The Path to Embedded Vision & AI using a Low Power Vision DSP

Yair Siegel, Director of Segment Marketing
Hotchips – August 2016
Presentation Outline

- Introduction
- The Need for Embedded Vision & AI
- Vision Processor
- Deep Learning Path to Low Power AI Devices
- Summary
CEVA – Licensor of ultra-low-power signal processing IP’s for embedded devices

Imaging & Vision
Audio, Voice, Sensing
Connectivity
Communication

>7 Billion CEVA-powered devices shipped world-wide
Corporate Introduction

Corporate Facts

- Headquartered in Mountain View, Calif.
- Publicly traded - NASDAQ:CEVA
- 273 employees, >200 eng.
- Profitable, cash positive, >$138M cash
- World Leading IP Supplier since 1991

Worldwide Operations

- US (Mountain View, CA, Austin, TX & Detroit, MI), Israel, Ireland, France, U.K, Sweden, China, Taiwan, Korea & Japan
CEVA Value Chain

**CEVA**
- Silicon IP
- Development Tools
- Support, SW
- Cooperation

**IDM**
- Partner
- Cooperation with Foundries and Design Service Companies

**Chips**

**OEM**
- Solutions (HW + SW + Services)

Providing hardware, software, tools and support from license to deployment!
The Need for Embedded Vision & AI

- Security & Surveillance
- Visual Perception & Analytics
- ADAS & Autonomous Cars
- Augmented Reality
- Drones
CEVA-XM4 Vision DSP Highlights

1. 1.5GHz max frequency @28nm HPM
   - 14 stage deep pipeline

2. 128 MACs, 8-way VLIW, 4096-bit vector engine

3. 32-way parallel random memory access
   - Enables serial code vectorization

4. Sliding window & sliding pattern mechanisms
   - Efficient 2-dimension data processing

5. Silicon proven, 15+ design wins for variety of products

6. Completed ISO 26262 compliance & process certification
CEVA-XM4 Performance Highlights

- Fixed- and floating-point math
  - Available both in vector and scalar structures
- 8/16/32/64-bit fixed point precision
- Native non linear operations in single cycle
  - $\frac{1}{x}$, $\sqrt{x}$, $\sqrt[3]{x}$
  - Supported both in fixed and float operations
- Multiple 128 or 256-bit AXI interfaces
- Extendible ISA

<table>
<thead>
<tr>
<th></th>
<th>Multipliers per cycle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scalar fixed-point</td>
<td>4</td>
</tr>
<tr>
<td>Scalar Floating-point</td>
<td>4</td>
</tr>
<tr>
<td>Vector fixed-point</td>
<td>128</td>
</tr>
<tr>
<td>Vector Floating-point</td>
<td>16</td>
</tr>
</tbody>
</table>

High Performance, yet flexible in precision and operation
CEVA-XM4 Block Diagram

- **L1 Program Memory**
- **Program Mem Subsystem**
  - Program DMA
  - Program Cache
- **Program Control Unit**
  - PSU
  - Emulation
    - Profilers
    - Real-Time Trace
- **Scalar Processing Unit**
  - SPU0
  - SPU1
  - SPU2
  - SPU3
- **Load / Store Unit**
  - LS0
  - LS1
- **Data Memory Subsystem**
  - Message Queues
  - DMA Q Manager
  - Data DMA
- **Multi-Core Interfaces**
- **Automatic management of HW accelerators**
- **4-Way I-Cache**
- **AXI Matrix**
- **Specialized DMA**
- **CEVA-Xtend**
  - User Instructions (Vector & Scalar)
- **L1 Data Memory**
- **CEVA-Connect**
  - Customer HW accelerators

- **Parallel Random Mem Access**
- **Scalar Floating-Point**
- **128 MACs**
- **Vector Floating-Point**
CEVA Vision Platform Layers

**Software Layer**
- CEVA Software Products
  - Digital Video Stabilizer (DVS)
  - Super-Resolution (SR)
- Partner Software Products
  - Face Detection & Recognition
  - Universal Object Recognition
  - Pedestrian Detection
  - ADAS Algorithms (FCW, LDW)
  - 3D Depth Map Creation

**App Dev. Kit (ADK)**
- Host CEVA-CV API / OpenVX
- RTOS
- CPU-DSP Link – Communication Layer
- CPU offload
- CEVA-CV Libraries
- CEVA CNN Framework (CDNN)

**Hardware Layer**
- CEVA-XM4 DSP Core

---
*available for XM4 only

CEVA Inc. All rights reserved
CEVA-ToolBox™ - Eclipse based SW Dev. Tools

- Advanced Eclipse-based IDE
- Optimizing C/C++ tailored compiler
  - Auto vectorization
  - Extensive Vec-C support
    - C language extensions in OpenCL-like syntax
    - Vector types for C/C++ - short8, ushort32,…
    - Vectorization from operators
- Linker and Utilities
- Automatic Build Optimizer
- CEVA-Xtend GUI Instruction builder
- Built-in Debugger, Simulator & Profiler
- Target Emulation dev. kit

Complete Software Development tools.
Focused on ease of use and quick SW porting for performance optimization
Parallel Random Memory Access Mechanism

- CEVA-XM4 scatter-gather capability enables load/store vector elements from/into multiple memory locations in a single cycle
  - Enables serial code vectorization
  - Able to load values from 32 addresses per cycle

Example: Image histogram requires random accesses to memory per pixel
  - Each “value” (within a vector) can generate an address that allows it to vectorize/parallel the operations into a multiple operation within a single cycle
Histogram Power Analysis

Special histogram HW consumes 1.89x higher power per cycle, but kernel is 100x faster $\Rightarrow$ ~50x better power efficiency for entire calculation!

Using Special Histogram HW Support

Without Histogram Special HW Support (conventional DSP / CPU Approach)
Sliding-Window Data Processing Mechanism

- CEVA-XM4 processes 2D data efficiently
- Takes advantage of pixel overlap in image processing by reusing same data to produce multiple outputs
  - Significantly increases processing capability
  - Saves external memory bandwidth and frees system buses for other tasks
  - Reduces power consumption

For 16 MAC with 512-bit bandwidth, only 176 bits actually loaded
The vector instruction performs the arithmetic operation multiple times in parallel (producing multiple results). The next sliding-window operation is offset by a step from the previous operation.

<table>
<thead>
<tr>
<th>Input Data Vector A</th>
<th>0 0 1 2 2 3 3 4 4 5 6 7 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input Data Vector B</td>
<td>8 9 10 10 11 12 13 14 15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X X X X X</th>
<th>X X X X X</th>
<th>X X X X X</th>
<th>X X X X X</th>
<th>X X X X X</th>
</tr>
</thead>
<tbody>
<tr>
<td>A B C D</td>
<td>A B C D</td>
<td>A B C D</td>
<td>A B C D</td>
<td>A B C D</td>
</tr>
</tbody>
</table>

+ 0
+ 1
+ 2
+ 3
Sliding-Window Pattern Mechanism

- Sliding-window pattern (i.e. sliding pattern) efficiently handles sparse filters
- Sparse filters are defined as filters that contain some coefficients equal to zero
- Instead of multiplying the input data by zero, the sliding pattern skips zero coefficients and improves efficiency (and keeps power low)
Sliding-Window Pattern Example

Example below describes a sparse filter
- ~45% non-zero coefficients

Achieves 92% multiplier utilization, completing filter in 3 cycles

“Standard” implementation: ~55% multiplier utilization in 5 cycles
### Sliding-Window Pattern Power Savings

<table>
<thead>
<tr>
<th></th>
<th>Sliding Window Utilization</th>
<th>Sliding Pattern Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution 5x5</td>
<td>62.5%</td>
<td>89%</td>
</tr>
<tr>
<td>Sobel 3x3</td>
<td>75%</td>
<td>75%</td>
</tr>
<tr>
<td>Average 7x7</td>
<td>87.5%</td>
<td>94%</td>
</tr>
</tbody>
</table>

#### CEVA-XM4 "Sliding Window"

- Convolution 5x5: 89%
- Sobel 3x3: 75%
- Average 7x7: 94%

#### CEVA-XM4 "Sliding Pattern"

- Convolution 5x5: 89%
- Sobel 3x3: 75%
- Average 7x7: 94%
Sliding Window, Sliding Pattern Power Savings

Relative Power Consumption

- CPU
- GPU
- CEVA-XM4 Scalar Only
- CEVA-XM4 "Sliding Window"
- CEVA-XM4 "Sliding Pattern"
CEVA Deep Neural Network (CDNN2)

- 2nd gen SW framework support
  - Caffe and TensorFlow Frameworks
  - Various networks*
  - All network topologies
  - All the leading layers
  - Variable ROI
  - “Push-button” conversion from pre-trained networks to optimized real-time
  - Accelerates machine learning deployment for embedded systems
  - Optimized for CEVA-XM4 vision DSP

(*) Including AlexNet, GoogLeNet, ResNet, SegNet, VGG, NIN and others
CEVA Network Generator

Image Database
(For Training)

Network Structure
(Deploy.prototxt)

Normalization Image
(Deploy.binaryproto)

Network Labeling
(Deploy.labels)

Caffe
TensorFlow

Pre-Trained Network
(Deploy.caffemodel)

CEVA Network Generator

CEVA-XM4
Optimized Network
Real-Time CDNN2 Application Flow

Input Image → Application Pre-process: Scaling, background reduction, ROI selection → ROI → CNN Application API → Full Network Implementation → CDNN2 Real-Time Libraries → Application Post-process: Complete recognition, ... → “Horse”
Real-Time CNN Object Recognition Demo

- Live Alexnet object recognition
- Enables milli-watt products vs. watts on GPU

Live CDNN2 demo
Convert network in under 10min!
Vision & Deep Learning SoC Example

- Typical SoC for recording devices such as action / dash / surveillance cameras
- Enables OEM differentiation by adding highest-end camera features
- Ultra low power: Full vision system below 1w@28nm

![vision_soc_diagram]
High-end vision system, Available Early 2017
AR/VR Ref System Example using Inuitive NU4000

High-end AR/VR system supporting depth sensing, deep learning, SLAM and vision, but cutting power by a factor
Why DSP Is Better as Vision Processor?

- Domain-specific architecture – focused on computer vision
  - Dedicated ISA & mechanisms enable higher perf & utilization
  - 8-way VLIW - up to 8 separate instructions combined in parallel
  - 32-way parallel load-store maintainable per cycle

- Combination of strong vector and scalar types of code
  - CPU good mostly in scalar code
  - GPU good mostly for parallel portion, weak on memory accesses

- Enables flexible fixed-point math for better power efficiency
  - Combines floating-point for time-to-market and higher dynamic range

- Maximizing local data reuse, limiting DDR memory bandwidth
  - Significant memory power saving, GPU not able to utilize well
  - Enables more efficient deep learning algo development

**CEVA-XM4 in combination with SW platform saves time-to-market and extends device battery life**
Welcome to visit us at the demo table to hear more

Thank You

Contact: Yairs@ceva-dsp.com