Architecting Always-On, Context-Aware, On-Device AI Using Flexible Low-power FPGAs

Deepak Boppana – Senior Director Product & Segment Marketing
Gordon Hands – Director Solutions Marketing
Rapidly Emerging Edge Computing Trend
Driven by Latency, Privacy, and Bandwidth Limitations

Unit growth for edge devices with AI will explode increasing over 110% CAGR over the next five years – Semico Research
Always-on, On-device AI Applications
Human Presence Detection Example

Smart Home Appliance
LCD turns on when needed

Consumer Electronics
TV turns off when no one is present

Smart DoorBell
Rings automatically when needed

Vending Machine
LCD turns on when needed

Security Camera
Alerts when intruder present, not a cat

Smart Doors
Opens when person is present
Always-on, On-device AI Applications

Other Examples

- Smart speakers
  - Key phrase detection
- Retail store cameras
  - Face tracking
- Selfie drones
  - Face tracking
- Toll gate camera
  - Vehicle classification
- Machine vision
  - Object counting
- After market automotive cameras
  - Speed sign detection
Always-on, On-device AI Requirements
Unmet Need for Ultra-Low Power, Scalable, and Flexible Inferencing

- Few mWs of Power Consumption
- Few $s of BOM Cost Adder
- Few mm² of Board Area

Flexible Legacy Interface Support

- I²C
- SPI
- PCIE

Neural Network Accelerator

- Ethernet
- USB

Customized Performance/Accuracy
Flexible and Scalable Inferencing at the Edge
From under 1 mW to 1 W with Lattice sensAI

![Graph showing power vs. performance for different zones: MPU Zone, GPU Zone, HIGH-END FPGA Zone, MCU Zone.](image-url)
Stand-alone, Integrated FPGA Solution

- Always-on, integrated solutions on ECP5 or iCE40 UltraPlus FPGA
- Low latency and secure implementation
- Small form factor packages from 5.5 mm² to 100 mm²
FPGA as Activity Gate to ASIC/ASSP

- **iCE40 UltraPlus FPGA** for always-on detection of key-phrases or objects
- Wakes-up a high performance ASIC/ASSP for further analytics only when required
- Reduces overall system power consumption
FPGA as a Co-Processor to MCU

- Scalable performance/power with ECP5 based neural network acceleration
- ECP5 based IO flexibility to seamlessly interface to on-board legacy devices including sensors
- Low-end MCU for flexible system control
Delivering Edge CNN Acceleration in Lattice FPGA

Caffe

CNN Accelerator IP

System Interfaces

Neural Network Complier

RTL

FPGA Bitstream

Instructions

Lattice FPGA
CNN Accelerator IP Architecture

Control Unit:
- AXI Master
- CMD Queue
- State Machine

Engine Pool:
- Conv EU
- FC EU
- Pooling EU

Sequence:
- Seq Gen 0
- Seq Gen 1
- Seq Gen 15

Memory Pool:
- Mem 0
- Mem 1
- Mem 15

DRAM:
- AXI Master
- Save/Load
- Input/Output/Intermediate

Parameters:
- Upward arrow connecting Conv EU and FC EU to Pooling EU.
CNN Compact Accelerator IP Architecture

- **Input**
  - Activation storage
  - Convolution
  - Pool

- **Output**
  - Fully Connected
  - Scaler
  - ReLU

- **Commands & weights**

- **Engine**
  - Control Unit
  - FIFO
  - Control

- **Control Unit**
Translating Trained Neural Network Into Lattice CNN Accelerator Instructions

1. Load
2. Review
3. Analyze
4. Compile
5. Simulate
## On-device AI – Complex Optimization

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Design Factors</th>
<th>Device</th>
<th>Network</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td># of Engines</td>
<td>Local Memory</td>
</tr>
<tr>
<td>Power (W)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Device Size</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance (fps)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Accuracy (%)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Small Object (% fov)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Correlation Between Design Factors and Product Attributes
## Examples for Illustration

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Number of Multiplications</th>
<th>Input Size</th>
<th>Quantization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Face Detection</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGG style</td>
<td>290,816</td>
<td>32<em>32</em>3</td>
<td>16-bit fixed point</td>
</tr>
<tr>
<td>VGG style</td>
<td>14,353,920</td>
<td>90<em>90</em>3</td>
<td>16-bit fixed point</td>
</tr>
<tr>
<td><strong>Human Presence Detection</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGG style</td>
<td>8,570,880</td>
<td>64<em>64</em>3</td>
<td>16-bit fixed point</td>
</tr>
<tr>
<td>VGG style</td>
<td>338,558,976</td>
<td>128<em>128</em>3</td>
<td>16-bit fixed point</td>
</tr>
</tbody>
</table>
Image Based Neural Networks on Lattice FPGAs

ECP5

- CNN Accelerator
  1 – 8 engines
  0.25 – 2 Mbit Local Memory
- ISP Engine
- Overlay Engine
- SPI to DDR loader
- DDR Memory
- SPI Memory

UltraPlus

- CNN Accelerator
  8 Multipliers
  0.5 – 1 Mbit Local Memory
- Down sample
- SPI Memory
Image Based Neural Networks Lattice Hardware

Himax HM01B0 UPduino Shield

Embedded Vision Development Kit
Face Detect Implementations

32 x 32 Input

- UltraPlus
- ECP5-25
- ECP5-45
- ECP5-85

![Graph](image)

* Running at 5 frames per second

90 x 90 Input

- UltraPlus
- ECP5-25
- ECP5-45
- ECP5-85

![Graph](image)

- 0.5 W 100 mm²
- 0.6 W 100 mm²
- 0.8 W 100 mm²

www.latticesemicom/sensAI
Human Presence Detect Implementations

64 x 64 Input

128 x 128 Input

* Running at 5 frames per second
## Bringing It Together

<table>
<thead>
<tr>
<th>Network</th>
<th>Smallest Object</th>
<th>UltraPlus 1 – 7 mW* 5.5 mm²</th>
<th>ECP5-25 0.5 W 100 mm²</th>
<th>ECP5-45 0.6 W 100 mm²</th>
<th>ECP5-85 0.8 W 100 mm²</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face Detection 32 x 32 Input</td>
<td>50%</td>
<td>465</td>
<td>3360</td>
<td>4511</td>
<td>5251</td>
</tr>
<tr>
<td>Face Face Detection 90 x 90 Input</td>
<td>20%</td>
<td>--</td>
<td>28</td>
<td>82</td>
<td>101</td>
</tr>
<tr>
<td>Human Presence Detect 64 x 64 Input</td>
<td>20%</td>
<td>18</td>
<td>115</td>
<td>161</td>
<td>338</td>
</tr>
<tr>
<td>Human Presence Detect 128 x 128 Input</td>
<td>10%</td>
<td>--</td>
<td>2.3</td>
<td>3.5</td>
<td>5.4</td>
</tr>
</tbody>
</table>

* Running at 5 frames per second
Summary

- AI at the edge solves real world problems
- FPGAs can implement AI standalone or in conjunction with other components
- sensAI stack components provide edge AI building blocks
  - Silicon, soft IP, tools, development boards & reference designs
- Configurable engine size and bit widths coupled with multiple target devices allows system optimization
  - 1 mW – 1 W
  - 5.5 mm^2 – 100 mm^2
Resources

Please visit latticesemi.com/sensAI for more information and downloads

- 4 ECP5 Based Reference Designs / Demonstrations – Free
- 4 iCE40 Based Reference Designs / Demonstrations – Free
- CNN Accelerator IP – Free Evaluation
- CNN Compact Accelerator IP – Free
- Neural Network Compiler – Free
- Embedded Vision Development Kit – $199 Promotional Price
- Himax HM01B0 UPduino Shield – Available November ~$49
Empowering Product Creators to Harness Embedded Vision

The Embedded Vision Alliance (www.Embedded-Vision.com) is a partnership of 90+ leading embedded vision technology and services suppliers, and solutions providers.

Mission: Inspire and empower product creators to incorporate visual intelligence into their products.

The Alliance provides low-cost, high-quality technical educational resources for product developers.


The Alliance enables vision technology providers to grow their businesses through leads, ecosystem partnerships, and insights.

For membership, email us: membership@Embedded-Vision.com.
The only industry event focused on enabling product creators to create “machines that see”

• “Awesome! I was very inspired!”
• “Fantastic. Learned a lot and met great people.”
• “Wonderful speakers and informative exhibits!”

Embedded Vision Summit 2019 highlights:

• Inspiring keynotes by leading innovators
• High-quality, practical technical, business and product talks
• Exciting demos of the latest apps and technologies

Visit www.EmbeddedVisionSummit.com to sign up for updates
Q & A
Thank you