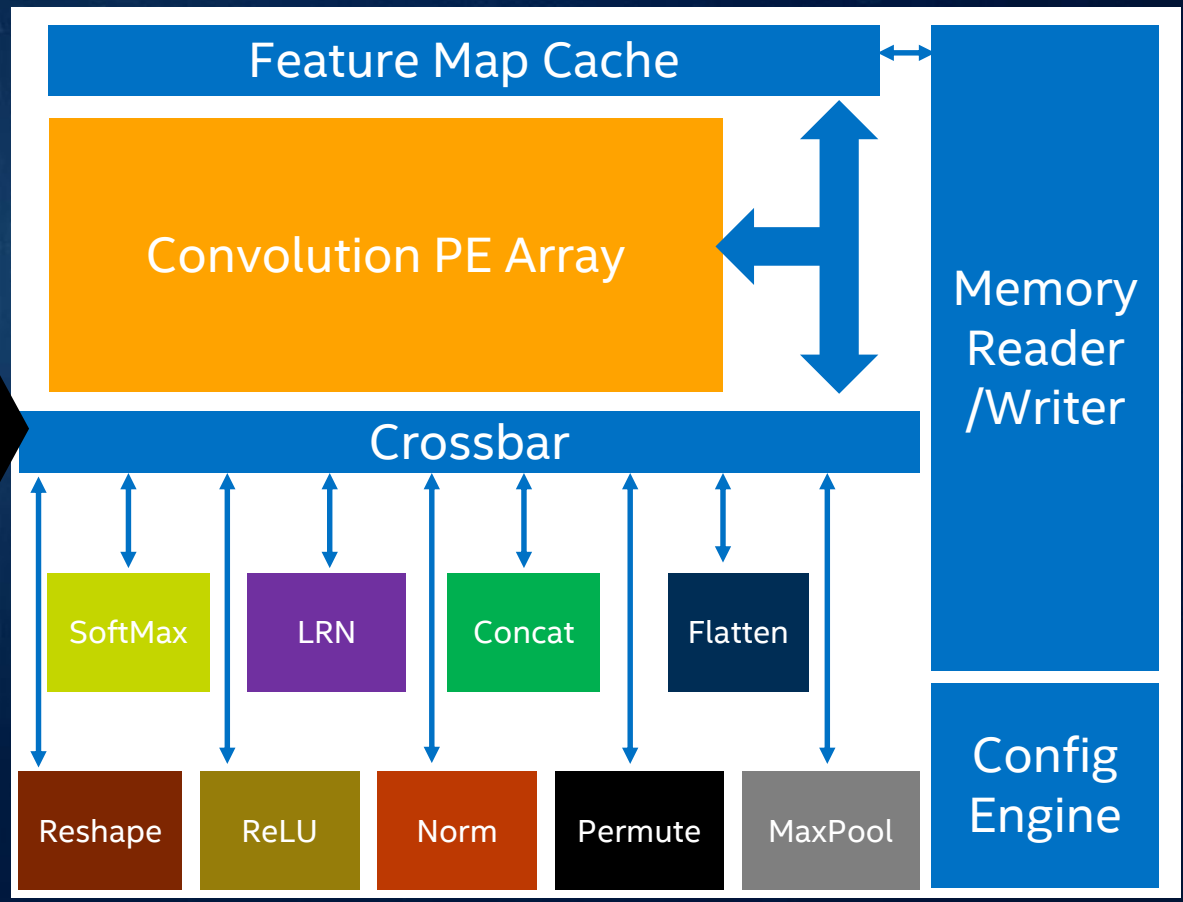
NO NEED OF FPGA EXPERIENCE AND KNOWLEDGE

SUPPORT FOR DIFFERENT TOPOLOGIES

Tradeoff between features and performance

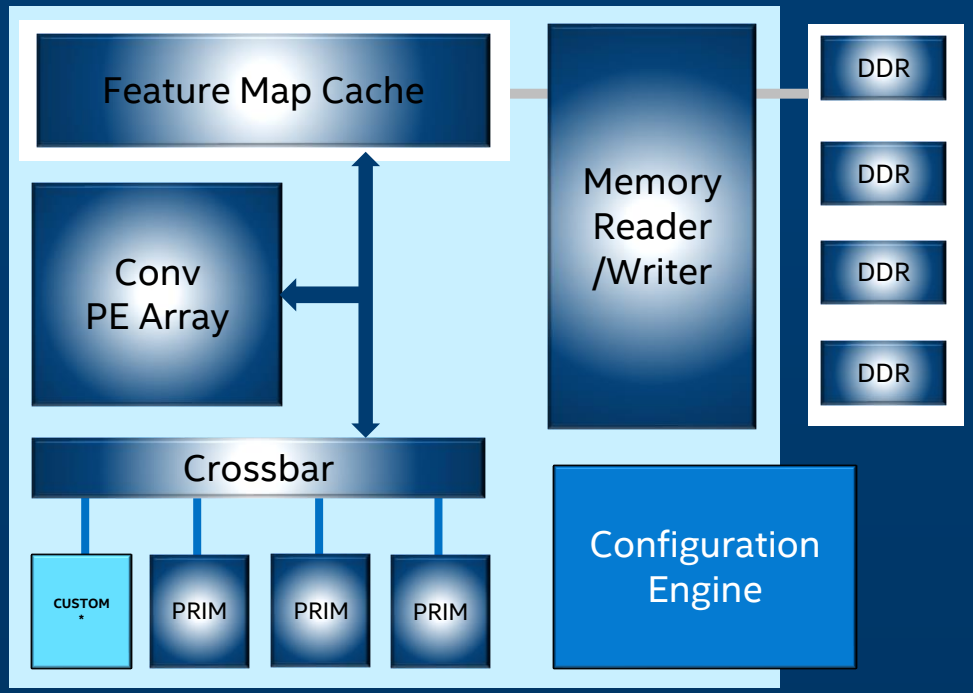


vs



INTEL® FPGA DEEP LEARNING ACCELERATION SUITE

PRE-COMPILED GRAPH ARCHITECTURE



***Deeper customization options COMING SOON!**

EXAMPLE TOPOLOGIES

AlexNet	GoogleNet	Tiny Yolo
VGG16	ResNet 18	SqueezeNet
ResNet 101	ResNet 50	SqueezeNet SDD
MobileNet	ResNetSSD	...*

***More topologies added with every release**

TARGET HW CARD

Intel® FPGA DLA Suite is compatible to Intel® programmable acceleration platforms & the OpenCL compiler for Intel® FPGA

Intel® PAC (Rush Creek)

Intel® Arria®10 GX Development Kit

Intel® HDDL-F (Pyramid Lake)

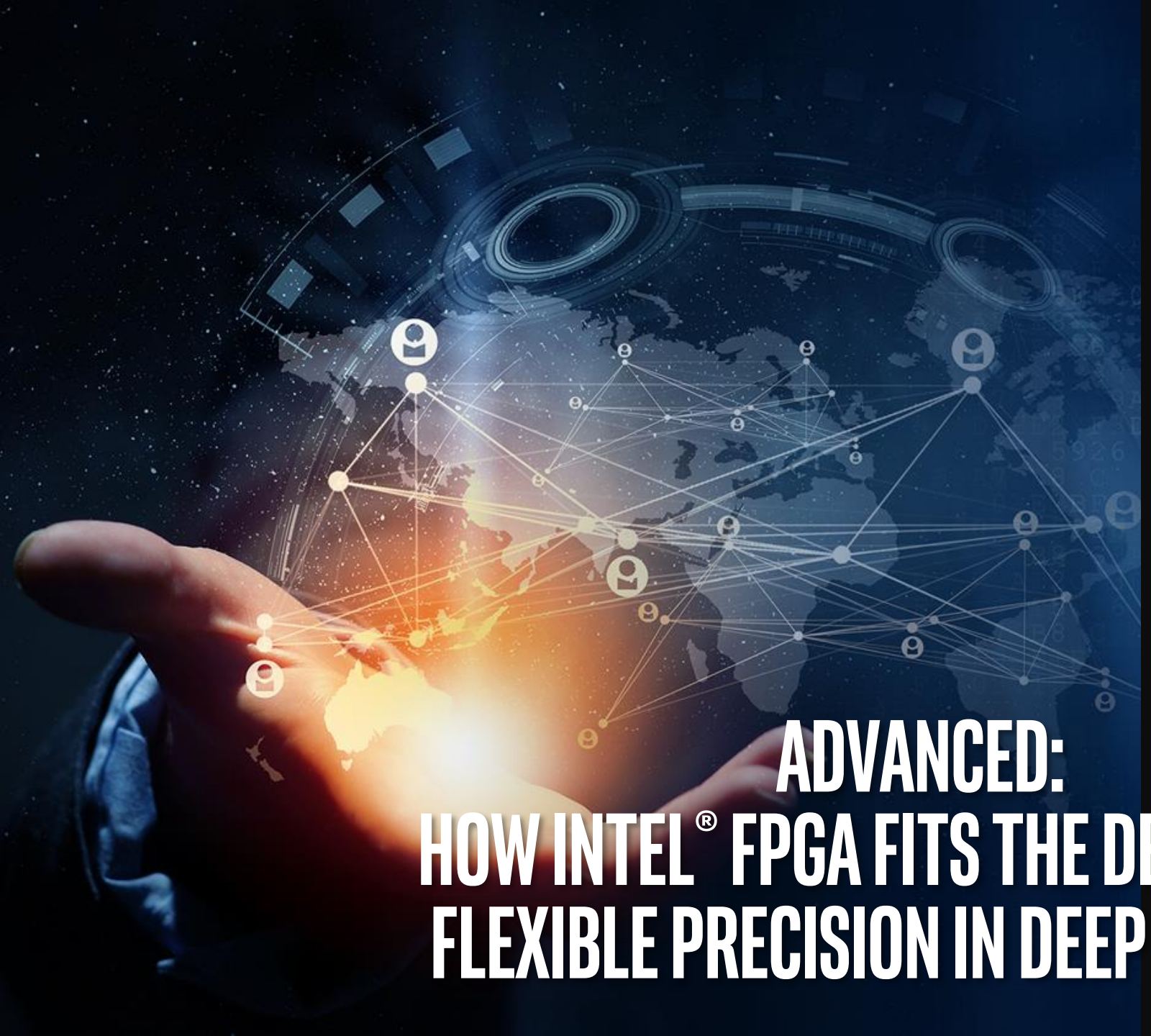
AVAILABLE NOW

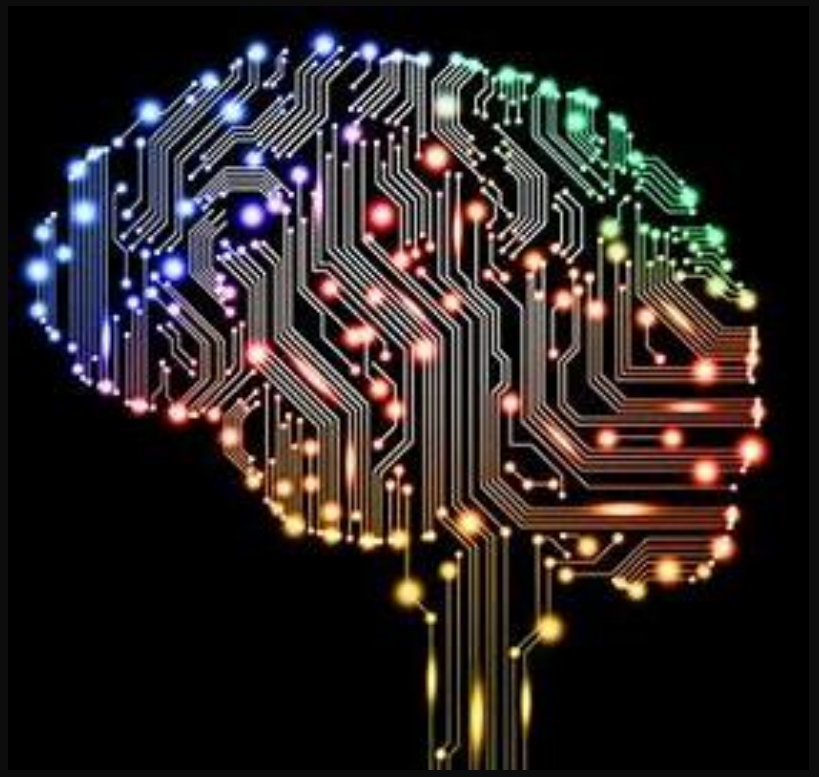
Darby Creek (S10SX)

S10MX, etc.

HDDL-FS (A10SX)

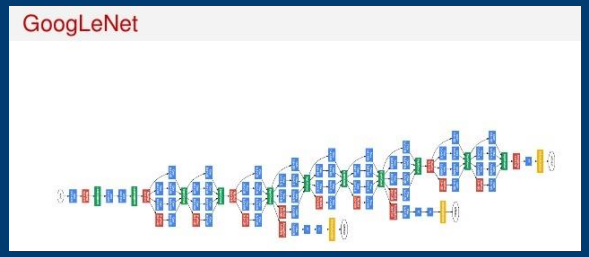
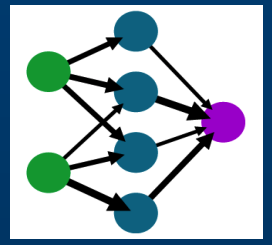
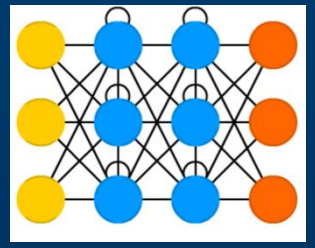
COMING SOON





ADVANCED:
HOW INTEL® FPGA FITS THE DEMANDS OF FLEXIBLE PRECISION IN DEEP LEARNING

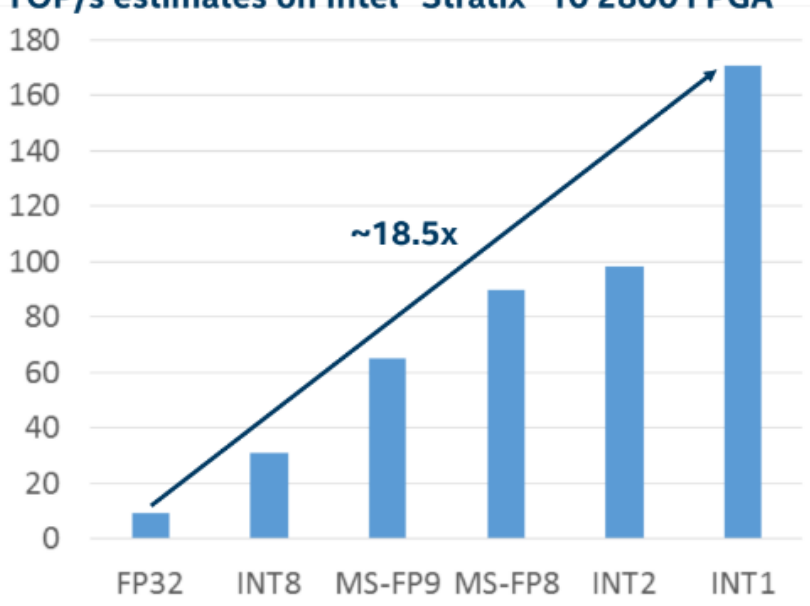
EVOLVING DEEP LEARNING REQUIREMENTS

2017	2018
	
Convolutional Neural Network (CNN)	Recurrent Neural Network (RNN)
	
Floating Point	Floating Point
FP32	FP16 FP11 FP9 BFLOAT

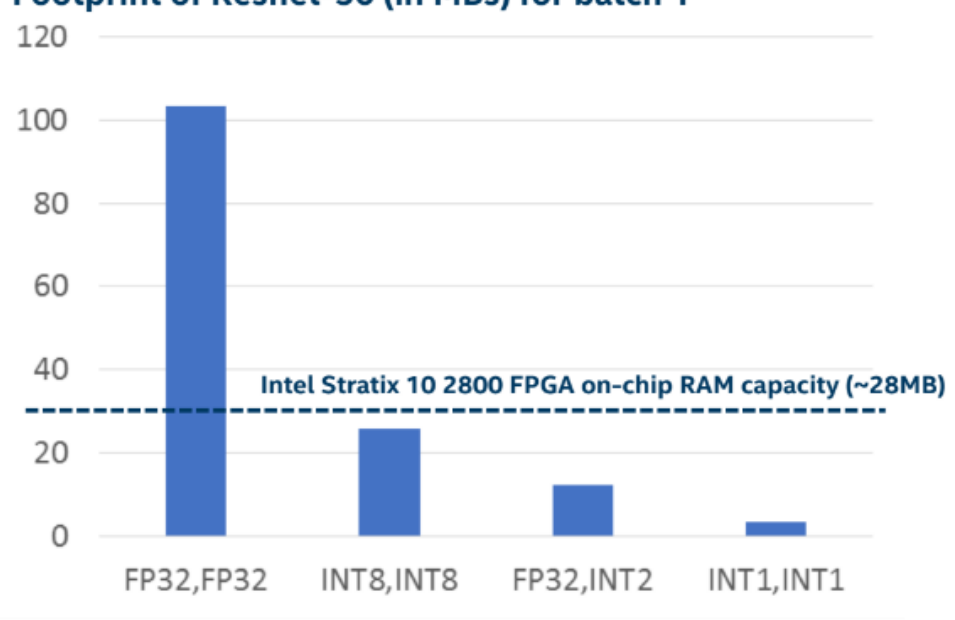
PEAK PERFORMANCE

With FPGA's flexibility in data processing precision, it offers extreme high performance and efficiency

Improved Throughput -> Custom data types



Smaller foot print → "persistent" DNNs



FPGAs are great for custom low precisions (e.g., MS-FP9, INT2, INT1)

Softwa… are me… consult… products. For more complete information visit http://www.intel.com/performance.

HOW LOW BIT YOU CAN GO

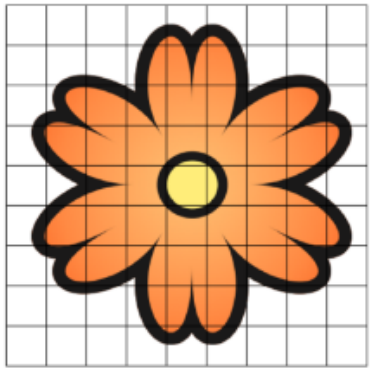
Intel researching on WRPN

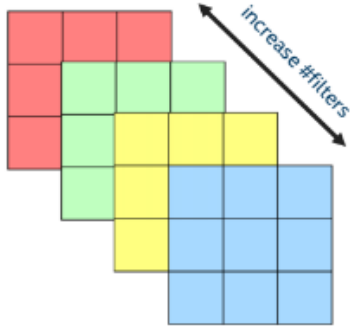
On Convolutional Neural Networks (CNNs) for images (ImageNet)

Wide Reduced-Precision Networks (WRPN)

https://arxiv.org/pdf/1704.03079.pdf

- Helps recover low precision classification accuracy loss
- Widen network by increasing the number of filters





3 years ago: 32bit training and ~16 bit inference were the norm.
Now: 16bit training and 8bit inference + promising evidence on sub 8bit.

EVOLVING PRECISION FOR AI

- Intel FPGAs enable exploration of precision, topology and accuracy tradeoffs
- Example of gaining 4X performance with the same FPGA while maintaining accuracy

		ResNet-34 1x Wide		ResNet-34 2x Wide		ResNet-34 3x Wide	
Activation	Weight	Eq TOPS	Top-1 Acc	Eq TOPS	Top-1 Acc	Eq TOPS	Top-1 Acc
FP32	FP32	**7**	**0.7359**	NR	NR	NR	NR
8-bit	8-bit	8	0.7093	2	NR	1	NR
8-bit	Ternary	43	0.6919	11	NR	5	NR
8-bit	Binary	52	NR	13	NR	6	NR
4-bit	4-bit	18	0.7033	5	0.7453	2	NR
3-bit	3-bit	51	NR	13		6	NR
2-bit	2-bit	85	0.6793	21	0.7332	9	NR
2-bit	Ternary	98	0.6793	**25**	**0.7332**	11	NR
1-bit	1-bit	**267**	**0.6054**	67	0.6985	**30**	**0.7238**

Throughput and Accuracy for various PE configurations on ResNet Topologies

EVOLVING TO MEMORY BOUND WORKLOADS

- Intel® FPGAs are estimated to accelerate DeepSpeech by greater than 6.5X compared to the P4 GPU with an RNN optimized core
- Intel® Stratix 10 MX can further reduce latency by directly ingesting the speech signal
- Intel Stratix 10 MX offers 512GBps bandwidth via multiple independent HBM channels

Stream Length	P4 (measured) (32 bit)	Intel Stratix® 10 MX (estimated*) (16 bit)	Intel® Stratix® 10 MX (estimated*) (8 bit)
1s	0.3s	0.047s	24.1ms
10s	5.22s	0.464s	226.8ms
20s	6s	0.928s	452.1ms
40s	11.76s	1.855s	902.6ms

Mozilla DeepSpeech topology implementation

*Estimations performed by Manjeera Design Systems
Assumption: ~4.4 TOPs of 16b compute (8192 MACs at 266MHz) for Intel Stratix 10 MX

USE CASE 1: SEARCH

Solution Search

Looking for a quick path to deploy and accelerate instant reverse image searches of products for retail convenience

Solution Success

Intel FPGAs offered real-time AI inferencing using OpenVINO Toolkit. This enabled engineers to map neural networks to FPGA, accelerating image searches with increased throughput and lower latency, all without the need for FPGA programming experience

OPENVINO TOOLKIT

Accelerating workloads, enabling deep learning capabilities for smarter and faster ways to transform data for competitive edge

ACCELERATION STACK FOR INTEL® XEON® CPU WITH FPGAS

Abstracting programming complexity and maximizing ease of use by hot-swapping accelerators and enabling application portability for Intel FPGA based acceleration solutions

INTEL PROGRAMMABLE ACCELERATION CARD WITH INTEL® ARRIA® 10

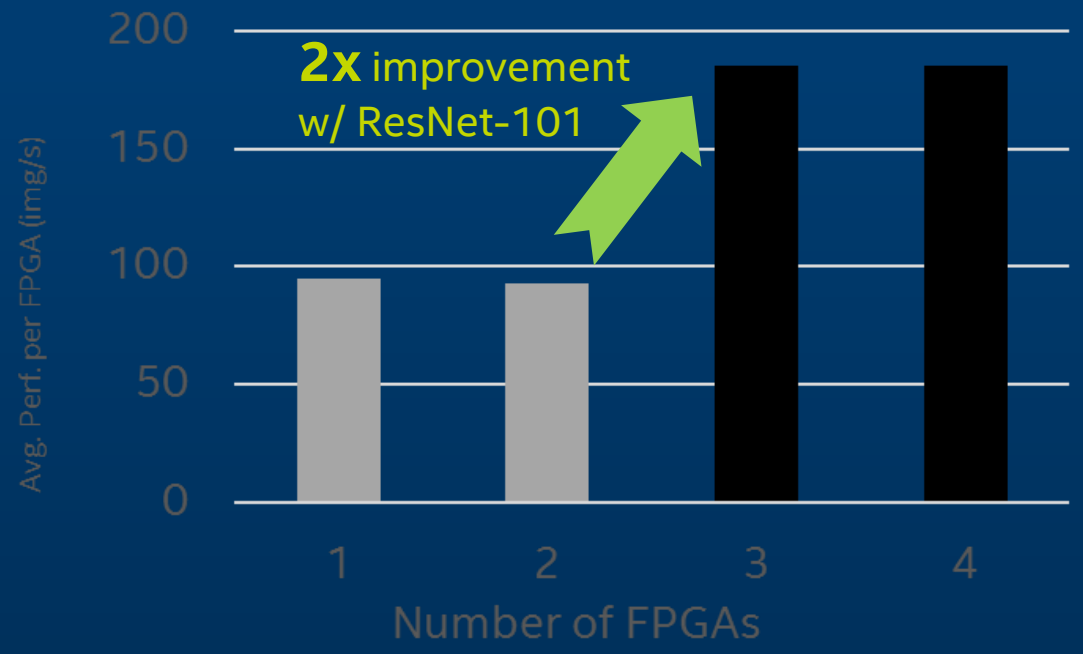
Deployment ready PCIe-based card with versatile built-in multifunction acceleration capabilities with low-power dissipation and low-profile form factor

REAL-TIME AI OPTIMIZED FOR PERFORMANCE, POWER AND COST

USE CASE 2: MICROSOFT'S AI FOR EARTH

MSFT leverages the multimode capabilities of Intel FPGAs to push through the memory wall to maximize performance

Per-chip performance **increases** when scaled



Project Brainwave with Intel® Stratix® 10 gives Performance/$ → only $42 of compute*

*Microsoft's Blog

200M Images, 20TB

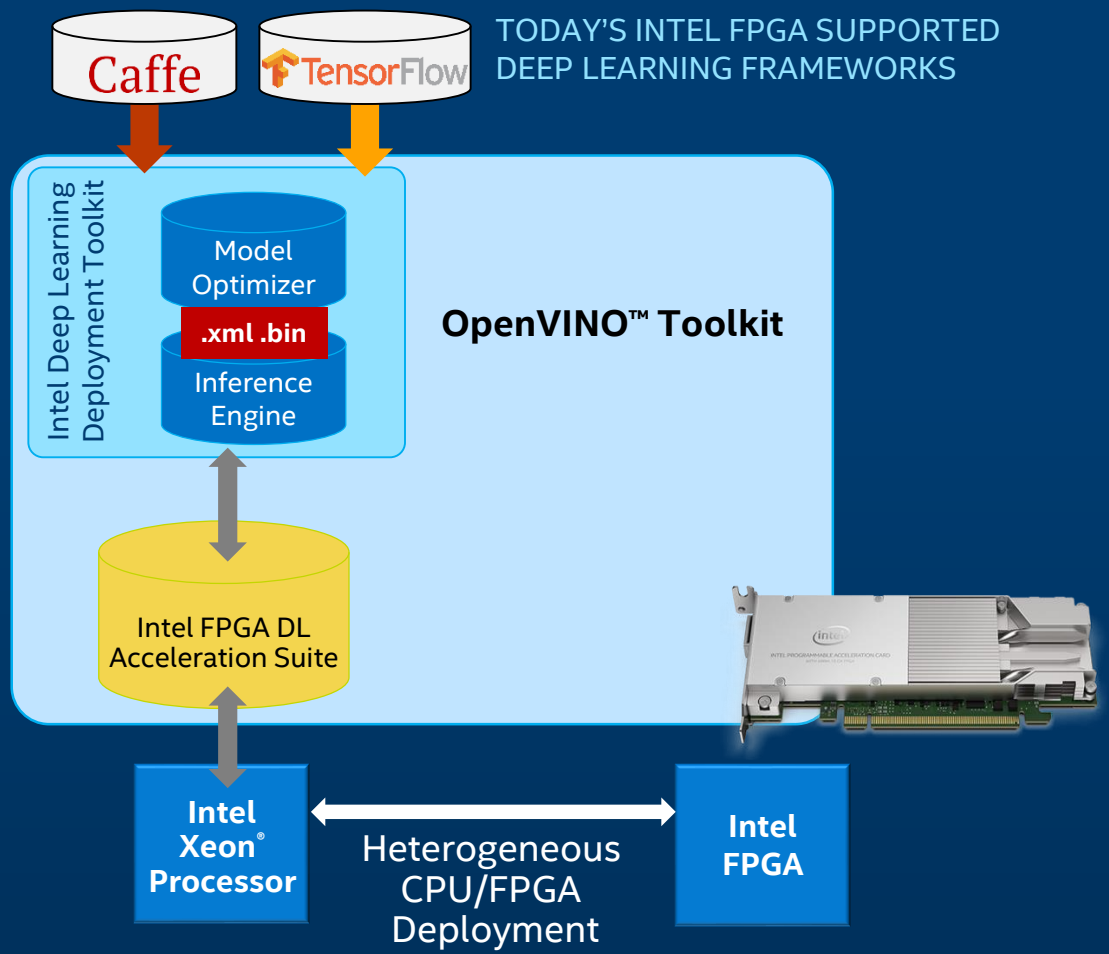
Land cover mapping for the whole US

10+ minutes

OPENVINO ON FPGA

The detail of OpenVINO

OPENVINO TOOLKIT FOR INTEL FPGAS



AN ALL-IN-ONE SOLUTION TO EASILY HARNESS THE BENEFITS OF FPGAS

- Enables developers and data scientists to take their prototype application to production
- Drives power, cost and development efficiencies
- Utilize API-based & direct coding to maximize performance
- Deeper customization capabilities coming soon

Free Download
software.intel.com/openvino-toolkit

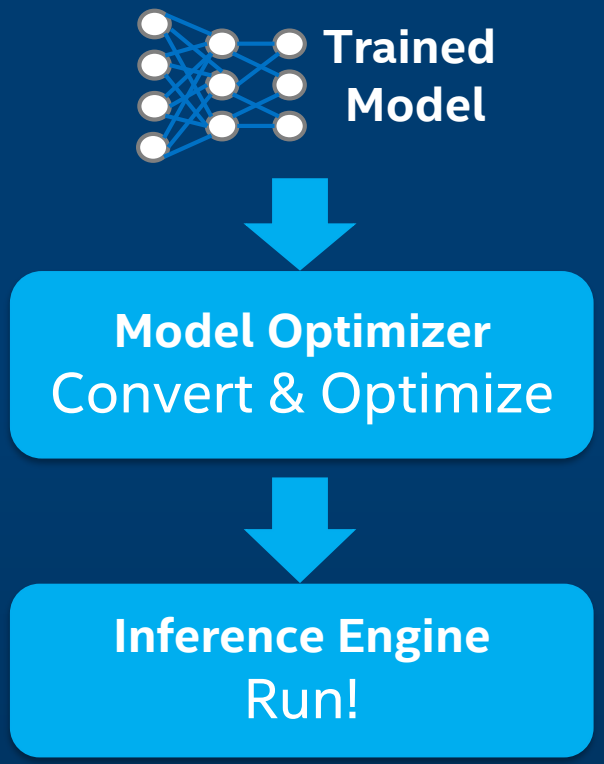
DEEP LEARNING DEPLOYMENT TOOLKIT

Model Optimizer

- Imports trained models from popular deep learning frameworks regardless of training hardware
- Enhances model for improved execution, storage & transmission

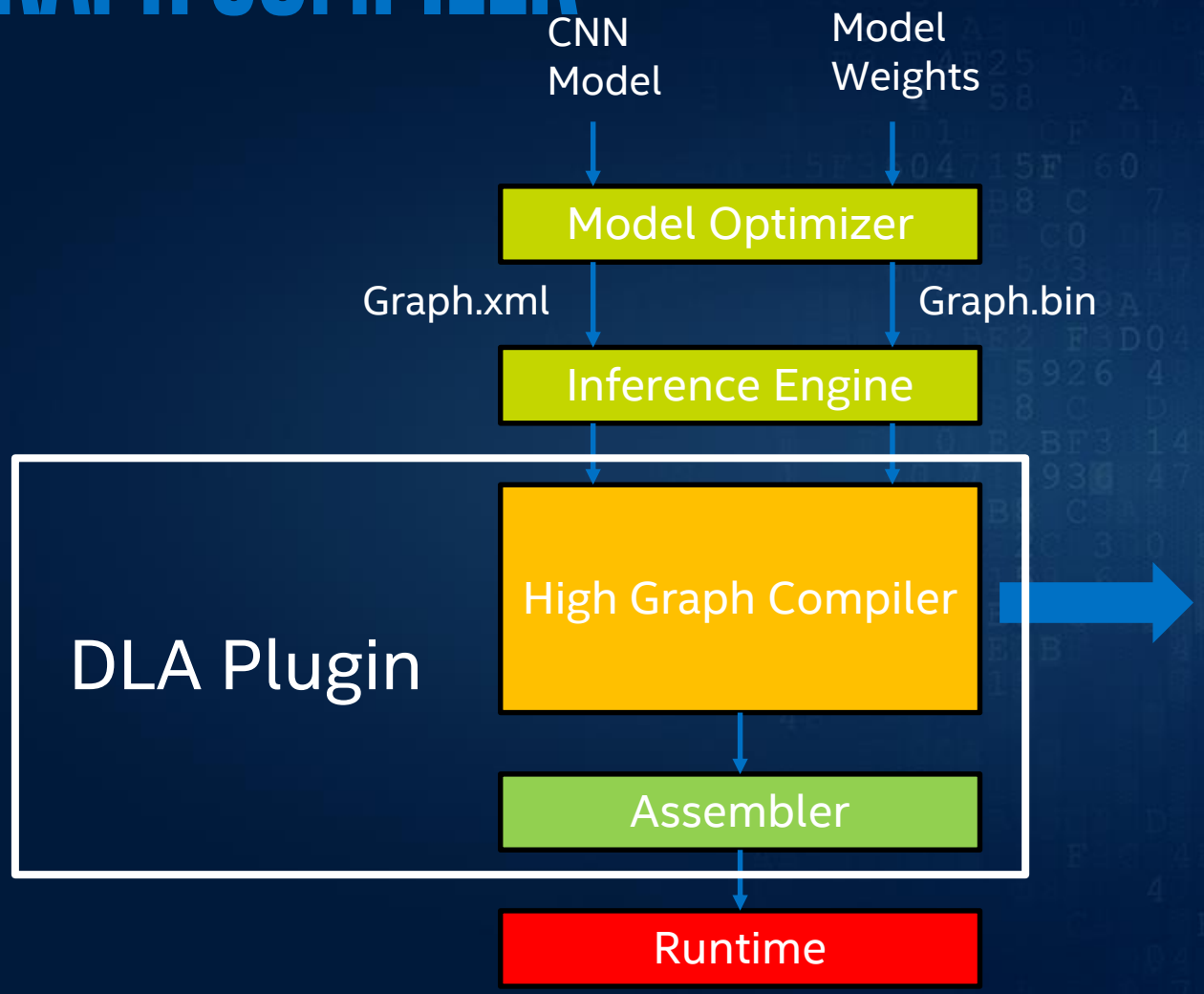
Inference Engine

- Optimizes Inference execution for target hardware (computational graph analysis, scheduling, model compression, quantization)
- Enables seamless integration with application logic
- Delivers embedded friendly Inference solution



Ease of use + Embedded friendly + Extra performance boost

HIGH GRAPH COMPILER



Compiles a graph into a format that can be handled by DLA

Contains graph analysis and transformation passes

Analysis:

- Slice Analysis
- Scheduling
- Addressing
- Slice Offsets

Transformation Passes:

- Eltwise Pass
- Eltwise Conv Merging Pass
- Slice Pass
- Destride Pass
- Global Average Pool Pass
- Pool Concat Pass
- Constant Propagation
- Fusion Pass
- FC to Convolution Pass
- Identity Insertion Passes
- Etc.

APPLYING DEVICE AFFINITIES TO LAYERS: AUTOMATICALLY, USING THE FALLBACK *POLICY*, 1

$ object_detection_sample_ssd **-d HETERO:FPGA,CPU**
-m ssd.xml -i snake.bmp

All IE samples support that

You can load CPU and GPU extensions as usual (“-l” and “-c”)

Regular “-pc” (perf counters) works and gives nice per-subgraph statistics

The “priorities” just defines a greedy behavior

- Keeps all layers that can be executed on the device (FPGA)
- Carefully respecting the topological and other limitations

APPLYING DEVICE AFFINITIES TO LAYERS: AUTOMATICALLY, USING THE FALLBACK *POLICY*, 2

```
HeteroPluginPtr plugin(make_plugin_name("HeteroPlugin"));
CNNNetReader reader;
reader.ReadNetwork(“Model.xml”);
reader.ReadWeights(“Model.bin”);
CNNNetwork network = reader.getNetwork();

plugin->SetConfig({ { "TARGET_FALLBACK", “FPGA,CPU"} });
plugin->LoadNetwork(exeNetwork, network,{}, &response);
```

APPLYING DEVICE AFFINITIES TO LAYERS: EXPLICIT, USING THE API

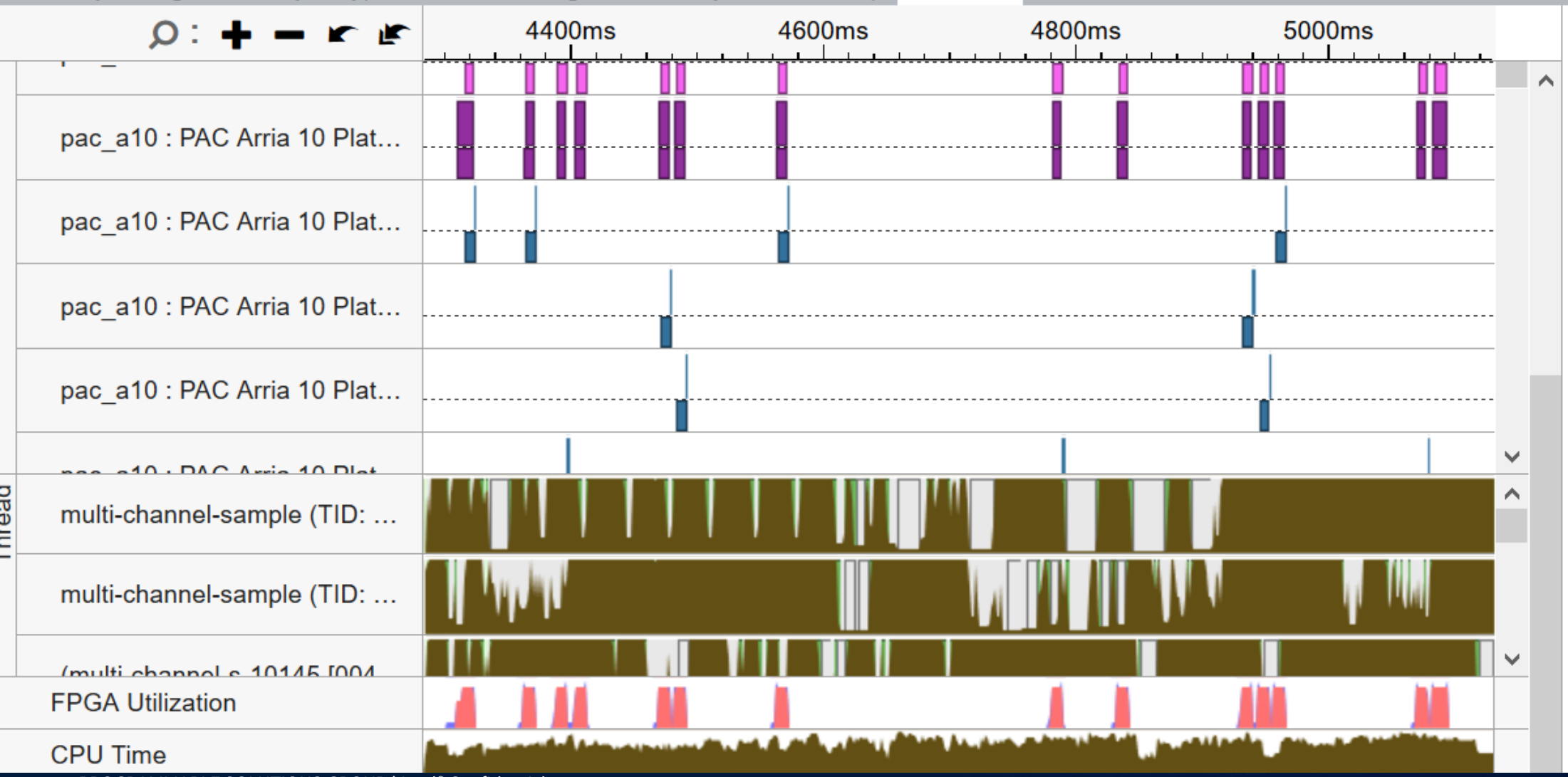
```
HeteroPluginPtr plugin(make_plugin_name("HeteroPlugin"));
CNNNetReader reader;
reader.ReadNetwork(“Model.xml”);
reader.ReadWeights(“Model.bin”);
CNNNetwork network = reader.getNetwork();
plugin->SetConfig({ { "TARGET_FALLBACK", “FPGA,CPU"} }, &response);

plugin->SetAffinity(network,{}, &response);

auto network = netBuilder.getNetwork();
        auto it = network.begin();
        while (it != network.end()) {
            CNNLayer::Ptr layer = *it++;
            layer->affinity = “FPGA";
            if (layer->name == “conv1") || layer->kernel_size >= 15) {
                 layer->affinity = “CPU";
            }
        }
 status = plugin_ptr->LoadNetwork(ie_net.getNetwork(), &dsc);
```

LIVE DEMO / VIDEO

```
[root@localhost ~]# aocl program acl0 /opt/intel/computer_vision_sdk/a10_dcp_bitstreams/2-0-1_RC_FP11_ResNet50-101.aocx
aocl program: Running program from /root/inteldevstack/a10_gx_pac_ias_1_1_pv/opencl/opencl_bsp/linux64/libexec
Program succeed.
[root@localhost ~]# LD_PRELOAD=/root/inteldevstack/a10_gx_pac_ias_1_1_pv/opencl/opencl_bsp/linux64/lib/libintel_opae_mmd.so /opt/intel/computer_vision_sdk/deployment_tools/inference_engine/samples/build/intel64/
Release/interactive_face_detection_sample -m=/opt/intel/computer_vision_sdk/deployment_tools/intel_models/face-detection-retail-0004/FP32/face-detection-retail-0004.xml -m_ag=/opt/intel/computer_vision_sdk/deplo
yment_tools/intel_models/age-gender-recognition-retail-0013/FP32/age-gender-recognition-retail-0013.xml -m_hp=/opt/intel/computer_vision_sdk/deployment_tools/intel_models/head-pose-estimation-adas-0001/FP32/head
-pose-estimation-adas-0001.xml -m_em=/opt/intel/computer_vision_sdk/deployment_tools/intel_models/emotions-recognition-retail-0003/FP32/emotions-recognition-retail-0003.xml -i /root/Videos/obama.mp4 -d HETERO:FP
GA,CPU -d_ag HETERO:FPGA,CPU -d_em HETERO:FPGA,CPU -d_hp HETERO:FPGA,CPU -pc -n_ag=6 -n_em=6
Inference Engine:
        API version ............ 1.2
        Build .................. 13911
[ INFO ] Parsing input parameters
[ INFO ] Reading input
[ INFO ] Loading plugin HETERO:FPGA,CPU

        API version ............ 1.2
        Build .................. heteroPlugin
        Description ....... heteroPlugin
[ INFO ] Loading network files for Face Detection
[ INFO ] Batch size is set to 1
[ INFO ] Checking Face Detection inputs
[ INFO ] Checking Face Detection outputs
[ INFO ] Loading Face Detection model to the HETERO:FPGA,CPU plugin
[ INFO ] Loading network files for AgeGender
[ INFO ] Batch size is set to 6 for Age Gender
[ INFO ] Checking Age Gender inputs
[ INFO ] Checking Age Gender outputs
[ INFO ] Age layer: age_conv3
[ INFO ] Gender layer: prob
[ INFO ] Loading Age Gender model to the HETERO:FPGA,CPU plugin
[ INFO ] Loading network files for Head Pose detection
[ INFO ] Batch size is set to  16 for Head Pose Network
[ INFO ] Checking Head Pose Network inputs
[ INFO ] Checking Head Pose network outputs
[ INFO ] Loading Head Pose model to the HETERO:FPGA,CPU plugin
[ INFO ] Loading network files for Emotions recognition
[ INFO ] Batch size is set to 6 for Emotions recognition
[ INFO ] Checking Emotions Recognition inputs
```

```
subgraph1: 2. input transf... EXECUTED      layerType:                  realTime: 299       cpu: 0          execType:
subgraph1: 3. FPGA execute... EXECUTED      layerType:                  realTime: 2103      cpu: 0          execType:
subgraph1: 4. output trans... EXECUTED      layerType:                  realTime: 62        cpu: 0          execType:
subgraph1: 5. FPGA output ... EXECUTED      layerType:                  realTime: 23        cpu: 23         execType:
subgraph1: 6. softmax/copy    EXECUTED      layerType:                  realTime: 27        cpu: 27         execType:
subgraph2: Scale1/Mul_/Fus... NOT_RUN       layerType: Input            realTime: 0         cpu: 0          execType: unknown
subgraph2: detection_out      EXECUTED      layerType: DetectionOutput  realTime: 453       cpu: 453        execType: unknown
subgraph2: fc7_mbox_conf      NOT_RUN       layerType: Input            realTime: 0         cpu: 0          execType: unknown
subgraph2: fc7_mbox_conf_flat NOT_RUN       layerType: Flatten          realTime: 0         cpu: 0          execType: unknown
subgraph2: fc7_mbox_conf_perm EXECUTED      layerType: Permute          realTime: 39        cpu: 39         execType: unknown
subgraph2: fc7_mbox_loc       NOT_RUN       layerType: Input            realTime: 0         cpu: 0          execType: unknown
subgraph2: fc7_mbox_loc_flat  NOT_RUN       layerType: Flatten          realTime: 0         cpu: 0          execType: unknown
subgraph2: fc7_mbox_loc_perm  EXECUTED      layerType: Permute          realTime: 62        cpu: 62         execType: unknown
subgraph2: fc7_mbox_priorbox  NOT_RUN       layerType: PriorBoxClustered realTime: 0        cpu: 0          execType: unknown
subgraph2: mbox_conf_flatten  NOT_RUN       layerType: Flatten          realTime: 0         cpu: 0          execType: unknown
subgraph2: mbox_conf_reshape  NOT_RUN       layerType: Reshape          realTime: 0         cpu: 0          execType: unknown
subgraph2: mbox_conf_softmax  EXECUTED      layerType: SoftMax          realTime: 100       cpu: 100        execType: ref_any
subgraph2: out_detection_out  NOT_RUN       layerType: Output           realTime: 0         cpu: 0          execType: unknown
Total time: 3646     microseconds
[ INFO ] Performance counts for Age Gender

subgraph1: 1. input prepro... EXECUTED      layerType:                  realTime: 722       cpu: 722        execType:
subgraph1: 2. input transf... EXECUTED      layerType:                  realTime: 408       cpu: 0          execType:
subgraph1: 3. FPGA execute... EXECUTED      layerType:                  realTime: 2812      cpu: 0          execType:
subgraph1: 4. output trans... EXECUTED      layerType:                  realTime: 27        cpu: 0          execType:
subgraph1: 5. FPGA output ... EXECUTED      layerType:                  realTime: 1         cpu: 1          execType:
subgraph1: 6. softmax/copy    EXECUTED      layerType:                  realTime: 17        cpu: 17         execType:
subgraph2: out_prob           NOT_RUN       layerType: Output           realTime: 0         cpu: 0          execType: unknown
subgraph2: prob               EXECUTED      layerType: SoftMax          realTime: 6         cpu: 6          execType: ref_any
Total time: 3993     microseconds
[ INFO ] Performance counts for Head Pose

subgraph1: 1. input prepro... EXECUTED      layerType:                  realTime: 918       cpu: 918        execType:
subgraph1: 2. input transf... EXECUTED      layerType:                  realTime: 592       cpu: 0          execType:
subgraph1: 3. FPGA execute... EXECUTED      layerType:                  realTime: 6561      cpu: 0          execType:
subgraph1: 4. output trans... EXECUTED      layerType:                  realTime: 62        cpu: 0          execType:
subgraph1: 5. FPGA output ... EXECUTED      layerType:                  realTime: 9         cpu: 9          execType:
subgraph1: 6. softmax/copy    EXECUTED      layerType:                  realTime: 57        cpu: 57         execType:
Total time: 8199     microseconds
```

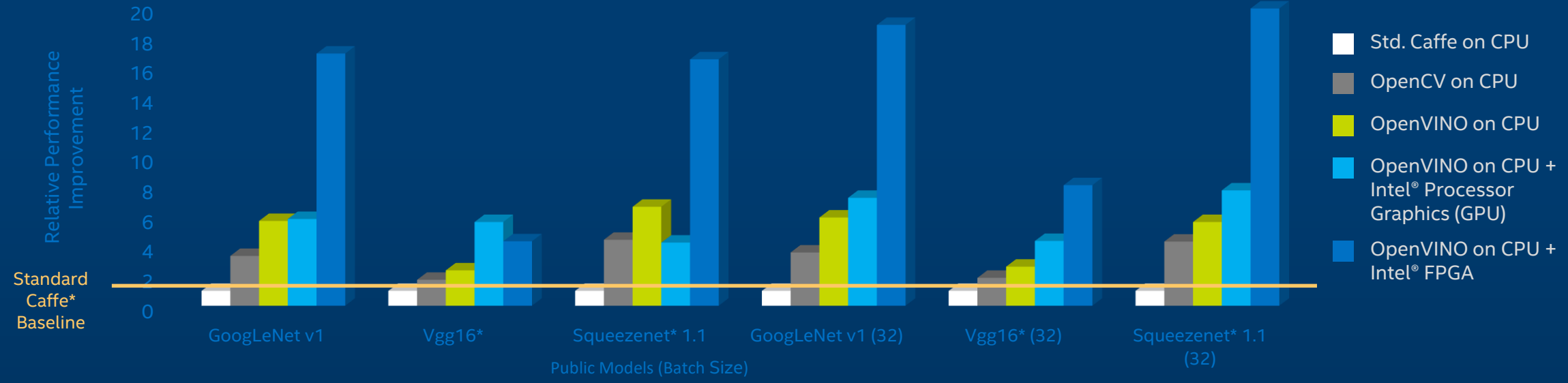
VTune, FPGA is coming!



CPU + FPGA ACCELERATE AI APPLICATIONS

COMPARISON OF FRAMES PER SECOND (FPS)

19.9x[1]



GET AN EVEN BIGGER PERFORMANCE BOOST WITH INTEL® FPGA

[1]Depending on workload, quality/resolution for FP16 may be marginally impacted. A performance/quality tradeoff from FP32 to FP16 can affect accuracy; customers are encouraged to experiment to find what works best for their situation. Performance results are based on testing as of June 13, 2018 and may not reflect all publicly available security updates. See configuration disclosure for details. No product can be absolutely secure. For more complete information about performance and benchmark results, visit www.intel.com/benchmarks. **Configuration:** Testing by Intel as of June 13, 2018. Intel® Core™ i7-6700K CPU @ 2.90GHz fixed, GPU GT2 @ 1.00GHz fixed Internal ONLY testing, Test v3.15.21 – Ubuntu* 16.04, OpenVINO 2018 RC4, Intel® Arria® 10 FPGA 1150GX. Tests were based on various parameters such as model used (these are public), batch size, and other factors. Different models can be accelerated with different Intel hardware solutions, yet use the same Intel software tools.

Intel's compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor-dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice. Notice revision #20110804

SUMMARY

- OpenVINO™ Toolkit is free to download and enables you to deploy on Intel FPGAs directly from TensorFlow or Caffe
- Intel's FPGA architecture is a versatile choice for deep learning applications

INTEL FPGAS ENABLE

First to market to accelerate evolving AI workloads

Flexible system level functionality for key AI system requirements

RESOURCES

IFTD18

Intel FPGA Training

https://www.altera.com/support/training/catalog.html

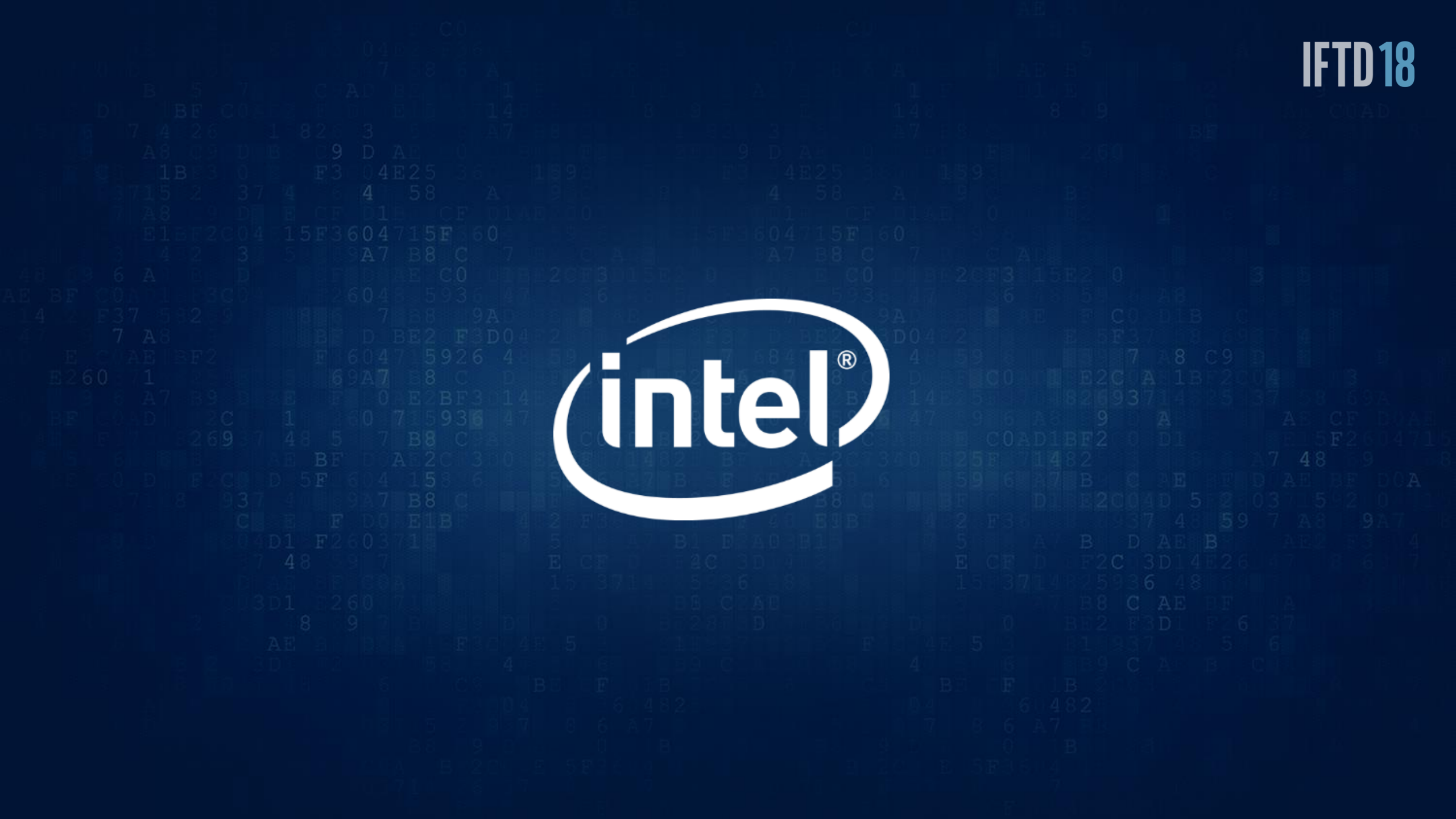
Download ▸
Free OPENVINO™ toolkit

Get started quickly with:

- Find out more online at www.intel.com/ai and Intel FPGA website
- Developer resources
- Intel Tech.Decoded online webinars, tool how-tos & quick tips
- Hands-on in-person events

Support

- Connect with Intel engineers & AI experts via the public Community Forum


IFTD18
intel®