## SCHEDULE FOR THE DAY

### Morning Session

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:15 – 10:15</td>
<td>Loihi Architecture Overview</td>
</tr>
<tr>
<td>10:15 – 10:25</td>
<td>Break / Hardware Q&amp;A</td>
</tr>
<tr>
<td>10:25 – 10:55</td>
<td>NxSDK Architecture</td>
</tr>
<tr>
<td>11:00 – 11:35</td>
<td>NxNet Intro</td>
</tr>
<tr>
<td></td>
<td>• Add compartments/connections (code, basic behavior)</td>
</tr>
<tr>
<td></td>
<td>• STDP Learning and eligibility traces</td>
</tr>
<tr>
<td></td>
<td>• Kapoho Bay DVS Demo</td>
</tr>
<tr>
<td></td>
<td>• Multi-compartment neurons – <em>time permitting</em></td>
</tr>
<tr>
<td>11:35 – 11:45</td>
<td>Software Q&amp;A</td>
</tr>
</tbody>
</table>

### Afternoon Session

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1:00 – 1:10</td>
<td>NxSDK Overview (quick version)</td>
</tr>
<tr>
<td>1:10 – 2:05</td>
<td>Algorithmic Demos with NxNet</td>
</tr>
<tr>
<td></td>
<td>• Single-Layer Image Classification</td>
</tr>
<tr>
<td></td>
<td>• Solving LASSO w/ Spiking LCA</td>
</tr>
<tr>
<td></td>
<td>• Constraint satisfaction</td>
</tr>
<tr>
<td>2:05 – 2:35</td>
<td>Algorithmic Demos with Nengo</td>
</tr>
<tr>
<td></td>
<td>• Nonlinear oscillator</td>
</tr>
<tr>
<td></td>
<td>• Learning w/ Prescribed Error Sensitivity</td>
</tr>
<tr>
<td></td>
<td>• MNIST classification with Nengo DL</td>
</tr>
<tr>
<td></td>
<td>• Keyword spotting with Nengo DL</td>
</tr>
<tr>
<td>2:35 – 2:50</td>
<td>Graph Search and Multi-Chip Scaling</td>
</tr>
<tr>
<td>2:50 – 3:00</td>
<td>Closing / Q&amp;A</td>
</tr>
</tbody>
</table>
SNN Algorithms Discovery and Development

New Ideas Guided by Neuroscience
- Olfaction-inspired rapid learning
- Dynamic Neural Fields
- SLAM
- Evolutionary search
- Cortical models

Mathematically Formalized
- Locally Competitive Algorithm for LASSO
- Neural Engineering Framework (NEF)
- Stochastic SNNs for solving CSPs
- Parallel graph search
- Phasor associative memories
- Random diffusion walkers

Deep Learning Derived Approaches
- DNN -> SNN conversion
- SNN backpropagation
- Online SNN pseudo-backprop

Competitive Computer Architectures

Machine Learning

Neuroscience
Intel’s Objectives for INRC

1) **Accelerate** research in neuromorphic computing
   By stimulating *algorithms* and *applications* research focusing on Loihi architecture

2) **Quantify** the value of neuromorphic computing today
   Discipline of a quantified approach is critical for progress and mainstream adoption

3) **Inform Loihi’s architectural development**
   Algorithms & application findings provide insights for future silicon revisions

4) **Build an ecosystem** that can provide a market for neuromorphic chips
   Intel hopes to sell chips, eventually... we need customers and a broad user base
Join the Community

Available to members:
- Access to member website
- Project documentation
- Access to GitHub site
- Participation in INRC workshops
INRC Engagement Process

1) Email inrc_interest@intel.com
   We’ll send you our RFP and project proposal template

2) Submit a project proposal
   Tell us what you want to investigate and accomplish with Loihi

3) Execute the INRC participation agreement
   Requires signature of someone who can legally bind your organization

4) Receive Neuromorphic Research Cloud accounts
   You get a private VM on our system + accounts for your team members

5) Request Loihi hardware
   We’ll loan you physical systems when and if you need them...
Loihi Systems

- **Q4 2017**
  - Wolf Mountain
  - Remote Access
  - 4 Loihi/Board

- **Q2 2018**
  - Nahuku
  - Arria10 Expansion Board
  - For cloud & local use
  - 8-32 Loihi/Board

- **Q3 2018**
  - Kapoho Bay
  - 1-2 Loihi
  - DVS interface
  - USB host interface

- **Q2 2019**
  - Pohoiki Springs
  - Remote Access
  - Up to 768 chips
  - (100M neurons)
NAHUKU 32-CHIP PLATFORM

- 32 Loihi chips
- 4 x 4 mesh of chips
- Top and bottom sides
System Architecture

Conventional sensors, actuators, etc. for application demos

Host/FPGA

Loihi

“Super Host” CPU
- Owns the high-level application
- Compilation, visualization, debug, UI

FPGA/Host
- Manages Loihi chips
- Interfaces to outside world

Loihi x86
- System management
- Some custom user code

Loihi Neurocore
- Spiking neural network hardware

Neuromorphic sensors
- DVS camera
- Silicon cochlea

Multi-chip scalability
Objective

efficiently map abstract spiking neural network definitions onto our heterogeneous hierarchical implementation

Architectural Principles

Programmability – Must be accessible in the lingua franca of machine learning (python) and at multiple levels of abstraction.

Simplicity and Modularity – Hardware details are abstracted away. Easy things are easy and complexity is added incrementally. Functionality available through modular interfaces.

Efficiency and Scalability – Additional hardware resource retains efficiency and adds to performance and scalability not to complexity.

Observability – Rich ability to probe and monitor networks as they execute.

Flexibility/Extensibility – Designed to snap into higher level APIs and enable a variety of sensors and actuators.
A module is a complete NxNet defined algorithm (I/O, documentation, etc.)

**Nx SDK Software Architecture**

**Computational Modules**
- LCA
- LSNN
- EPL
- VSA
- TPAM
- SLIC
- CSS
- Path Planning
- DNF
- Astro

**3rd party APIs and Frameworks**
- Nengo
- EONS
- NRP
- ROS
- Tensorflow
- PyNN

**NxNet API**

**Sequential Neural Interfacing Processes (SNIPs)**

**Spiking Neural Network (SNN)**

**NxCompiler**

**NxCore/NxDriver/NxRuntime**
Toolchain

NRC

slurm

Team VMs

ubuntu®

16.04.5 LTS

Python 3.5.2

PIP3

GCC

Loihi Board

sinfo – see what’s available
queue – see what’s running
scancel – stop your job if it gets out of hand

Kapoho Bay

* Other names and brands may be claimed as the property of others
Thank You!

Email inrc_interest@intel.com for more information
This presentation contains the general insights and opinions of Intel Corporation ("Intel"). The information in this presentation is provided for information only and is not to be relied upon for any other purpose than educational. Intel makes no representations or warranties regarding the accuracy or completeness of the information in this presentation. Intel accepts no duty to update this presentation based on more current information. Intel is not liable for any damages, direct or indirect, consequential or otherwise, that may arise, directly or indirectly, from the use or misuse of the information in this presentation.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

No computer system can be absolutely secure. No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document. Intel, the Intel logo, Movidius, Core, and Xeon are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others

Copyright © 2019 Intel Corporation.