Heterogeneous Architectures for Intelligent Vision Systems

Microsystems Design Lab (MDL), Department of Computer Science & Engineering,
The Pennsylvania State University, University Park, PA 16802

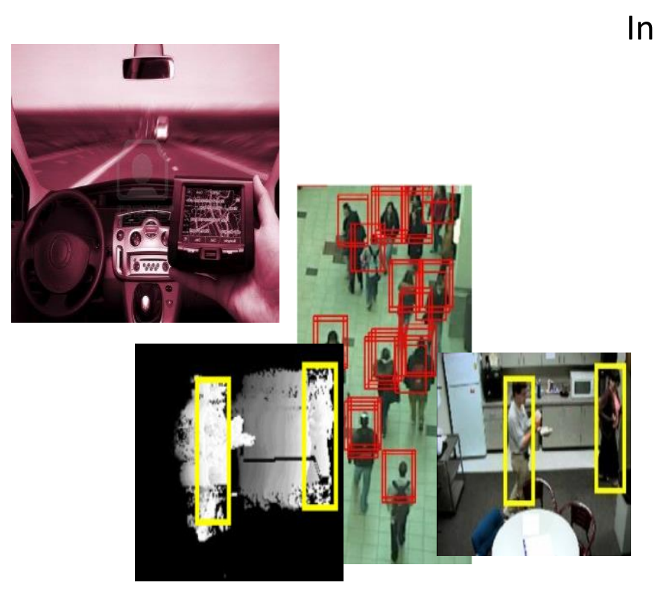
Biologically-inspired Vision

Perceptual systems as efficiently as humans

Recognition Accuracy – 1000s of classes
Performance –milliseconds
Power – < 20 Watts

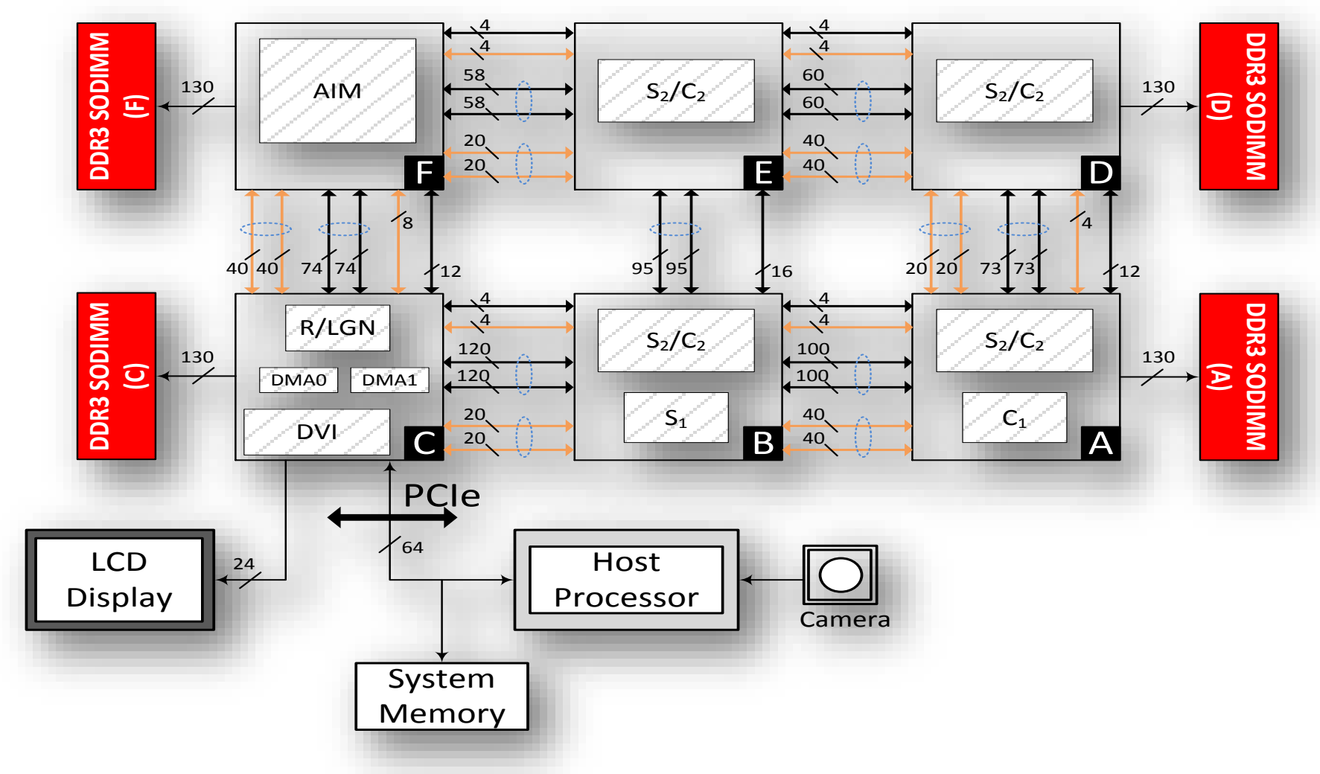
Leverage knowledge of Visual Cortex

Retinal Enhancement
Visual Attention
Inference

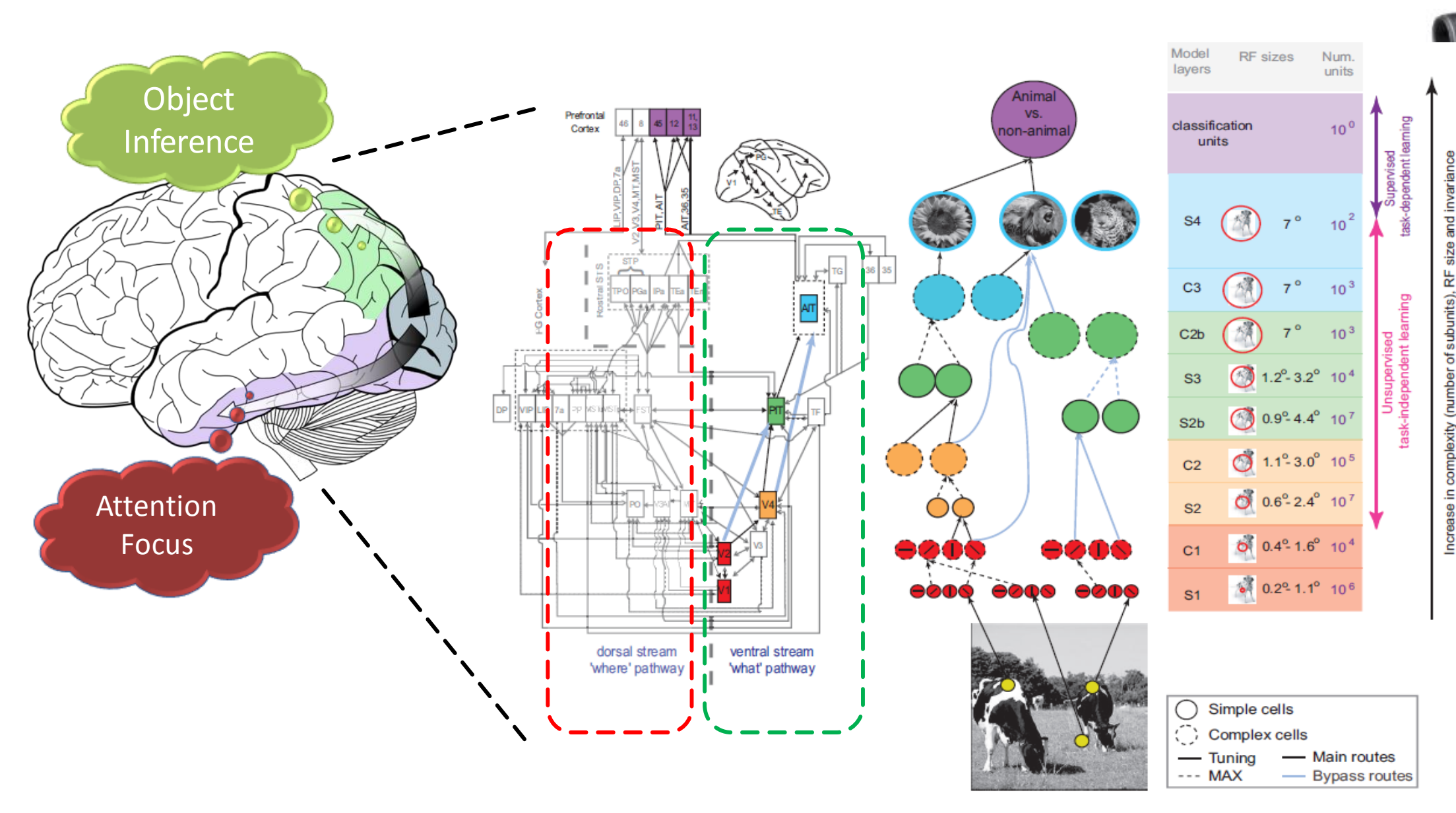


Diverse Application Domains

Vehicle Navigation Assist
First Person Analytics
Biometrics

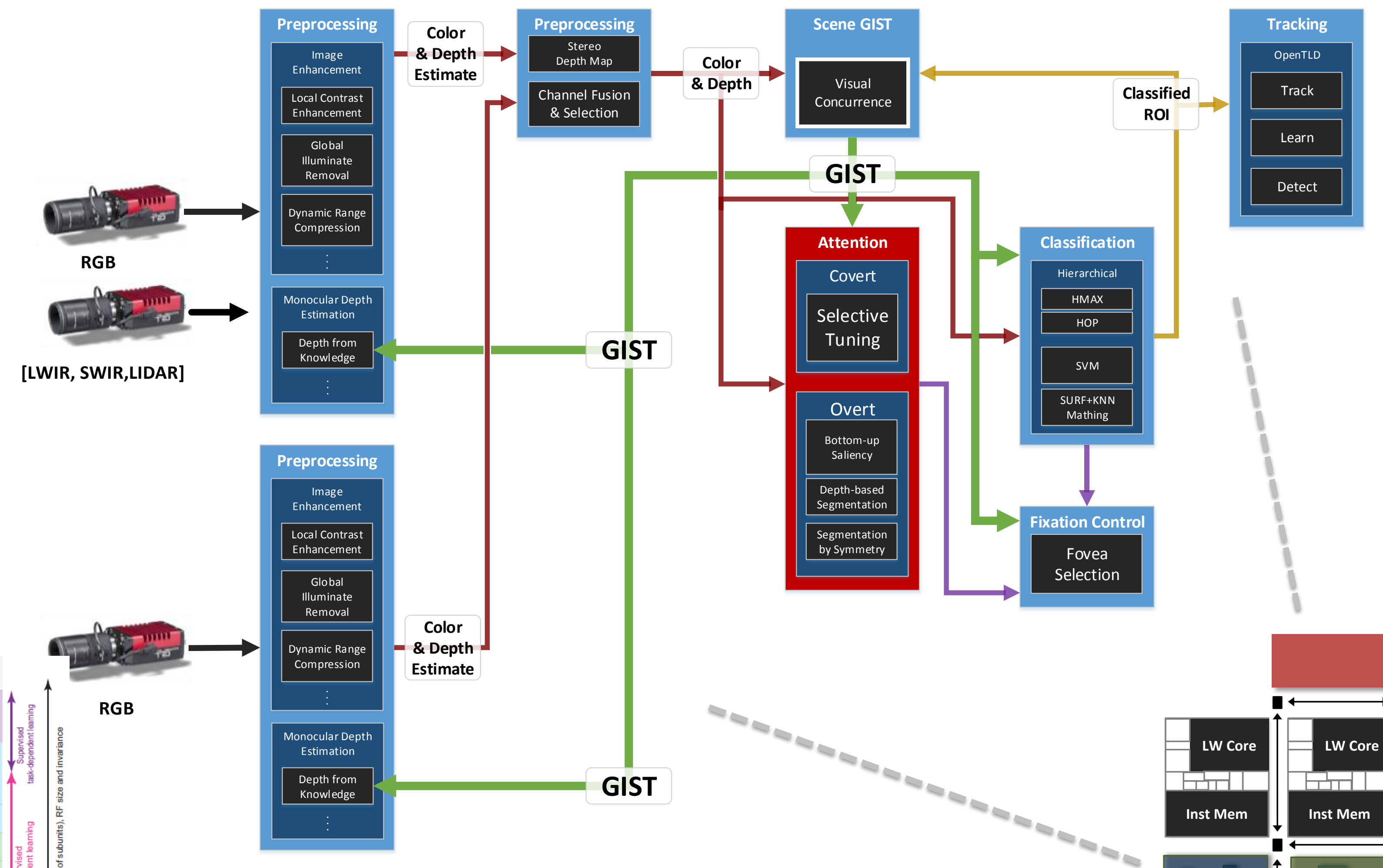


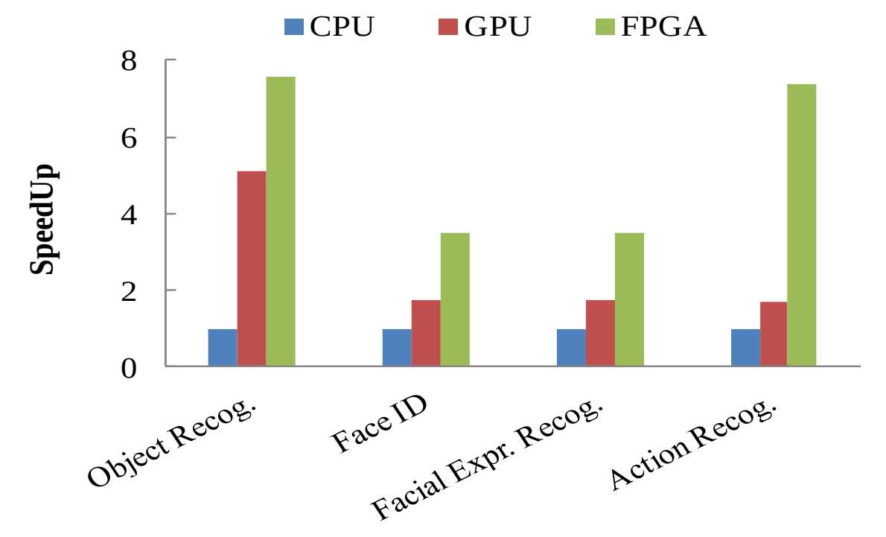
Hardware Architecture


Object Inference
Attention Focus
Prefrontal Cortex
dorsal stream 'where' pathway
ventral stream 'what' pathway
Animal vs. non-animal
Model layers
RF sizes
Num. units
classification units
S4
C3
C2b
S3
S2b
C2
S2
C1
S1
Supervised task-dependent learning
Unsupervised task-independent learning
Increase in complexity (number of subunits), RF size and invariance
Simple cells
Complex cells
Tuning
MAX
Main routes
Bypass routes

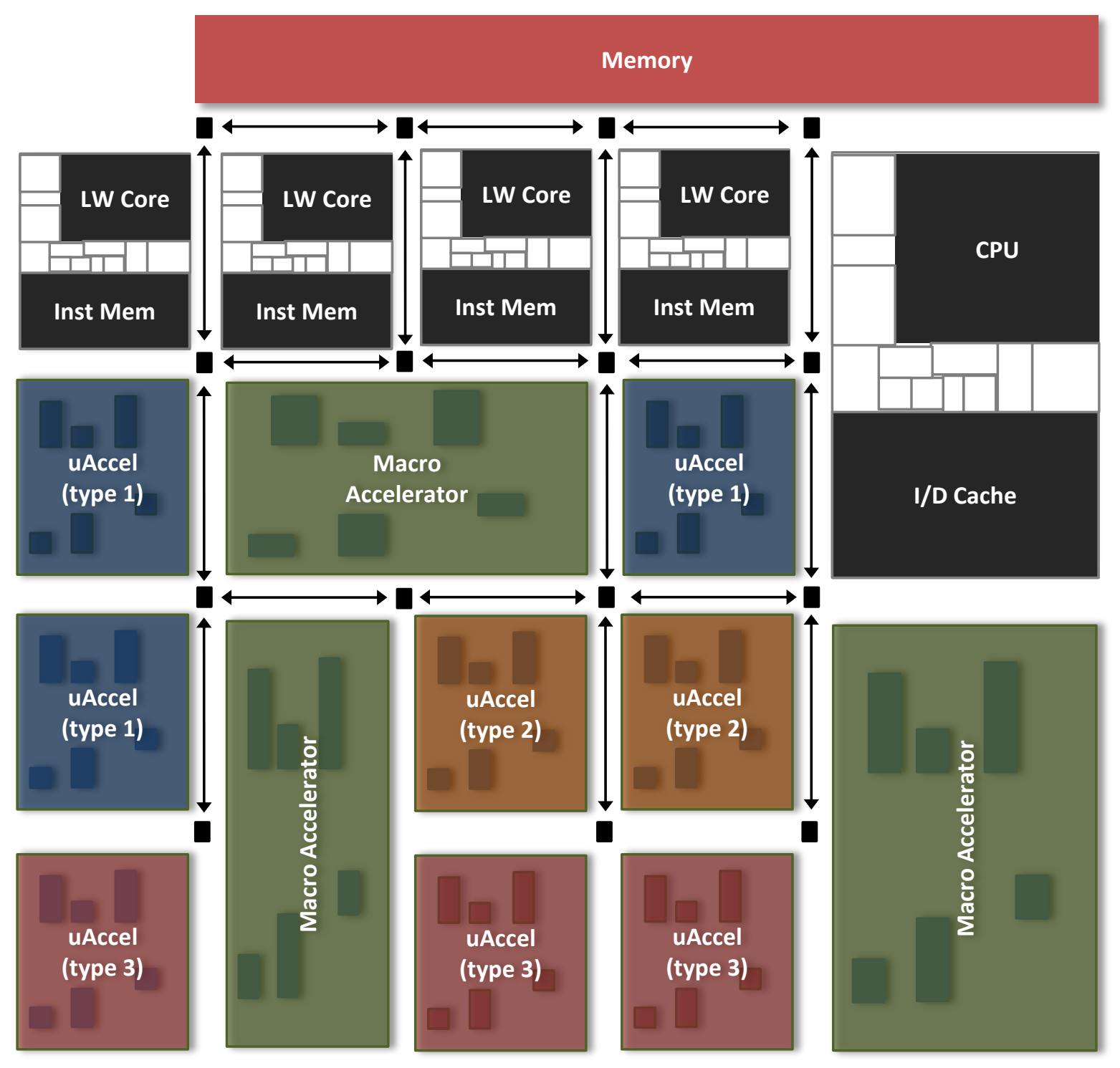
Neuromorphic Vision Models

Visual Perception Pipeline on a SoC





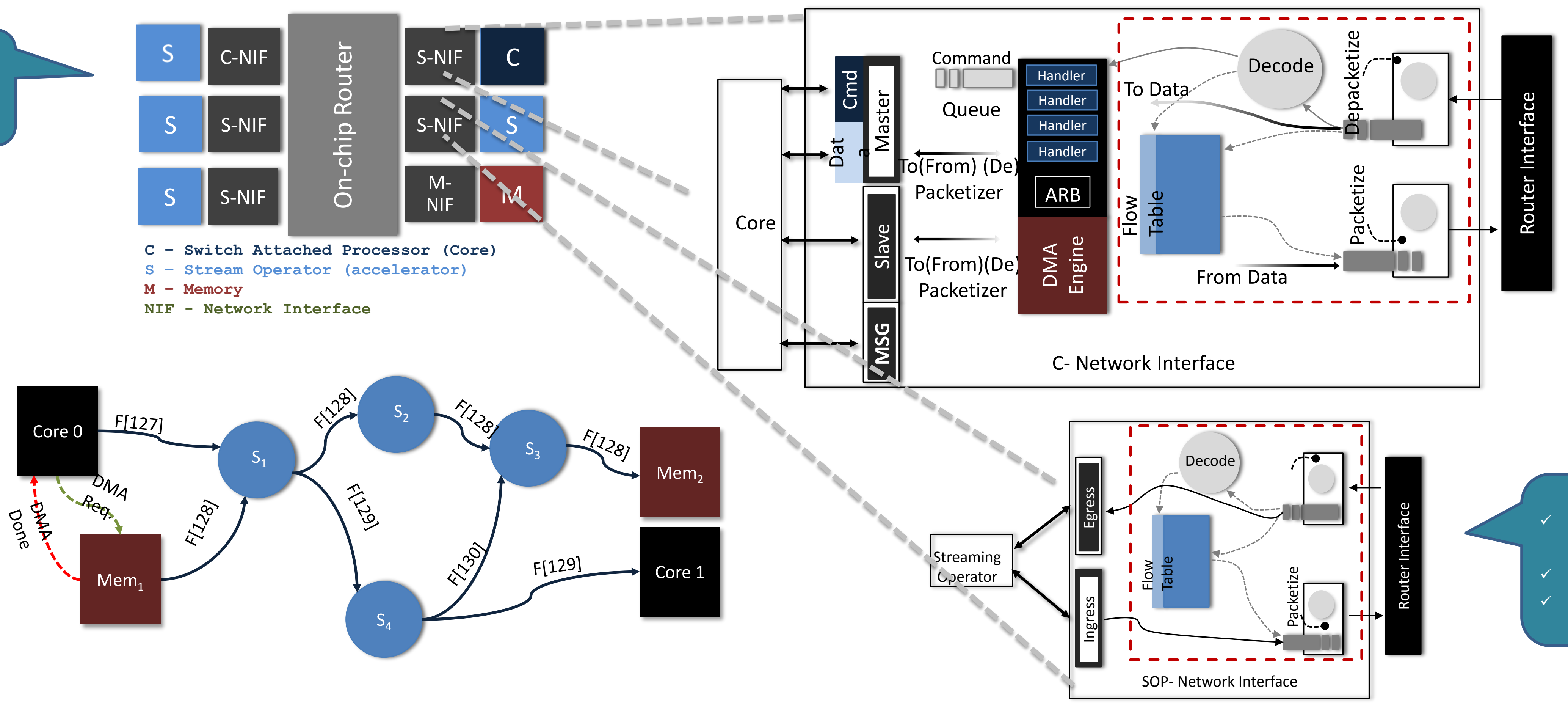
Speedup: up to 7.6X (4.3) compared to CPU (GPU)



Visual Perception Platform – Vortex Network Infrasturcture

- ✓ Accelerators as shared services
- ✓ Native support for dataflow processing
- ✓ Support for low latency control-flow
- ✓ Accelerator composition and configurability

- ✓ Streaming Operators accelerate operations such as 2-d Convolution, Euclidean distance calculation
- ✓ On data arrival from network, Flow id is resolved to Opcode, which determines the mode of operation of accelerator and is forwarded to accelerator
- ✓ Accelerator output data is packetized and Next Hop is resolved based on Flow id
- ✓ Header is updated and data injected into network

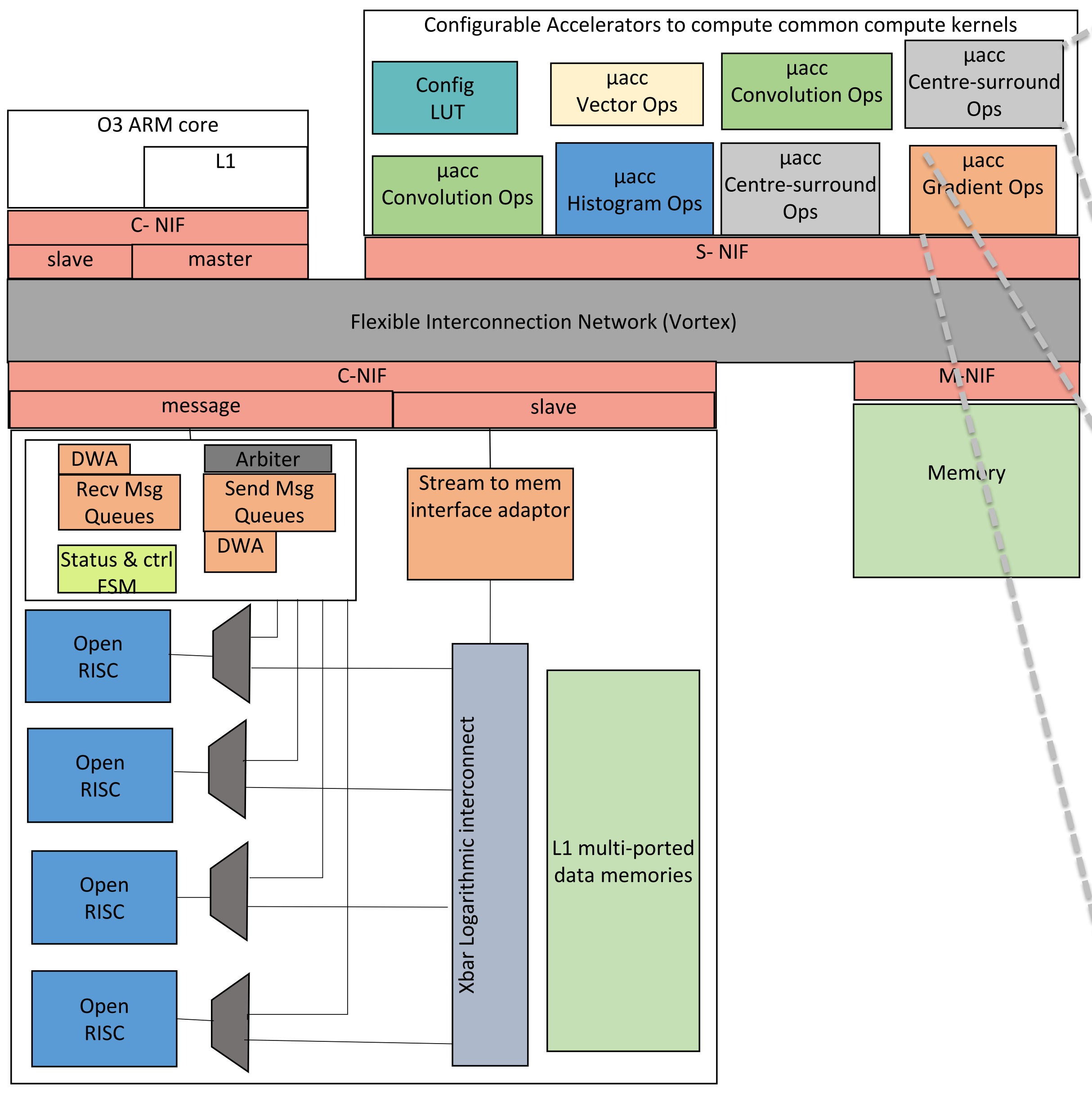


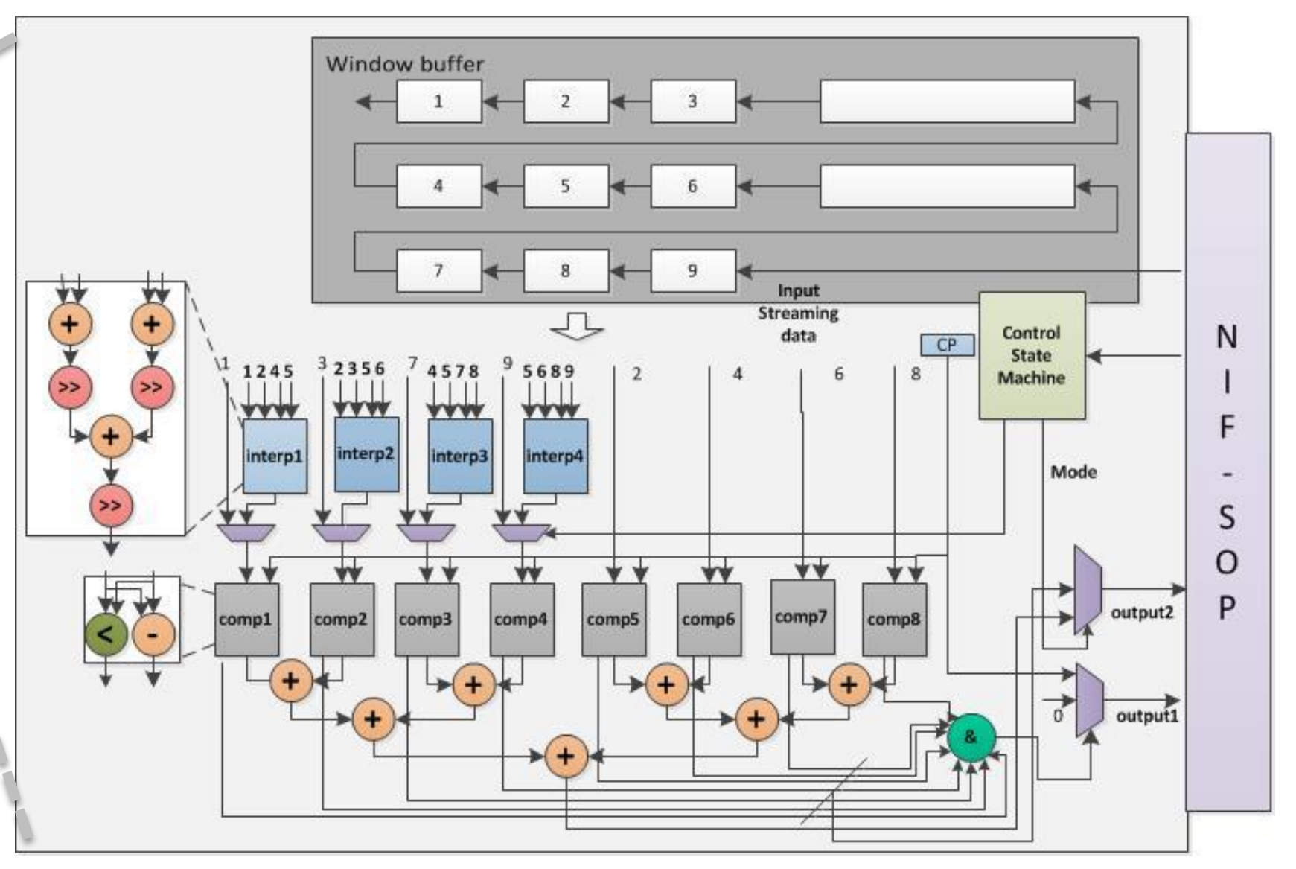
C – Switch Attached Processor (Core)
S – Stream Operator (accelerator)
M – Memory
NIF - Network Interface

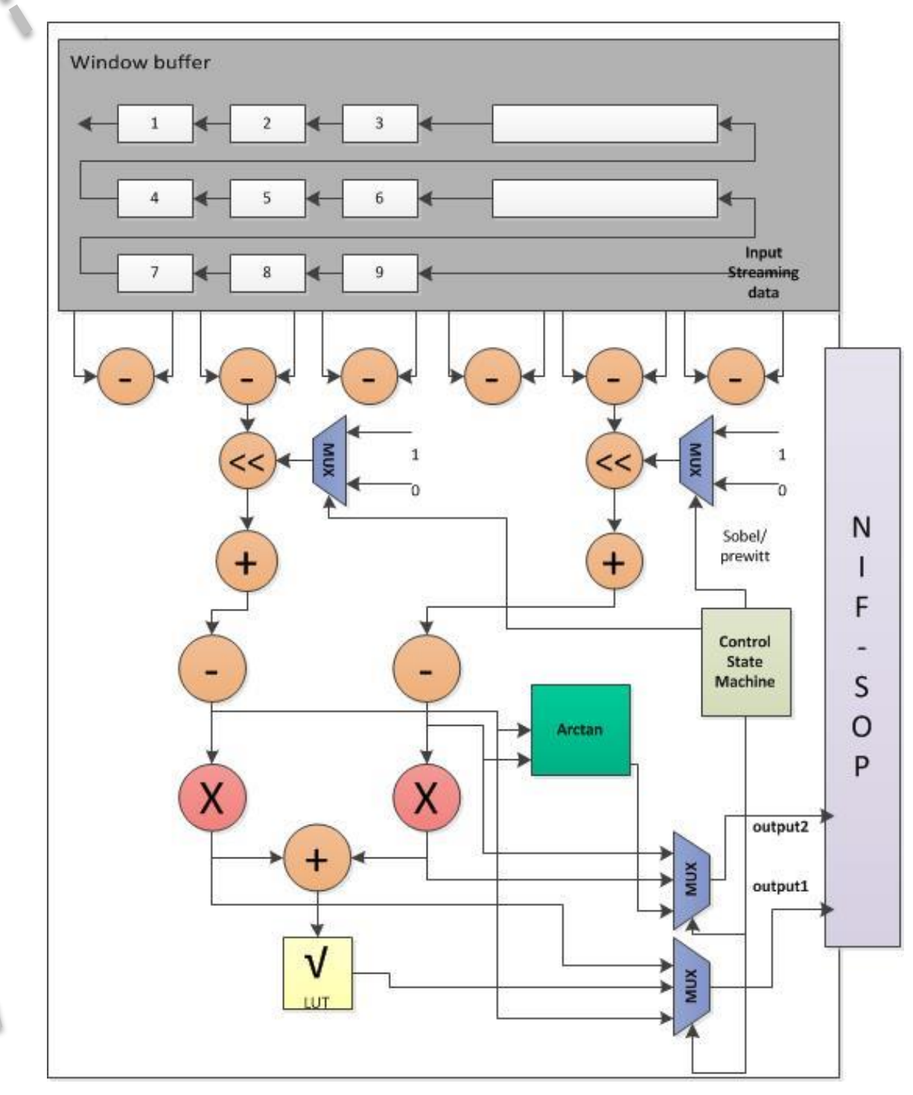
- ✓ Integrated network awareness using data flows
- ✓ Flow = sequence of operations which can be mapped to a node (core or SOP).
- ✓ Flow id represents a unique sequence of operations; Next hop of data flow at each node determined using flow id embedded in each packet by referencing a flow table
- ✓ Efficient direct streaming data transfer between nodes of a data flow thereby “cascading” operations
- ✓ Resource sharing between multiple data flows
- ✓ High aggregate bandwidth to ensure multiple data flows are processed concurrently

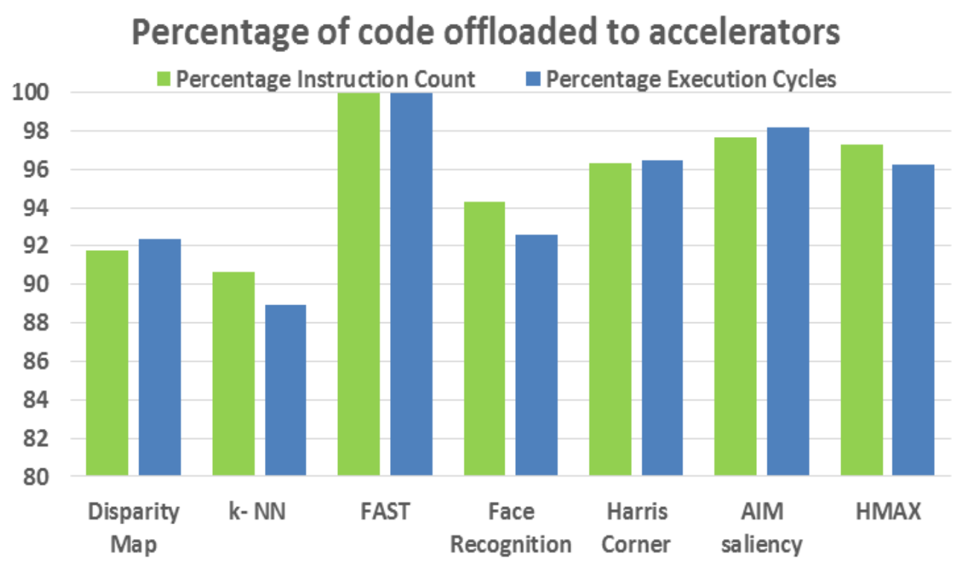
- ✓ Core initiates R/W transactions through master interface’s FIFO-like handshaking or DMA-like transactions between local memory space and remote address space accessible through slave interface
- ✓ Message interface allows to send light weight messages to communicate with other nodes.
- ✓ Multiple outstanding non-blocking operations managed by NIF, each with a different flow-id

Heterogeneous Multi-core Accelerator for Visual Perception (HMAP)



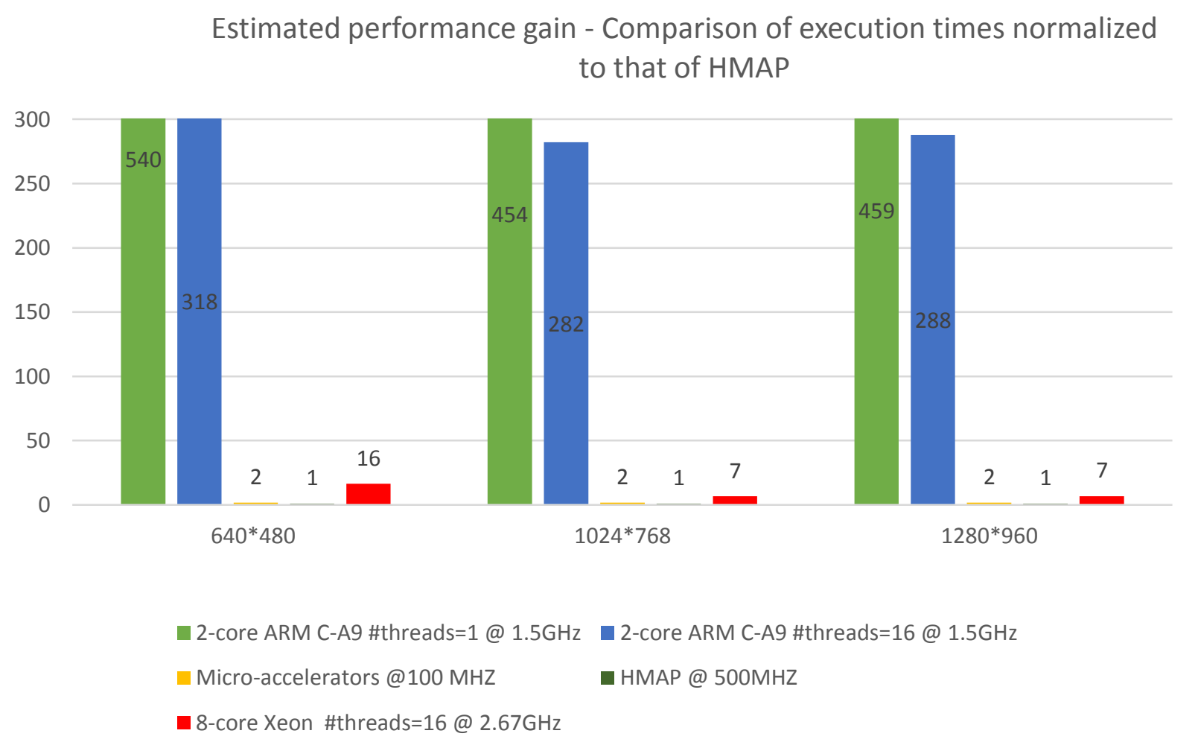






- ✓ Can we achieve performance gain and energy savings comparable to that of an ASIC?
- ✓ What are sources of performance gain and energy?
- ✓ How does Configurability compare with customization?

- ✓ Complex cores to initiate transactions
- ✓ Low power OpenRISC core cluster to extract coarse-grain parallelism
- ✓ Configurable micro-accelerators that can be cascaded to compute common kernels from a range of algorithms with custom storage, routing and DMA memory access
- ✓ Efficient data and control flow that facilitates stream direct data transfers between accelerators that offload computation from openRISC core cluster or complex core
- ✓ Support for multiple transactions at each node


Estimated performance gain - Comparison of execution times normalized to that of HMAP
640*480
1024*768
1280*960
2-core ARM C-A9 #threads=1 @ 1.5GHz
2-core ARM C-A9 #threads=16 @ 1.5GHz
Micro-accelerators @100 MHZ
HMAP @ 500MHZ
8-core Xeon #threads=16 @ 2.67GHz