Xilinx AI \ ML for Automotive

Presented By

Pechorsky Michael
Xilinx FAE, Avnet Israel
Feb 2020
Xilinx’s Steady Growth in Automotive

Shipments: 138M total, 62M ADAS

<table>
<thead>
<tr>
<th>CY</th>
<th>Makes</th>
<th>Models</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>14</td>
<td>29</td>
</tr>
<tr>
<td>2015</td>
<td>19</td>
<td>64</td>
</tr>
<tr>
<td>2016</td>
<td>23</td>
<td>85</td>
</tr>
<tr>
<td>2017</td>
<td>26</td>
<td>96</td>
</tr>
<tr>
<td>2018</td>
<td>29</td>
<td>111</td>
</tr>
</tbody>
</table>

Tier-1 Suppliers

OEMs
News: XA MPSoC Family for Safety Critical ADAS & AD

Leading Quality | Beyond AEC-Q100 Certified to ISO26262 | Right Performance/Watt
Xilinx in the Automotive
ADAS and AD Focus Applications

ADAS Applications
- Front Camera
- LiDAR
- Full Display Mirror
- Surround View Camera
- RADAR
- Driver Monitoring System

AD Central Module Functions
- Compute Acceleration
- Data Aggregation & Pre-processing
- Sensor Fusion

System
XA offers a Scalable and “Migratable” Family of Devices for ADAS/AD

- **Driver Assistance/Partial Automation – L1/2**: Sensor Fusion, Warning & Temporary limited Control - E.g. Lane Keeping or Emergency Braking
- **Driver Awareness/Assistance – L0/1**: Camera, Radar, Ultrasonic Warning - E.g. Park Assist, Blind Spot Warning, LDW
- **Partial/Conditional Automation – L2/3**: Sensor Fusion + Complex Control at limited speeds - E.g. Stop and Go Traffic

Device Qual/Rel, Des Integrity, Diag/Failure Mitigation

Processing Complexity within Product Constraints

Functional Safety (ISO26262)

- Zynq-7020
- Zynq-7030
- Zynq Ultrascale+ (ZU2-ZU5)
- Artix-7

- **S6 / A7**: Zynq-7010, Zynq-7020

Scalability across and within device families

High Volume Automotive Production Today

Automotive Production Early 2018

In Final Definition and Development

7nm Next Gen AP SoC

Performance, Power, Integration, Productivity

ECU/Processing Platform
Why Xilinx for Automated Driving?

- Extensible External Device interfacing (e.g., Sensor)
- Parallel & Multi-Channel Data Preprocessing
- High Bandwidth Connectivity for Data Aggregation
- Power Efficient Processing for DNN Inference
- Computing Power
- Power consumption
- Scalable Device Families: Optimum Cost vs. Compute Performance
- Multi-channel Processing Pipelines for Diverse Functionality and Safety
- Upgradeable Security & Authentication
- Automotive Quality Processes and History
- Roadmap to 7nm process
- Xilinx Automotive Grade from 2004
Xilinx Solutions for AI
Deep Learning: Training and Inference

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

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

Xilinx Focus
Unique, Patented Deep Learning Acceleration Techniques

> Best paper awards for breakthrough DL acceleration

> Xilinx compression technology can:

  >> Reduce DL accelerator footprint into smaller devices

  >> Increase performance per watt (higher performance and/or lower energy)

Unique Pruning Technology Provides a Significant Advantage
### Supported DNN (Deep Neural Network) by Applications

<table>
<thead>
<tr>
<th>Application</th>
<th>Function</th>
<th>Algorithm</th>
<th>Developed</th>
<th>Pruned</th>
<th>Deployed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face</td>
<td>Face detection</td>
<td>SSD, Densebox</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Landmark Localization</td>
<td>Coordinates Regression</td>
<td>✔</td>
<td>N/A</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Face recognition</td>
<td>ResNet + Triplet / A-softmax Loss</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Face attributes recognition</td>
<td>Classification and regression</td>
<td>✔</td>
<td>N/A</td>
<td>✔</td>
</tr>
<tr>
<td>Pedestrian</td>
<td>Pedestrian Detection</td>
<td>SSD</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Pose Estimation</td>
<td>Coordinates Regression</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Person Re-identification</td>
<td>ResNet + Loss Fusion</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Video Analytics</td>
<td>Object detection</td>
<td>SSD, RefineDet</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Pedestrian Attributes Recognition</td>
<td>GoogleNet</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Car Attributes Recognition</td>
<td>GoogleNet</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Car Logo Detection</td>
<td>DenseBox</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Car Logo Recognition</td>
<td>GoogleNet + Loss Fusion</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>License Plate Detection</td>
<td>Modified DenseBox</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>License Plate Recognition</td>
<td>GoogleNet + Multi-task Learning</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ADAS/AD</td>
<td>Object Detection</td>
<td>SSD, YOLOv2, YOLOv3</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>3D Car Detection</td>
<td>F-PointNet, AVOD-FPN</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Lane Detection</td>
<td>VPGNet</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Traffic Sign Detection</td>
<td>Modified SSD</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Semantic Segmentation</td>
<td>FPN</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td></td>
<td>Drivable Space Detection</td>
<td>MobilenetV2-FPN</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Multi-task (Detection+Segmentation)</td>
<td>Deephi</td>
<td>✔</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Pruning Example - SSD

Performance Speedup

<table>
<thead>
<tr>
<th>Operations (G)</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>117G</td>
<td>18</td>
</tr>
<tr>
<td>19G</td>
<td>71</td>
</tr>
<tr>
<td>11.6G</td>
<td>103</td>
</tr>
</tbody>
</table>

2x DPU-4096@ZU9
Performance Optimization with End-to-end Profiling

> Before optimization

Preprocessing: 9.5ms
DPU: 19.5ms
Softmax: 17ms
Pixel classification: 16ms
Frame overlay: 39ms
Others: 4ms

> After optimization

Preprocessing: 9.5ms
DPU: 19.5ms
Softmax: 1ms
Frame overlay: 2ms
Pixel classification: 6ms
Others: 4ms

- Use Vitis libraries / custom hardware IPs
- Use NEON to accelerate the computing
- Use Eigen/BLAS library to perform linear superposition
## Pruning Results

<table>
<thead>
<tr>
<th>Classification Networks</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>Top-5 ΔTop5 ratio</td>
<td>Top-5 ΔTop5 ratio</td>
</tr>
<tr>
<td>Resnet50 [7.7G]</td>
<td>91.65%</td>
<td>91.23% -0.42% 40%</td>
<td>90.79% -0.86% 32%</td>
</tr>
<tr>
<td>Inception_v2 [4.0G]</td>
<td>91.07%</td>
<td>90.37% -0.70% 60%</td>
<td>90.07% -1.00% 55%</td>
</tr>
<tr>
<td>SqueezeNet [778M]</td>
<td>83.19%</td>
<td>82.46% -0.73% 89%</td>
<td>81.57% -1.62% 75%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Detection Networks</th>
<th>Baseline mAP</th>
<th>Pruning Result 1</th>
<th>Pruning Result 2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP ΔmAP ratio</td>
<td>mAP ΔmAP ratio</td>
<td></td>
</tr>
<tr>
<td>DetectNet [17.5G]</td>
<td>44.46 45.7 +1.24 63%</td>
<td>45.12 +0.66 50%</td>
<td></td>
</tr>
<tr>
<td>SSD+VGG [117G]</td>
<td>61.5 62.0 +0.5 16%</td>
<td>60.4 -1.1 10%</td>
<td></td>
</tr>
<tr>
<td>[A] SSD+VGG [173G]</td>
<td>57.1 58.7 +1.6 40%</td>
<td>56.6 -0.5 12%</td>
<td></td>
</tr>
<tr>
<td>[B] Yolov2 [198G]</td>
<td>80.4 81.9 +1.5 28%</td>
<td>79.2 -1.2 7%</td>
<td></td>
</tr>
</tbody>
</table>
## Quantization Result

<table>
<thead>
<tr>
<th>Classification Networks</th>
<th>Float32 baseline</th>
<th>8 bit quantized</th>
<th>After quantized finetune</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top1</td>
<td>Top5</td>
<td>Top1</td>
</tr>
<tr>
<td>Inception_v1</td>
<td>66.90%</td>
<td>87.68%</td>
<td>66.54%</td>
</tr>
<tr>
<td>Inception_v2</td>
<td>72.78%</td>
<td>91.04%</td>
<td>71.93%</td>
</tr>
<tr>
<td>Inception_v3</td>
<td>77.01%</td>
<td>93.29%</td>
<td>76.26%</td>
</tr>
<tr>
<td>Inception_v4</td>
<td>79.74%</td>
<td>94.80%</td>
<td>79.04%</td>
</tr>
<tr>
<td>ResNet-50</td>
<td>74.76%</td>
<td>92.09%</td>
<td>73.74%</td>
</tr>
<tr>
<td>VGG16-3fc-float</td>
<td>70.97%</td>
<td>89.85%</td>
<td>70.67%</td>
</tr>
<tr>
<td>MobileNet_v1</td>
<td>70.61%</td>
<td>89.63%</td>
<td>68.01%</td>
</tr>
</tbody>
</table>

## Detection Networks

<table>
<thead>
<tr>
<th>Detection Networks</th>
<th>Dataset</th>
<th>Float mAP</th>
<th>8 bit quantized mAP</th>
<th>ΔmAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD_VGG</td>
<td>VOC 21 classes</td>
<td>76.47%</td>
<td>76.27%</td>
<td>-0.20%</td>
</tr>
<tr>
<td>SSD_MobileNet_v2</td>
<td>BDD100k 11 classes</td>
<td>30.80%</td>
<td>29.70%</td>
<td>-1.10%</td>
</tr>
<tr>
<td>SSDLite_MobileNet_v2</td>
<td>Customer's data</td>
<td>20.28%</td>
<td>20.12%</td>
<td>-0.16%</td>
</tr>
</tbody>
</table>
Xilinx- AI/ML Toolchain flow

Xilinx AI Development

- Edge/Embedded
  - 20+ pruned / customized / basic models

- Cloud/DC
  - Deephi Pruning
    - Deephi Quantizer
    - Deephi Compiler
    - Deephi Runtime
  - Vitis AI
    - xDNN Compiler
    - xDNN Runtime

Software Stack

- Deephi DPU
- Deephi LSTM

FPGA IP

- Deephi DPU
- Deephi LSTM

Platforms

- Z7020 Board
- Z7020 SOM
- ZU2/3 SOM
- ZU2/3 Card
- ZU9 Card
- ZCU102
- ZCU104
- Ultra96
- Xilinx U200, U250, U280

AVNET

Microsoft Azure

Nimbix

Xilinx
Enables All Developers to Build and Deploy AI to All Platforms
Tracif{(is uyvy)} {uyvy2bgr(in_mat, in_rgb)}
else {yuyv2bgr(in_mat, in_rgb)}

resize<INTERPOLATION_AREA, MAX_IN_HEIGHT, MAX_IN_WIDTH, MAX_OUT_HEIGHT, MAX_OUT_WIDTH, NPC, MAX_DOWN_SCALE>(in_r, out_r);
cv.cpp

Application Example: Smart City
Xilinx DeePhi Soft IP
DeePhi DPU Scalability

Typical DPU IP Configurations

> **B1024**
  >> target Z7020/ZU2

> **B1152**
  >> target Z7020/ZU2/ZU3

> **B4096**
  >> target ZU5 and above

Integration in Xilinx Tools

Software callable ML IP

```
void dpu_set_finish(uint32_t *finish);
void dpu_set_start(uint32_t *start);
void dpu_set_profen(uint32_t *prof_en);
void dpu_set_finish_clr(uint32_t finish);
```
## DPU Peak Perf & Power

<table>
<thead>
<tr>
<th></th>
<th>LUT</th>
<th>Flip-Flops</th>
<th>Block RAM</th>
<th>DSP&lt;sup&gt;1)&lt;/sup&gt;</th>
<th>DPU Config</th>
<th>Peak&lt;sup&gt;3)&lt;/sup&gt; performance</th>
<th>Frequency</th>
<th>Board Power&lt;sup&gt;4)&lt;/sup&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>7020</td>
<td>53200</td>
<td>106400</td>
<td>4.9Mb</td>
<td>220</td>
<td>1xB1152</td>
<td>230GOPS</td>
<td>200MHz</td>
<td>3.5 / 6.5W</td>
</tr>
<tr>
<td>ZU2</td>
<td>47000</td>
<td>94000</td>
<td>5.3Mb</td>
<td>240</td>
<td>1xB1152</td>
<td>576GOPS</td>
<td>500MHz</td>
<td>5.5 / 9.1W</td>
</tr>
<tr>
<td>ZU3</td>
<td>71000</td>
<td>141000</td>
<td>7.6Mb</td>
<td>360</td>
<td>1xB2304</td>
<td>852GOPS</td>
<td>370MHz</td>
<td>N/A</td>
</tr>
<tr>
<td>ZU5</td>
<td>117000</td>
<td>234000</td>
<td>5.1Mb+18Mb</td>
<td>1248</td>
<td>1xB4096</td>
<td>1.433TOPS</td>
<td>350MHz</td>
<td>N/A</td>
</tr>
<tr>
<td>ZU7EV</td>
<td>230000</td>
<td>461000</td>
<td>11Mb+27Mb</td>
<td>1728</td>
<td>2xB4096</td>
<td>2.7TOPS</td>
<td>330MHz</td>
<td>N/A</td>
</tr>
<tr>
<td>ZU9</td>
<td>274000</td>
<td>548000</td>
<td>32.1Mb</td>
<td>2520</td>
<td>3xB4096</td>
<td>4.05TOPS</td>
<td>330MHz</td>
<td>17 / 31W</td>
</tr>
</tbody>
</table>

1) One DSP48E is used for two int8 multiplication
2) MACs is constructed by DSP and LUT (if DSP is not enough)
3) Peak performance is calculated by MACs: GOPS = 2*MACs*Frequency
4) The power listed here is average and peak power
**Example of Video Surveillance AI Solutions**

- **Platform**: MPSoC ZU2EG
- **Size**: 50*70 mm
- **DPU**: B1152
- **Peak perf.**: 576Gops (500Mhz)

<table>
<thead>
<tr>
<th>Platform</th>
<th>Size: 50*70 mm</th>
<th>DPU: B1152</th>
<th>Peak perf.: 576Gops (500Mhz)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>FPS</th>
<th>Power</th>
<th>FPS</th>
<th>Power</th>
<th>FPS</th>
<th>Power</th>
<th>FPS</th>
<th>Power</th>
<th>FPS</th>
<th>Power</th>
<th>FPS</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GOP 200MHz</strong></td>
<td>7.7</td>
<td>15.3</td>
<td>5.3</td>
<td>18.38</td>
<td>5.55</td>
<td>22.3</td>
<td>5.9</td>
<td>24.87</td>
<td>6.32</td>
<td>27</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td><strong>GOP 250MHz</strong></td>
<td>3.8</td>
<td>23.6</td>
<td>5.29</td>
<td>28.24</td>
<td>5.54</td>
<td>33.5</td>
<td>5.89</td>
<td>37.8</td>
<td>6.2</td>
<td>39.6</td>
<td>6.52</td>
<td></td>
</tr>
<tr>
<td><strong>GoogLeNet</strong></td>
<td>3.2</td>
<td>36.5</td>
<td>5.26</td>
<td>45.90</td>
<td>5.54</td>
<td>53.8</td>
<td>5.97</td>
<td>61.7</td>
<td>6.43</td>
<td>68.2</td>
<td>6.74</td>
<td></td>
</tr>
<tr>
<td><strong>GoogLeNet^1)</strong></td>
<td>1.6</td>
<td>62</td>
<td>5.24</td>
<td>77.11</td>
<td>5.59</td>
<td>93</td>
<td>6.03</td>
<td>109.4</td>
<td>6.41</td>
<td>116</td>
<td>6.72</td>
<td></td>
</tr>
<tr>
<td><strong>SSD</strong></td>
<td>117</td>
<td>1.63</td>
<td>5.4</td>
<td>2.03</td>
<td>5.71</td>
<td>2.44</td>
<td>6.22</td>
<td>2.835</td>
<td>6.52</td>
<td>3.23</td>
<td>7.05</td>
<td></td>
</tr>
<tr>
<td><strong>SSD^1)</strong></td>
<td>11.6</td>
<td>13.2</td>
<td>5.47</td>
<td>16.35</td>
<td>5.77</td>
<td>19.34</td>
<td>6.18</td>
<td>22.43</td>
<td>6.65</td>
<td>25.3</td>
<td>7.05</td>
<td></td>
</tr>
</tbody>
</table>

---

**Platform**: MPSoC ZU2EG

**Size**: 50*70 mm

**DPU**: B1152

**Peak perf.**: 576Gops (500Mhz)
Framework Support

Caffe
- Pruning tool @Caffe
- Quantization tool @Caffe

Tensorflow
- Quantization tool
- Pruning tool

PyTorch

Darknet
- Pruning tool @darknet
- Quantization tool @darknet
- Convertor for caffe deploy
- Yolo V2 compression
Vitis AI

Frameworks
- TensorFlow
- Caffe
- PyTorch

Vitis AI Models
- AI Optimizer
- AI Quantizer
- AI Compiler
- AI Runtime
- AI Library
- AI Profiler

Vitis AI Development Kit
- XRT

DSA
- CNN DPU

50+ pretrained, optimized reference models
Performance improvement up to 10-20x
Unified runtime enables edge to cloud

* Coming Soon
Recap of Xilinx Value Proposition in Edge ML

Xilinx offers the optimal tradeoff among latency, power, cost, flexibility, scalability & time-to-market for Edge ML.
Thank You