OpenVX Extensions
May 2018

Radhakrishna Giduthuri | Editor, OpenVX Working Group
radha.giduthuri@amd.com | @RadhaGiduthuri
Why extensions?

- Why not keep all features in the main specification?
  - A burden when not required by customers
  - Some features are not feasible across all hardware

- How to make new features quickly available?
  - Main specification release cycles are long
  - Extensions can be released quickly

- Get feedback from industry about certain features
  - Publish the features as a provisional extension
# OpenVX Extensions

<table>
<thead>
<tr>
<th>Extension Name</th>
<th>OpenVX Versions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vx_khr_nn</td>
<td>1.2</td>
<td>Neural network extension</td>
</tr>
<tr>
<td>vx_khr_ix</td>
<td>1.2, 1.1</td>
<td>Export and import extension for graphs &amp; objects</td>
</tr>
<tr>
<td>vx_khr_import_kernel</td>
<td>1.2, 1.1</td>
<td>Import kernel from vendor binary</td>
</tr>
<tr>
<td>vx_khr_pipelining</td>
<td>1.2, 1.1</td>
<td>Pipelining, streaming, and batch processing extension</td>
</tr>
<tr>
<td>vx_khr_opencl</td>
<td>1.2, 1.1</td>
<td>OpenCL interop extension</td>
</tr>
<tr>
<td>vx_khr_class</td>
<td>1.2</td>
<td>Classifier extension</td>
</tr>
<tr>
<td>vx_khr_s16</td>
<td>1.1</td>
<td>S16 extension</td>
</tr>
<tr>
<td>vx_khr_tiling</td>
<td>1.0</td>
<td>Tiling extension</td>
</tr>
<tr>
<td>vx_khr_xml</td>
<td>1.0</td>
<td>XML schema extension</td>
</tr>
<tr>
<td>vx_khr_icd</td>
<td>1.0</td>
<td>Installable client driver extension</td>
</tr>
</tbody>
</table>

provisional
OpenVX Neural Network Extension

- Convolution Neural Network topologies can be represented as OpenVX graphs
  - Layers are represented as OpenVX nodes
  - Layers connected by multi-dimensional tensors objects
  - Layer types include convolution, activation, pooling, fully-connected, soft-max
  - CNN nodes can be mixed with traditional vision nodes

- Import/Export Extension
  - Efficient handling of network Weights/Biases or complete networks

- OpenVX will be able to import NNEF files into OpenVX Neural Nets
OpenVX Neural Network Extension

- Tensor types supported: INT16(Q8.8), INT8(Q8.0), and UINT8(UQ8.0)
  - Other types may be supported by a vendor

- Eight neural network “layer” nodes:

<table>
<thead>
<tr>
<th>vxActivationLayer</th>
<th>vxConvolutionLayer</th>
<th>vxDeconvolutionLayer</th>
</tr>
</thead>
<tbody>
<tr>
<td>vxFullyConnectedLayer</td>
<td>vxNormalizationLayer</td>
<td>vxPoolingLayer</td>
</tr>
<tr>
<td>vxSoftmaxLayer</td>
<td>vxROIPoolingLayer</td>
<td>...</td>
</tr>
</tbody>
</table>

- Example supported networks
  - Overfeat, Alexnet, GoogLeNet versions (Inception)
  - LSTM, RNN/BiRNN, Faster-RCNN, FCN

- Conformance tests will be up to some “tolerance” in precision
  - To allow for optimizations, e.g., weight compression
OpenVX Export & Import Extension

- Provide a way of exporting and importing pre-verified graphs or other objects
- Divides functionality into “development” and “deployment”

Use Cases
- Embedded systems using fixed graphs, to minimize the amount of code required
- CNN extensions which require the ability to import binary objects
- Safety-critical systems where the OpenVX library does not have a node API
OpenVX Kernel Import Extension

• Provide a way of importing an OpenVX kernel from a vendor binary URL

• Use Cases
  - Offline compilation of DNN from NNEF, ONNX, Tensorflow (.pb), Caffe Model
  - Offline compilation of custom kernels
OpenVX Pipelining Extension

- Provide a way of Pipelining, streaming, and batch processing
- Multiple initiations of a graph with different inputs and outputs
- Increase hardware utilization and throughput

Example SoC, with three independent Heterogeneous compute units

Executes OpenVX Graph
OpenVX / OpenCL Interop Extension

- OpenCL Command Queue
- Application
- OpenVX Runtime
- Copy or export cl_mem buffers into OpenVX data objects
- Enables custom OpenCL acceleration to be used within OpenVX User Kernels
OpenCL Interop Extension

- Provide interop between OpenVX and an OpenCL application/user-kernel
- Efficient data exchange
  - OpenCL data objects can be imported into the OpenVX environment
  - OpenVX data objects can be accessed as OpenCL data objects
  - Fully asynchronous host-device operations are enabled during data exchange
- Features
  - share common cl_context
  - share a set of common in-order cl_command_queue
  - mechanism for an OpenCL application to export cl_mem buffers
  - mechanism for an OpenCL application to reclaim exported cl_mem buffers back
  - mechanism for an OpenCL application/user-kernel to temporarily map OpenVX data objects into cl_mem buffers
  - mechanism to copy between cl_mem buffers and OpenVX data objects
Classifier Extension

• Enable deployment of an image classifier
  - Classification is the problem of identifying category of an observation
  - Feature detection is the problem of identifying whether an image feature is present at every location

• Possible methods:
  - SVM
  - Cascade (Decision Tree)
## Summary

<table>
<thead>
<tr>
<th>Extension Name</th>
<th>OpenVX Versions</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vx_khr_nn</td>
<td>1.2</td>
<td>Neural network extension</td>
</tr>
<tr>
<td>vx_khr_ix</td>
<td>1.2, 1.1</td>
<td>Export and import extension for graphs &amp; objects</td>
</tr>
<tr>
<td>vx_khr_import_kernel</td>
<td>1.2, 1.1</td>
<td>Import kernel from vendor binary</td>
</tr>
<tr>
<td>vx_khr_pipelining</td>
<td>1.2, 1.1</td>
<td>Pipelining, streaming, and batch processing extension</td>
</tr>
<tr>
<td>vx_khr_opencl</td>
<td>1.2, 1.1</td>
<td>OpenCL interop extension</td>
</tr>
<tr>
<td>vx_khr_class</td>
<td>1.2</td>
<td>Classifier extension</td>
</tr>
<tr>
<td>vx_khr_s16</td>
<td>1.1</td>
<td>S16 extension</td>
</tr>
<tr>
<td>vx_khr_tiling</td>
<td>1.0</td>
<td>Tiling extension</td>
</tr>
<tr>
<td>vx_khr_xml</td>
<td>1.0</td>
<td>XML schema extension</td>
</tr>
<tr>
<td>vx_khr_icd</td>
<td>1.0</td>
<td>Installable client driver extension</td>
</tr>
</tbody>
</table>

*provisional*