Xilinx ML Landscape

Daniele Bagni
DSP / ML Specialist (EMEA Lead)
Agenda

- Introduction
- Edge
- Cloud & Server
- Wrap-up
Introduction
Deep Learning Models – Major Area of Focus

Multi-Layer Perceptron
- Classification
- Universal Function Approximator
- Autoencoder

Convolutional Neural Network
- Feature Extraction
- Object Detection
- Image Segmentation

Recurrent Neural Network
- Sequence and Temporal Data
- Speech to Text
- Language Translation

Classification

Object Detection

Segmentation

“Dog”
Two main phases: Training and Inference

1. **Training:** Process for machine to “learn” and optimize a model from data
2. **Inference/deployment:** use trained models to predict outcomes from new observations
What do customers need to start machine learning?

**Knowledge of..**
- Linux (as a user)
- Basic Python
  - Familiarity with Anaconda
- Framework
  - Caffe/Tensorflow

**For training..**
- A machine suitable for training
  - Nvidia GPU and as much memory as possible
  - Cloud: AWS, Nimbix, Google Co-Lab
- Nvidia tools
  - CUDA, cuDNN
- Datasets for training & validation
  - Do you have a dataset that covers all your use-cases?

**For inference/deployment on Cloud/DC**
- Xilinx ML Suite
- Cloud Service (Nimbix, AWS..)
- Alveo boards in server

**For inference/deployment on Edge**
- DNNDK toolset
  - Free version downloadable from deephi.com
  - Good idea to get a supported eval board
    - ZCU102, ZCU104
    - Ultra96, DP-8020
  - Xilinx SDSoC or Vivado
    - SDSoC integrates DPU as C-callable IP
There are many frameworks available. Currently a very fast-changing landscape. Xilinx is focused on Caffe & TensorFlow.
Xilinx Acquires DeePhi in September 2018
One of the hottest AI startups in the World

Xilinx Announces the Acquisition of DeePhi Tech
Deal to Accelerate Data Center and Intelligent Edge Applications

BEIJING and SAN JOSE, Calif., July 17, 2018 – Xilinx, Inc. (NASDAQ: XLNX), the leader in adaptive and intelligent computing, announced today that it has acquired DeePhi Tech, a Beijing-based privately held start-up with industry-leading capabilities in machine learning, specializing in deep compression, pruning, and system-level optimization for neural networks.

1. Unique Technology for Deep Learning Acceleration
2. Proven Competitive Advantage with Customers Worldwide
3. Data Center & Core Markets Growth Acceleration

DeePhi’s Products will be Marketed, Sold and Supported as Xilinx Products
## The Most Interactive Demo

<table>
<thead>
<tr>
<th>ML task</th>
<th>Models</th>
<th>Resolution</th>
<th>Operations</th>
<th>Throughput</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pose estimation</td>
<td>Pruned SSD + GoogleNet</td>
<td>480<em>360 (detection), 128</em>224 (pose)</td>
<td>4.3G + 0.6G</td>
<td>~12 FPS</td>
</tr>
<tr>
<td>Pedestrian detection</td>
<td>Pruned SSD</td>
<td>480*360</td>
<td>4.3G</td>
<td>~12 FPS</td>
</tr>
<tr>
<td>Face detection</td>
<td>DenseBox</td>
<td>640*360</td>
<td>1.2G</td>
<td>~12 FPS</td>
</tr>
<tr>
<td>Multi-class video analytics</td>
<td>Pruned SSD</td>
<td>480*360</td>
<td>5.5G</td>
<td>~12 FPS</td>
</tr>
</tbody>
</table>

**MIPI Camera**

**Logi Camera**

**GOpro Camera**

**ZCU102**

**AVI file on SD-card**
Edge & Cloud Stacks

Xilinx AI Development

<table>
<thead>
<tr>
<th>Models</th>
<th>Edge/Embedded</th>
<th>Cloud/DC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20+ pruned / customized / basic models</td>
<td></td>
</tr>
<tr>
<td>DeePhi Pruning</td>
<td>DePhi Pruning</td>
<td>DePhi Pruning</td>
</tr>
<tr>
<td>DePhi Quantizer</td>
<td>DePhi Quantizer</td>
<td>fxDNN Compiler</td>
</tr>
<tr>
<td>DePhi Compiler</td>
<td>SDSoc</td>
<td>SDAccel</td>
</tr>
<tr>
<td>DePhi Runtime</td>
<td>Xilinx AI Development</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Software Stack</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>DePhi DPU</td>
<td>DeePhi LSTM</td>
</tr>
<tr>
<td></td>
<td>DeePhi LSTM</td>
<td>xDNN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FPGA IP</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ZU9 Card</td>
<td>ZU102</td>
</tr>
<tr>
<td></td>
<td>ZCU102</td>
<td>ZCU104 Ultra96</td>
</tr>
<tr>
<td></td>
<td>Z7020 Board</td>
<td>ZU2/3 Card</td>
</tr>
<tr>
<td></td>
<td>Z7020 SOM</td>
<td>ZU2/3 SOM</td>
</tr>
<tr>
<td></td>
<td>ZU2/3 Card</td>
<td></td>
</tr>
</tbody>
</table>

Platforms

Xilinx U200, U250, U280

DeePhi
- Quantizer
- Compiler
- Runtime

SDSoC
- DePhi Quantizer

SDAccel
- fxDNN Compiler
- fxDNN Runtime

DeePhi
- LSTM

Xilinx U200, U250, U280

Platforms

ZU9 Card
ZCU102
ZCU104
Ultra96
Xilinx SDSoC (part of SDx) for ML on Edge

Xilinx ZU9

<table>
<thead>
<tr>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frames/s</td>
<td>60</td>
</tr>
<tr>
<td>Power (W)</td>
<td>4.8</td>
</tr>
<tr>
<td>Latency (ms)</td>
<td>16.7</td>
</tr>
<tr>
<td>Utilization</td>
<td>15%</td>
</tr>
</tbody>
</table>

```
main()
    imread(A);
    imread(B);
    denseOpticalFlowPyrLtr(A,B,out);
    imshow(out);
}
```

---

- nVidia number using CUDA OpenCV
- Both Xilinx and nVidia benchmarks do not include the camera inputs and HDMI/DP
- LK dense optical flow, non-pyramidal, non-iterative, Window size 53x53
Xilinx SDAccel (part of SDx) for ML on Cloud/DC

> Develop, profile and deploy OpenCL applications
  >> OpenCL uses standard APIs (code is portable)

> F1 platform aware

> Flexible kernels development
  >> C / C++ / OpenCL / RTL

Programming Steps

1. Identify Application for Acceleration
2. Code and Optimize Kernel, Host
3. Compile and Execute for GPU
4. Estimate Kernels
5. Debug Kernels with Cycle Accurate Models
6. Compile for FPGA
7. Execute and Validate on F1

Comprehensive debug and profiling environment
Edge & Cloud Stacks

Xilinx AI Development

Models

Edge/Embedded

Cloud/DC

Software Stack

DeePhi Quantizer

DeePhi Compiler

DeePhi Runtime

FPGA IP

DeePhi DPU

 Platforms

Z7020 Board
ZU9 Card
Z7020 SOM
ZU2/3 SOM
ZU2/3 Card
ZU2/3 Card
ZCU104
ZCU102
Ultra96

ZC104

DeePhi Pruning

DeePhi Quantizer

DeePhi Compiler

xfDNN Compiler

xfDNN Runtime

SDSoC

SDAccel

DeePhi LSTM

xDNN

Xilinx U200, U250, U280
Edge
Why Xilinx for Edge/Embedded ML?

1. Only HW/SW configurable device for fast changing networks
2. High performance / low power with custom internal memory hierarchy
3. Future proof to lower precisions
4. Low latency end-to-end
Unique, Patented Deep Learning Acceleration Techniques

- Best paper awards for breakthrough DL acceleration
- Deephi’s compression technology
  - Reduce DL accelerator footprint into smaller devices
  - Increase performance per watt (higher performance and/or lower energy)

Unique Pruning Technology Provides a Significant Competitive Advantage
DeePhi Solution Stack for Edge/Embedded ML

Models
- Face detection
- Pose estimation
- Video analytics
- Lane detection
- Object detection
- Segmentation

Framework
- Caffe
- Darknet
- TensorFlow

Tools & IP
- DeePhi DNNDK
- DeePhi DPU

HW Platforms
- Z7020 Board
- Z7020 SOM
- ZU2 SOM
- ZU2/3 Card
- ZU9 Card
- ZCU102
- ZCU104
- Ultra96

Deephi also has LSTM IP for KU115/VU9P as a part of Cloud ML
DNNDK – Deep Neural Network Development Kit

> DECENT (DEep ComprEssioN Tool)
> DNNC (Deep Neural Network Compiler)
> Runtime N²Cube (Cube of Neural Network)
> Profiler DSight

Customer Platform (Board, OS)
What is the DPU?

> The Deep Processor Unit (DPU) is a soft IP core.

> DPU looks a bit like a micro-processor
  > has an instruction set suitable to run CNN algorithms
  > DPU acts like a co-processor to ARM

> Once instantiated into the HW Embedded Design, the DPU architecture is fixed and does not change.
  > User can select from different architectures: B1152, B2034, B4096…
  > The number indicates the number of operations per clock cycle
    - The usual Programmable Logic rule of ‘more performance = more resources’ applies

> The DPU works in 8bit Fixed Point
  > The Quantization process generates an ELF file with Fixed Point weights and operations to be executed on the DPU

> Not every kind of network layer is supported by the DPU
  > some will need to be implemented on ARM
DPU Typical Options & Interfaces

> **3-level parallelism is exploited**
  >> Pixel * input channel * output channel

> **Small core - B1152**
  >> Parallelism: 4*12*12
  >> target Z7020/ZU2/ZU3

> **Big core - B4096**
  >> Parallelism: 8*16*16
  >> Target ZU5 and above

> > level parallelism is exploited
  >> Pixel * input channel * output channel

> **Small core - B1152**
  >> Parallelism: 4*12*12
  >> target Z7020/ZU2/ZU3

> **Big core - B4096**
  >> Parallelism: 8*16*16
  >> Target ZU5 and above
Perf Improvement with DPU_EU

Performance Comparison (FPS)

<table>
<thead>
<tr>
<th>Model</th>
<th>Current B4096*2 wo Prune</th>
<th>New B4096*3 wo Prune</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-SSD</td>
<td>12</td>
<td>28.3</td>
</tr>
<tr>
<td>VGG16</td>
<td>73</td>
<td>92</td>
</tr>
<tr>
<td>ResNet50</td>
<td>118</td>
<td>179</td>
</tr>
<tr>
<td>GoogLeNet</td>
<td>313</td>
<td>445</td>
</tr>
</tbody>
</table>

Resource Utilization Comparison

<table>
<thead>
<tr>
<th></th>
<th>DSP</th>
<th>LUT</th>
<th>FF</th>
<th>BRAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>DPU B4096*2</td>
<td>2048</td>
<td>156744</td>
<td>224650</td>
<td>501</td>
</tr>
<tr>
<td>DPU_EU B4096*3</td>
<td>1926</td>
<td>110311</td>
<td>255020</td>
<td>748.5</td>
</tr>
</tbody>
</table>

*The FPS of VGG-SSD of end to end performance
*The FPS of VGG16/ResNet50/GoogLeNet is of CONV part (w/o FC layer)
DPU Scalability

Peak INT8 OPS*

* With heterogenous DPUs

© Copyright 2019 Xilinx
Supported Operators

- Arbitrary Input Image Size
- Conv
  - Arbitrary Conv Kernel Size
  - Arbitrary Conv Stride/Padding
  - Dilation
- Pooling
  - Max/Avg Pooling
    - Arbitrary Max Pooling Size
    - Avg Pooling kernel size: 2x2~7x7
    - Arbitrary Pooling Stride/Padding
  - ReLU / Leaky Relu / Relu6
  - Concat
- Deconv
- Depthwise conv
- Elementwise
- FC(Int8/FP32)
- Mean scale
- Upsampling
- Batch Normalization
- Split
- Reorg
- Resize (Optional)
- Softmax (Optional)
- Sigmoid (Optional)
Framework Support

Caffe

- Pruning
- Quantization
- Compilation

Quantization & Compilation
- Eval version
- Pruning
  - Eval version

Pruning
- Quantization
- Convertor to Caffe

© Copyright 2019 Xilinx
Model Compression With DECENT

- Consists of two separate tools
  - Pruning Tool - decent_p
  - Quantization Tool – decent_q

- Effects
  - Compress model size 5x ~ 100x
  - Compress running time 1.5x – 10x

Pruning is optional, Quantization is mandatory
Quantization Results

**Uniform Quantization**
- 8-bit for both weights and activation
- A small set of images for calibration

<table>
<thead>
<tr>
<th>Networks</th>
<th>Float32 baseline</th>
<th>8-bit Quantization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top1</td>
<td>Top5</td>
</tr>
<tr>
<td>Inception_v1</td>
<td>66.90%</td>
<td>87.68%</td>
</tr>
<tr>
<td>Inception_v2</td>
<td>72.78%</td>
<td>91.04%</td>
</tr>
<tr>
<td>Inception_v3</td>
<td>77.01%</td>
<td>93.29%</td>
</tr>
<tr>
<td>Inception_v4</td>
<td>79.74%</td>
<td>94.80%</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>74.76%</td>
<td>92.09%</td>
</tr>
<tr>
<td>VGG16</td>
<td>70.97%</td>
<td>89.85%</td>
</tr>
<tr>
<td>Inception-ResNet-v2</td>
<td>79.95%</td>
<td>95.13%</td>
</tr>
</tbody>
</table>
Pruning Example - SSD

SSD+VGG @Deephi Surveillance 4classes

Pruning Speedup on DPU (SSD)
Pruning Makes Big Difference

Result of DeePhi Pruning

(SSD 480x360)

<table>
<thead>
<tr>
<th>Device</th>
<th>FPS (batch=1)</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jetson TX2</td>
<td>10W</td>
<td>10W</td>
</tr>
<tr>
<td>ZU9</td>
<td>10W</td>
<td>10W</td>
</tr>
<tr>
<td>ZU5</td>
<td>5W</td>
<td>3W</td>
</tr>
<tr>
<td>ZU2</td>
<td>3W</td>
<td>2W</td>
</tr>
<tr>
<td>7020</td>
<td>2W</td>
<td></td>
</tr>
</tbody>
</table>

© Copyright 2019 Xilinx
## Pruning Results

### Classification Networks

<table>
<thead>
<tr>
<th>Model</th>
<th>Baseline</th>
<th>Pruning Result 1</th>
<th>Pruning Result 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-5</td>
<td>ΔTop5</td>
<td>ratio</td>
</tr>
<tr>
<td>Resnet50 [7.7G]</td>
<td>91.65%</td>
<td>91.23%</td>
<td>-0.42%</td>
</tr>
<tr>
<td>Inception_v2 [4.0G]</td>
<td>91.07%</td>
<td>90.37%</td>
<td>-0.70%</td>
</tr>
<tr>
<td>SqueezeNet [778M]</td>
<td>83.19%</td>
<td>82.46%</td>
<td>-0.73%</td>
</tr>
</tbody>
</table>

### Detection Networks

<table>
<thead>
<tr>
<th>Model</th>
<th>Baseline mAP</th>
<th>Pruning Result 1</th>
<th>Pruning Result 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP</td>
<td>ΔmAP</td>
<td>ratio</td>
</tr>
<tr>
<td>DetectNet [17.5G]</td>
<td>44.46</td>
<td>45.7</td>
<td>+1.24</td>
</tr>
<tr>
<td>SSD+VGG [117G]</td>
<td>61.5</td>
<td>62.0</td>
<td>+0.5</td>
</tr>
<tr>
<td>[A] SSD+VGG [173G]</td>
<td>57.1</td>
<td>58.7</td>
<td>+1.6</td>
</tr>
<tr>
<td>[B] Yolov2 [198G]</td>
<td>80.4</td>
<td>81.9</td>
<td>+1.5</td>
</tr>
</tbody>
</table>
DNNDK Hybrid Compilation Model

Neural Network → DNCC → DPU Assembly → DNNAS → DPU ELF Object → Linker → HybridExecutable

C/C++ DL Application → GCC/LLVM → CPU Assembly → Assembler → CPU ELF Object
Edge: Runtime

Computer Vision App (DPU-accelerated)

User Space
- Industry-standard Libraries
- Loader
- Tracer
- Library

Kernel Space
- Operating System
- DPU Driver

Hardware Platform
- Host CPU
- DPU

Runtime N²Cube
- Library
- Loader
- Tracer
- Driver
Integration with SDSoC

- DPU is provided as C-callable IP library in SDSoC
  - Currently, only B4096 available

- Downloadable now on xilinx.com
  - DNNDK for SDSoC
  - UG1331
```c
int main(int argc, char *argv[])
{
    DPUKernel *kernel_conv;
    DPUKernel *kernel_fc;
    DPUSubTask *task_conv;
    DPUSubTask *task_fc;
    char *input_addr;
    char *output_addr;

    /* DNNDK API to attach to DPU driver */
    dpuInit();

    /* DNNDK API to create DPU kernels for CONV & FC networks */
    kernel_conv = dpuLoadKernel("resnet50_conv", 224, 224);
    kernel_fc = dpuLoadKernel("resnet50_fc", 1, 1);

    /* Create tasks from CONV & FC kernels */
    task_conv = dpuCreateTask(kernel_conv);
    task_fc = dpuCreateTask(kernel_fc);

    /* Set input tensor for CONV task and run */
    input_addr = dpuGetTensorAddress(dpuGetTaskInputTensor(task_conv));
    setInputImage(Mat Bimage, input_addr);
    dpuRunTask(task_conv);
    output_addr = dpuGetTensorAddress(dpuGetTaskOutputTensor(task_conv));

    /* Run average pooling layer on CPU */
    run_average_pooling(output_addr);

    /* Set input tensor for FC task and run */
    input_addr = dpuGetTensorAddress(dpuGetTaskInputTensor(task_fc));
    setFCInputData(task_fc, input_addr);
    dpuRunTask(task_fc);
    output_addr = dpuGetTensorAddress(dpuGetTaskOutputTensor(task_fc));

    /* Display the Classification result from FC task */
    displayClassificationResult(output_addr);

    /* DNNDK API to destroy DPU tasks/kernels */
    dpuDestroyTask(task_conv);
    dpuDestroyTask(task_fc);
    dpuDestroyKernel(kernel_conv);
    dpuDestroyKernel(kernel_fc);

    /* DNNDK API to detach from DPU driver and free DPU resources */
    dpuFinish();

    return 0;
}
```
Basic and Professional Editions

DeePhi Basic
- Compiler
- Quantizer
- Pruned Models
- Unlimited Deployment

DeePhi Professional
- 3-day On-site Training
- Pruning Tools
- Compiler
- Quantizer
- Pruned Models
- Unlimited Deployment

Free
- Everything you need to do it yourself

Node-Locked / Floating License
- Access Pruning Technology
- 3-day on-site training by a top-notch ML expert
- 30-day evaluation with encrypted pruning output
<table>
<thead>
<tr>
<th>Tutorial</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR10 Caffe Tutorial (UG1355)</td>
<td>Train, quantize, and prune custom CNNs with the CIFAR10 dataset using Caffe and the Xilinx® DNNDK tools.</td>
</tr>
<tr>
<td>Cats vs Dogs Tutorial (UG1336)</td>
<td>Train, quantize, and prune a modified AlexNet CNN with the Kaggle Cats vs Dogs dataset using Caffe and the Xilinx DNNDK tools.</td>
</tr>
</tbody>
</table>

https://github.com/Xilinx/Edge-AI-Platform-Tutorials
# Edge AI Hub

## Edge AI Tools

<table>
<thead>
<tr>
<th>Product</th>
<th>Documentation</th>
<th>Tool Download</th>
<th>File Size</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNNDK</td>
<td>DNNDK User Guide (UG1327)</td>
<td>xlnx_dnnndk_2.08_1901.tar.gz</td>
<td>841 MB</td>
<td>4af979bda0c0b830a1533442529cfb0c4</td>
</tr>
<tr>
<td>DNNDK for SDSoC</td>
<td>DNNDK User Guide for SDSoC (UG1331)</td>
<td>xlnx_dnnndk_for_sdsoo_2.08_1901.tar.gz</td>
<td>667 MB</td>
<td>f6cf20a4422cd40c94914eae6f5c23</td>
</tr>
</tbody>
</table>

## Edge AI Evaluation Boards

<table>
<thead>
<tr>
<th>Product</th>
<th>Documentation</th>
<th>Image Download</th>
<th>File Size</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCU102 Kit</td>
<td>ZCU102 User Guide (UG1182)</td>
<td>2018-12-04-zcu102-desktop-stretch.img.zip</td>
<td>571 MB</td>
<td>d0d5f8e8e80b965591d0975565a5d</td>
</tr>
<tr>
<td>ZCU104 Kit</td>
<td>ZCU104 User Guide (UG1267)</td>
<td>2018-12-04-zcu104-desktop-stretch.img.zip</td>
<td>571 MB</td>
<td>ada2420c4af8d9973e7a741e0917ee26</td>
</tr>
<tr>
<td>Avnet Ultra 96</td>
<td>Ultra 96 User Guide</td>
<td>xilinx-ultra96-desktop-stretch-2018-12-10.img.zip</td>
<td>566 MB</td>
<td>c5d422d63213b4bc4c18a32236eada8</td>
</tr>
</tbody>
</table>

## Platform Downloads

<table>
<thead>
<tr>
<th>Product</th>
<th>Download</th>
<th>File Size</th>
<th>MD5 Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZCU102 SDSoC 2018.3 Platform for DNNDK</td>
<td>zcu102-rv-s8-2018-3-dnnndk.tar.gz</td>
<td>1.3 GB</td>
<td>7102c6942eb65d8b9d258914f69c6eaa</td>
</tr>
<tr>
<td>ZCU104 SDSoC 2018.3 Platform for DNNDK</td>
<td>zcu104-rv-s8-2018-3-dnnndk.tar.gz</td>
<td>1.3 GB</td>
<td>d5bc80a8135a719e273e2f5ca85762</td>
</tr>
</tbody>
</table>

Video Surveillance ML Solutions

Intelligent IP Camera Solution

Video Analytics Acceleration Solution

Face recognition camera with Zynq7020

12-channel 1080P Video Analytics with ZU9EG
ADAS/AD ML Reference Design

2D/3D Object Detection

Lane Detection

Segmentation

Pedestrian Detection

Segmentation + Detection

Pose Estimation
8CH Detection Demo

- Xilinx device
  - ZU9EG

- Network
  - SSD compact version

- Input image size to DPU
  - 480 * 360

- Operations per frame
  - 4.9G

- Performance
  - 30fps per channel
4-ch Segmentation + Detection Demo

> Xilinx device
  > ZU9EG

> Network
  > FPN compact version
  > SSD compact version

> Input image size to DPU
  > FPN – 512 * 256
  > SSD – 480 * 360

> Operations per frame
  > FPN – 9G
  > SSD – 4.9G

> Performance
  > 15fps per channel
Edge & Cloud Stacks

Xilinx AI Development

Models
- Edge/Embedded
  - 20+ pruned / customized models
- Cloud/DC
  - DeePhi Pruning

Software Stack
- DeePhi Quantizer
- DeePhi Compiler
- DeePhi Runtime
- SDSoc
- SDAccel
- xfDNN Compiler
- xfDNN Runtime

FPGA IP
- DeePhi DPU

Platforms
- Z7020 Board
- Z7020 SOM
- ZU2/3 SOM
- ZU2/3 Card
- ZU9 Card
- ZCU102
- ZCU104
- Ultra96

Xilinx U200, U250, U280

DeePhi
- Runtime
- Compiler
- Quantizer
- Pruning

SDSoC

SDAccel

xDNN

DeePhi

Xilinx AI Development
Cloud & Server
FPGA Apps in the AWS Marketplace

**Mipsology**
- ZEBRA on 1 FPGA (image classification)
  - **Features**: Users FPGA-based class-lending association for neural network inference. The user-defined neural network works on Zebra just as it would on a GPU or CPU. Zebra...
  - **License**: DSP Core 7.2 or 8.0 on Amazon Machine Image (AMI)

**Zebra Deep-Learning engine for Caffe (1 FPGA)**
- **Features**: Supports 1.5x faster with 10 GPUs
  - **Starting from**: $85 to $89 per software + AWS usage fees
  - Zebra accelerates neural network inference using FPGA. User-defined neural networks are computed by Zebra (or they would be on a CPU or CPU). Zebra is fully integrated...
  - **License**: DL Core 12.0 to 14.9 on Amazon Machine Image (AMI)

**FPGA Accelerated Deep-Learning Inference with Binarized Neural Networks**
- **Features**: 1.3x faster with cutting-edge electronics.
  - **Starting from**: $50 to $59 per software + AWS usage fees
  - Image classification of the CIFAR10 dataset using the CNN neural network. Based on Xilinx public proof-of-concept implementation of a redefined precision. Binarized Neural Network...
  - **License**: L4 Core 10.0 to 11.9 on Amazon Machine Image (AMI)

**Visual System Integrator for FPGA and Embedded Development**
- **Features**: 1.0.6.1 updated by Systek Vision
  - **Starting from**: $200/hr for up to 128 FPGAs/Per hour: $500 usage fee for software + AWS usage fees
  - Visual System Integrator is a one-of-a-kind tool for embedded development which, for the first time, makes it possible to develop a fully functioning system. Visual System...
  - **License**: Control Core 7.3: $40 on Amazon Machine Image (AMI)

**Merlin Compiler AMI**
- **Features**: 1.0.6 updated by Falcon Computing Solutions, Inc.
  - 14 Day Free Trial Available - The Merlin Compiler AMI is provided by Falcon Computing Solutions, Inc. The AMI is pre-built with Merlin Compiler that provides push button CSCI ...
  - **License**: Control Core 7.3: $40 on Amazon Machine Image (AMI)

**FPGA accelerated HEIF-to-JPEG Transcoder, HEVC decoder.**
- **Features**: 1.0.8 updated by Fovea Technology Inc.
  - New Accelerate your HEVC decoding over FPGA using f1.20, using Pathpartner’s HEVC decoder available as ffmpeg plugin. This component can be invoked in the ffmpeg chain...
  - **License**: Control Core 7.3: $40 on Amazon Machine Image (AMI)

**InTime**
- **Features**: 2.8.0.1 updated by Ruffly
  - In Time is an automated optimization software for FPGA design by Ruffly. It optimizes timing and design performance using machine learning to find the best combination of...
  - **License**: Control Core 7.3: $40 on Amazon Machine Image (AMI)

---

**NGCode HEVC/H.265 Encoder D01**
- **Features**: 1.0.1.1 updated by NGCode
  - Starting from $150.00 to $500.00 for software + AWS usage fees
  - Using an F1 Instance, offshore HEVC encoding to an FPGA. This version of the NGCode Encoder features 1678 Frame encoding up to 1080p60 encoder/Frame rate. The performance...
  - **License**: Control Core 7.3: $40 on Amazon Machine Image (AMI)

**Machine Learning Development Stack from Xilinx, Preview Edition**
- **Features**: 1.0.7.16.1 updated by Xilinx
  - In-Machine Learning Development Stack, Preview Edition AMI users easily integrate machine learning into their current applications and deploy from quickly. Users can...
  - **License**: Control Core 7.3: $40 on Amazon Machine Image (AMI)

**Accelerated Machine Learning**
- **Features**: 1.0.4.10 updated by Xilinx
  - Starting from $200 to $250/hr for software + AWS usage fees
  - The Xilinx Accelerated Machine Learning Library allows the model to be deployed in the field, with performance and efficiency.
  - **License**: Control Core 16.0 to 16.9 on Amazon Machine Image (AMI)

**DRAGEN Complete Suite - Exome (approx. $2 per Exome)**
- **Features**: 1.0.2.11 updated by Bioxl Gene
  - Starting from $20 to $22.20/hr for software + AWS usage fees
  - The DRAGEN Complete Suite (Exome) analyzes ultra-rapid analysis of Next Generation Sequencing (NGS) data for small data sets, such as whole exomes and targeted panels. This...
  - **License**: Control Core 7.2 to 8.0 on Amazon Machine Image (AMI)

**DRAGEN Complete Suite - Genome (approx. $15 per Genome)**
- **Features**: 1.0.2.12 updated by Bioxl Gene
  - Starting from $10 to $12/hr for software + AWS usage fees
  - The DRAGEN Complete Suite (Genome) can analyze ultra-rapid analysis of Next Generation Sequencing (NGS) data for large data sets, such as whole genomes. This application uses...
  - **License**: Control Core 7.2 to 8.0 on Amazon Machine Image (AMI)
Amazon F1 Instances

<table>
<thead>
<tr>
<th>Model</th>
<th>#FPGA</th>
<th>Mem</th>
<th>SSD Storage</th>
<th>FPGA DDR4</th>
<th>Price / Hour</th>
</tr>
</thead>
<tbody>
<tr>
<td>f1.2xlarge</td>
<td>1</td>
<td>122 GB</td>
<td>470 GB</td>
<td>4x16 GB</td>
<td>$1.65</td>
</tr>
<tr>
<td>f1.16xlarge</td>
<td>8</td>
<td>976 GB</td>
<td>8 x 470 GB</td>
<td>8 x 4x16 GB</td>
<td>$13.20</td>
</tr>
</tbody>
</table>

EC2 instance type with up to 8 VU9P (16nmFF+) FPGA
Intel Xeon processors with up to 16 cores
Connected through PCIe gen3x16
4 local DDR4 (12 GBps) per FPGA

For development, the FPGA Developer AMI can be installed on cheaper instances such as the R4:
- R4.xlarge (30GB RAM) can be $0.13 per hour on “spot instance” (about double for “on-demand”)
- R4.2xlarge (61GB RAM) is $0.27 per hour on “spot instance” (about double for “on-demand”)
## Server Platforms
- Intel x86
- AMD Epyc
- Power9
- ARM

## Faas
- AWS Marketplace
- Nimbix
- Alibaba Cloud
- Huawei

## Xilinx Boards
- Alveo U200
- Alveo U250

*Under Development*
ML Deployment with Open Source Software
xDNN Compiler + Runtime

Deep Learning Frameworks

Deep Learning Frameworks

Frontend

Framework Tensor Graph to Xilinx Tensor Graph

Tensor Graph Optimization

Compiler

Quantizer

Runtime

Image

CPU Layers

FPGA Layers

Model Weights

Calibration Set

https://github.com/Xilinx/ml-suite
xfDNN Runtime Engine

> ML specific engine built on top of SDx runtime
> Lightweight and portable with no dependency on ML frameworks
> Extensive C++/Python API with simplified use model
> Asynchronous xDNN execution
> Streaming/Pipeline/Dataflow support for large imageset
> Multiple CNN models running on single FPGA
> Multiple FPGA support
Xilinx DNN Processor (xDNN)

- Configurable Overlay Processor
- DNN Specific Instruction Set
  - Convolution, Max Pool etc.
- Any Network, Any Image Size
- High Frequency & High Compute Efficiency
- Compile and run new networks
xDNN PEs Optimized for Throughput or Latency

<table>
<thead>
<tr>
<th>PE</th>
<th>#DSPs</th>
<th>Cache</th>
<th>16 bit GOP/s</th>
<th>8 bit GOP/s</th>
<th>Optimized For</th>
<th>Examples Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td>28x32</td>
<td>896</td>
<td>5 MB</td>
<td>896</td>
<td>1,792</td>
<td>Multi-Network, Maximum Throughput</td>
<td>ResNet50 (224x224)</td>
</tr>
<tr>
<td>56x32</td>
<td>1792</td>
<td>5 MB</td>
<td>1,792</td>
<td>3,584</td>
<td>Lowest Latency Streaming</td>
<td>Yolov2 (224x224)</td>
</tr>
<tr>
<td>56x32</td>
<td>1792</td>
<td>8 MB</td>
<td>1,792</td>
<td>3,584</td>
<td>Lowest Latency, High Resolution</td>
<td>Yolov2 (605x605), ResNet50 (512x512)</td>
</tr>
</tbody>
</table>
Flexible: Multi-Network Configuration

1 FPGA Provides 4 Virtual Accelerators
For Real Time Deep Learning
Rapid Feature and Performance Improvement

**xDNN-v1**
- Q4CY17
- Array of Accumulator
- Int16 (Batch=1) and Int8 (Batch=2) support
- Instructions: Convolution, ReLU, Pool, Elementwise
- Flexible kernel size(square) and strides
- 500 MHz

**xDNN-v2**
- Q2CY18
- All xDNN-v1 Features
- DDR Caching: Larger Image size
- New Instructions: Depth-wise Convolution, De-convolution, Up-sampling
- Rectangular Kernels
- 500 MHz

**xDNN-v3**
- Q4CY18
- New Systolic Array Implementation: 2.2x lower latency
- Instruction Level Parallelism – non-blocking data movement
- Batch=1 for Int8 – lower latency
- Feature compatible with xDNN-v2
- 720+ MHz
Alveo boards

<table>
<thead>
<tr>
<th></th>
<th>U200</th>
<th>U250</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUTs</td>
<td>892K</td>
<td>1,341K</td>
</tr>
<tr>
<td>Internal SRAM Capacity</td>
<td>35MB</td>
<td>54MB</td>
</tr>
<tr>
<td>Internal SRAM Bandwidth</td>
<td>31TB/s</td>
<td>38TB/s</td>
</tr>
<tr>
<td>CNN Throughput*</td>
<td>3100img/s</td>
<td>4100img/s</td>
</tr>
</tbody>
</table>

*Low-latency GoogLeNet v1
xDNN v3 Implementation on Alveo U200

- 3 Large 96x16 PEs – 1 in each SLR – 5.2 ML Shell
- Kernels @ 720 MHz/360MHz

<table>
<thead>
<tr>
<th>Resource</th>
<th>Count</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUTs</td>
<td>658k</td>
<td>52%</td>
</tr>
<tr>
<td>DSPs</td>
<td>5661</td>
<td>80%</td>
</tr>
<tr>
<td>BRAM</td>
<td>1258</td>
<td>58%</td>
</tr>
<tr>
<td>URAM</td>
<td>864</td>
<td>92%</td>
</tr>
</tbody>
</table>
xDNN v3 Implementation on Alveo U250

- 4 Large 96x16 PEs – 1 in each SLR – standard 5.2 Shell
- Kernels at 700 MHz/350 MHz

<table>
<thead>
<tr>
<th>Resource</th>
<th>Count</th>
<th>Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>LUTs</td>
<td>876k</td>
<td>51%</td>
</tr>
<tr>
<td>DSPs</td>
<td>7548</td>
<td>62%</td>
</tr>
<tr>
<td>BRAM</td>
<td>1632</td>
<td>61%</td>
</tr>
<tr>
<td>URAM</td>
<td>1152</td>
<td>90%</td>
</tr>
</tbody>
</table>
xDNN GoogLeNet v1 Performance – Image Size 224x224
xDNN YOLO v2 Performance – Image Size 608x608

Alveo U200 Latency Mode (INT8) - 34 ms

Alveo U250 Latency Mode (INT8) - 117 ms
Additional Information

> Visit Alveo Demo Room

> Refer to White Paper WP504
  >> “Accelerating DNNs with Xilinx Alveo Accelerator Cards”

Wrap - Up
Key Messages

➢ Our current focus is mainly on image processing applications
  ➢ For both Edge & Clouds solutions
  ➢ We don’t currently have solutions for other types of applications
    – Work on recurrent networks/LSTM underway
      ▪ Malware detect, DoS attacks, Natural Language Processing

➢ We can interface to standard ML frameworks..
  ➢ Edge (DNNDK, DPU): Caffe today, TensorFlow very soon
  ➢ Cloud (xFDNN, xDNN): Caffe, TensorFlow, MXNet

➢ We can’t do network training, only deployment/inference
  ➢ We don’t supply datasets
Highly Scalable Solutions

Featuring the Most Powerful FPGA in the Cloud

Deep Learning Applications

Cloud

Alveo

On Premises

Zynq SoC

Edge
# Edge v’s Cloud

<table>
<thead>
<tr>
<th>Applications</th>
<th>Edge</th>
<th>Cloud/DC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Conv Neural Nets</td>
<td>• Conv Neural Nets</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <em>On-going development for Recurrent networks</em></td>
</tr>
<tr>
<td>ML framework support</td>
<td>• Caffe</td>
<td>• Caffe</td>
</tr>
<tr>
<td></td>
<td>• Tensorflow soon</td>
<td>• Tensorflow</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• MxNet</td>
</tr>
<tr>
<td>Target platform</td>
<td>• Zynq-7000/UltraScale+</td>
<td>• Boards - Alveo</td>
</tr>
<tr>
<td>CPU</td>
<td>• ARM A9/A53</td>
<td>• x86 (via PCIe)</td>
</tr>
<tr>
<td>Development tools</td>
<td>• DNNDK</td>
<td>• ML Suite</td>
</tr>
<tr>
<td>Pruning</td>
<td>• Yes (DECENT_p)</td>
<td>• No (<em>Will use DECENT_p in future</em>)</td>
</tr>
<tr>
<td>Quantization</td>
<td>• Yes (DECENT_q)</td>
<td>• Yes (xdnnQuantize)</td>
</tr>
<tr>
<td>Programming API</td>
<td>• Yes (C++)</td>
<td>• Yes (Python)</td>
</tr>
</tbody>
</table>
Adaptable. Intelligent.
Please Read: Important Legal Notices

The information disclosed to you hereunder (the "Materials") is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of Xilinx’s limited warranty, please refer to Xilinx’s Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in such critical applications, please refer to Xilinx’s Terms of Sale which can be viewed at http://www.xilinx.com/legal.htm#tos.

Automotive Applications Disclaimer

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.

© Copyright 2013–2016 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. All other trademarks are the property of their respective owners.