A Dataflow Processing Chip for Training Deep Neural Networks

Dr. Chris Nicol
Chief Technology Officer

Wave Computing

Wave Computing Profile

Founded in 2010

- Tallwood Venture Capital
- Southern Cross Venture Partners

Headquartered in Campbell, CA

- World class team of 53 dataflow, data science, and systems experts
- 60+ patents

Invented Dataflow Processing Unit (DPU) architecture to accelerate deep learning training by up to 1000x

- Coarse Grain Reconfigurable Array (CGRA) Architecture
- Static scheduling of data flow graphs onto massive array of processors

Now accepting qualified customers for Early Access Program

Wave Computing

Challenges of Machine Learning

Extended training time due to increasing size of datasets

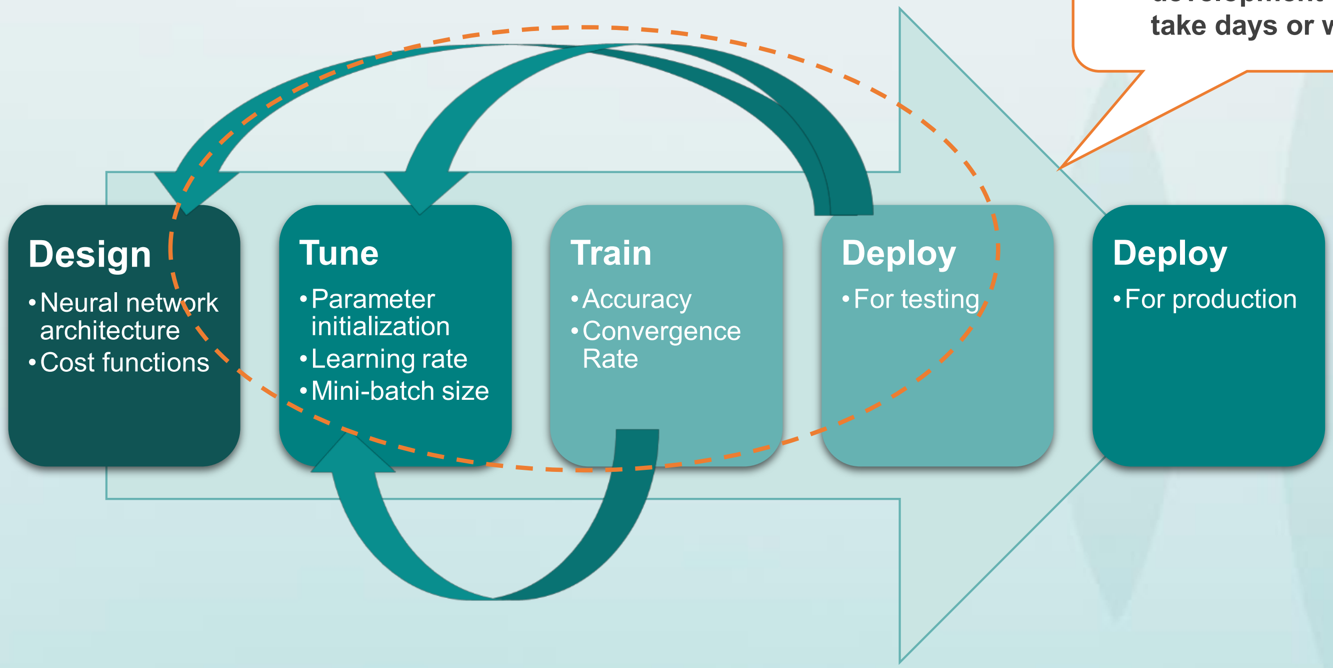
- Weeks to tune and train typical deep learning models

Hardware for accelerating ML was created for other applications

- GPUs for graphics, FPGA's for RTL emulation

Data coming in "from the edge" is growing faster than the datacenter can accommodate/use it…

➢ **Problem:** **Model development times can take days or weeks**



Problems with Existing Solutions

- **Co-processors must wait on the CPU for instructions**
- **This limits performance and reduces efficiency and scalability**
- **Restricts embedded use cases to inferencing-only**

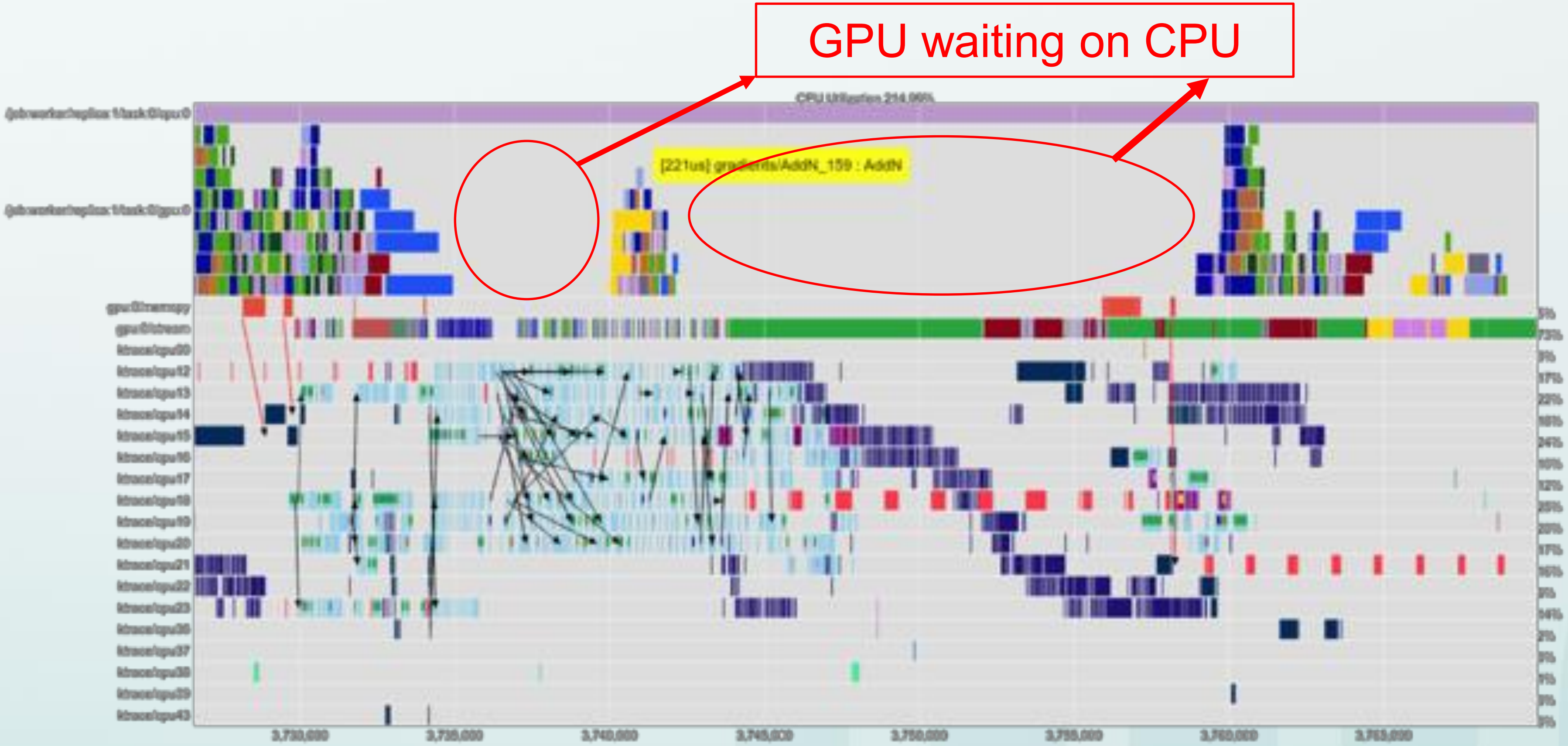


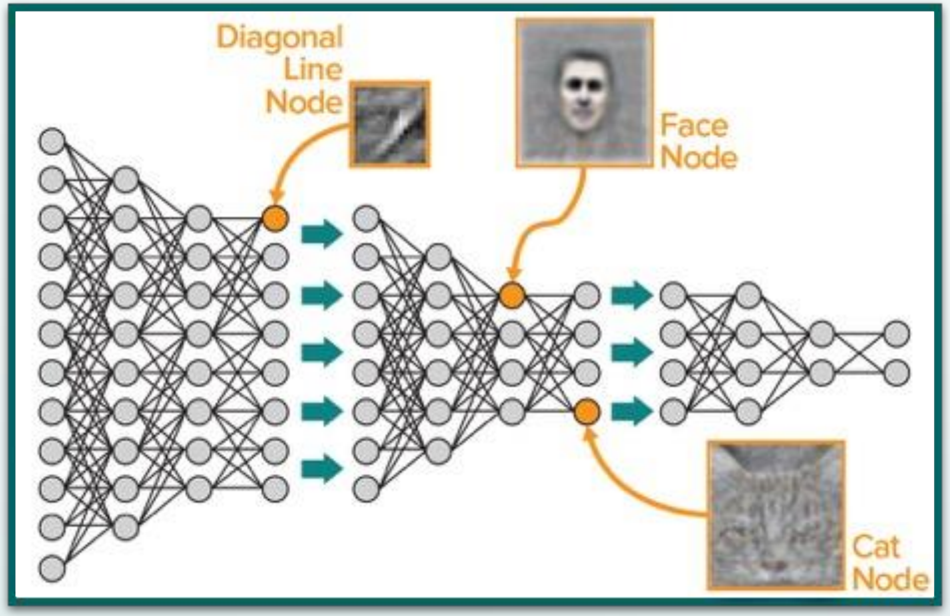
Figure 13: EEG visualization of Inception training showing CPU and GPU activity.

Source: Google; http://download.tensorflow.org/paper/whitepaper2015.pdf

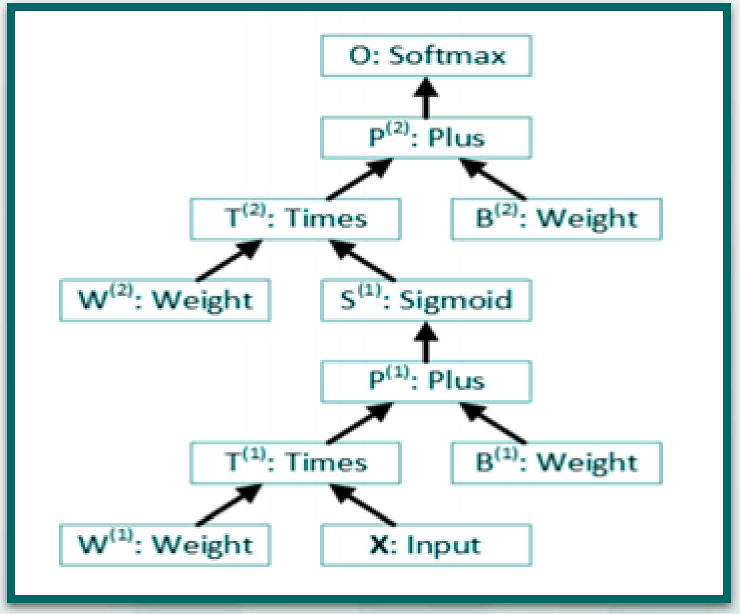
Wave Computing

Wave Dataflow Processor is Ideal for Deep Learning

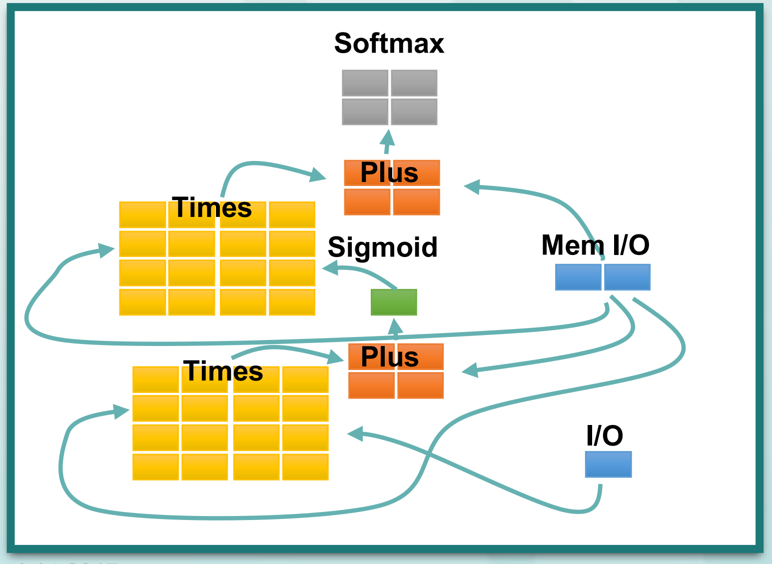
Deep Learning Networks are Dataflow Graphs



Programmed on Deep Learning Software

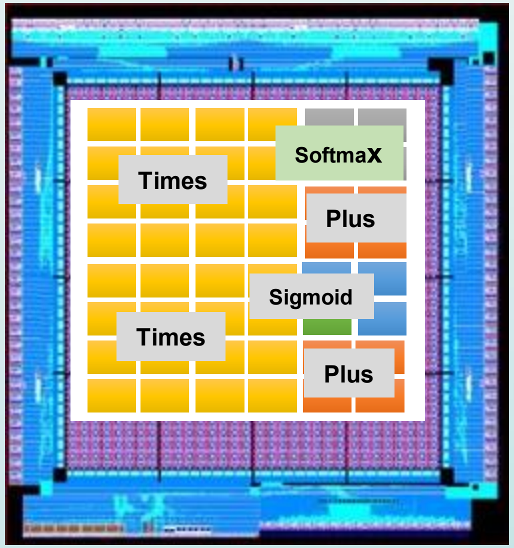


Assemble Dataflow Graph at Runtime



WaveFlow Agent Library

Run on Wave Dataflow Processor



Wave Dataflow Processor

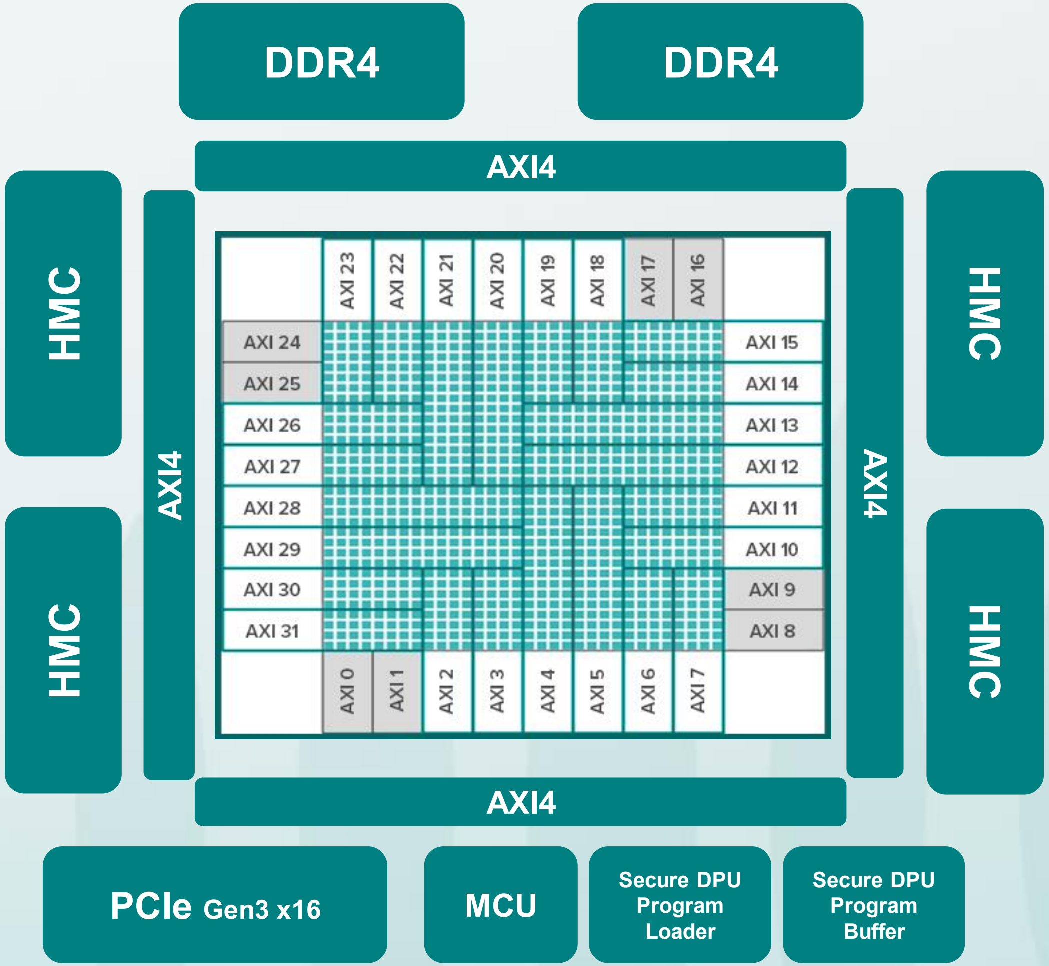
Wave Computing

Wave Dataflow Processing Unit

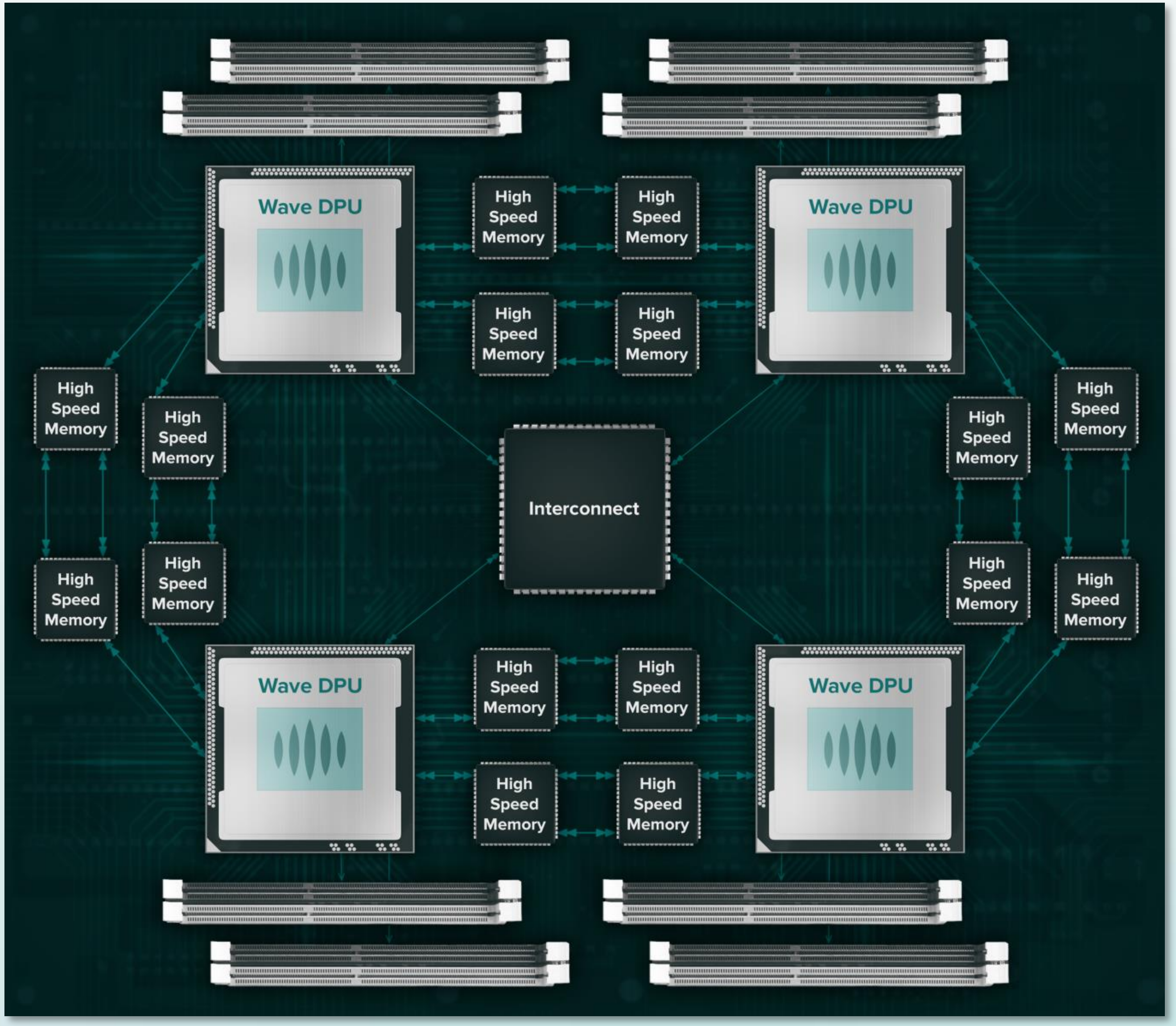
Chip Characteristics & Design Features

16ff CMOS Process Node	16K Processors, 8192 DPU Arithmetic Units	Self-Timed, MPP Synchronization
181 Peak Tera-Ops, 7.25 Tera Bytes/sec Bisection Bandwidth	16 MB Distributed Data Memory	8 MB Distributed Instruction Memory
1.71 TB/s I/O Bandwidth 4096 Programmable FIFOs	270 GB/s Peak Memory Bandwidth	2048 outstanding memory requests
4 Billion 16-Byte Random Access Transfers / sec	4 Hybrid Memory Cube Interfaces	2 DDR4 Interfaces
PCIe Gen3 16-Lane Host interface	32-b Andes N9 MCU	1 MB Program Store for Paging
Hardware Engine for Fast Loading of AES Encrypted Programs	Up to 32 Programmable dynamic reconfiguration zones	Variable Fabric Dimensions (User Programmable at Boot)

- Clock-less CGRA is robust to Process, Voltage & Temperature.
- Distributed memory architecture for parallel processing
- Optimized for data flow graph execution
- DMA-driven architecture – overlapping I/O and computation



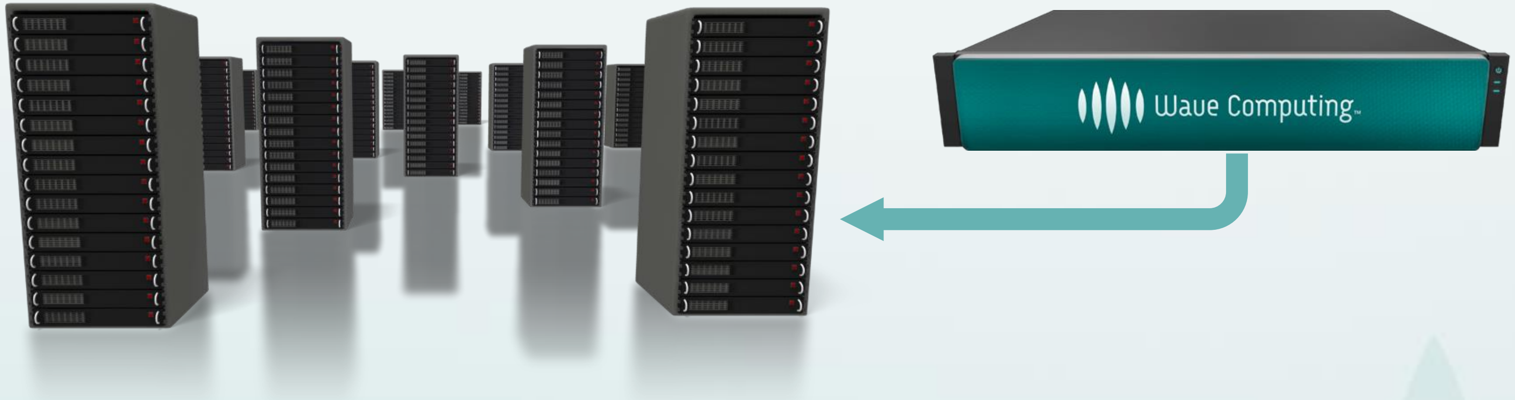
Wave Current Generation DPU Board

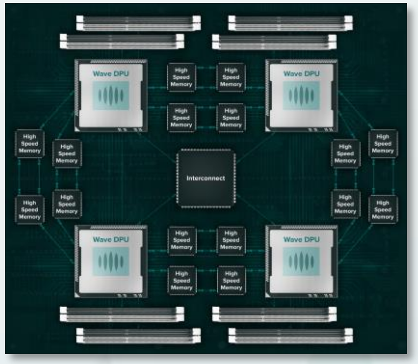


Key DPU Board Features

- 65,536 CGRA Processing Elements
- 4 Wave DPU chips per board
- Modular, flexible design
  - Multiple DPU boards per Wave Compute Appliance
- Off-the-shelf components
  - 32GB of ultra high-speed DRAM
  - 512GB of DDR4 DRAM
  - FPGA for high-speed board-to-board communication

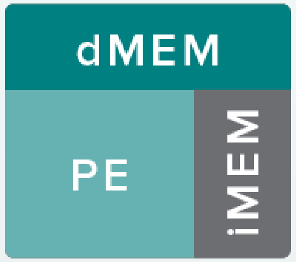
Wave's Solution: Dataflow Computer for Deep Learning





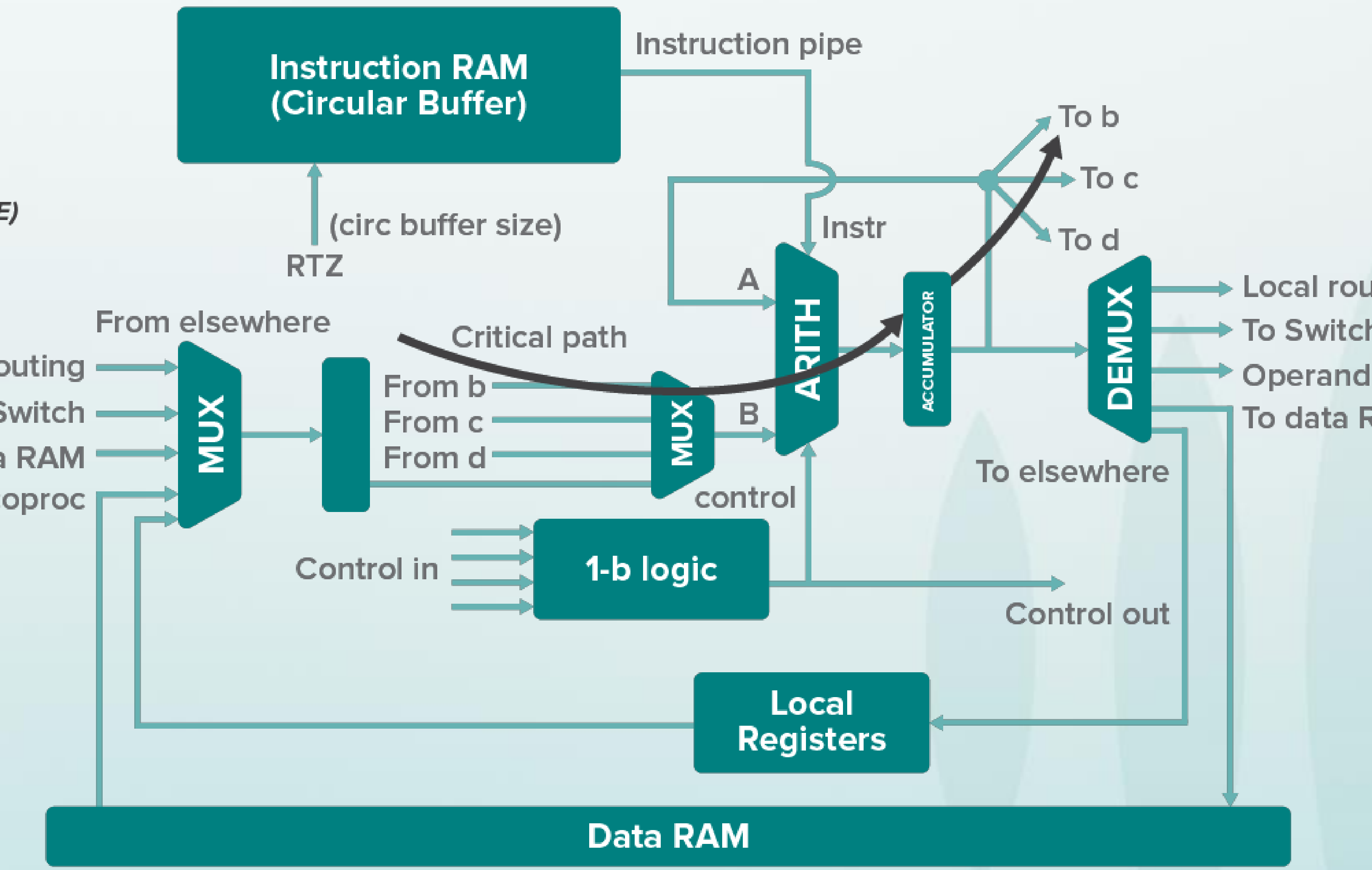
- Best-in-class, highly scalable deep learning training and inference
- More than orders of magnitude better compute-power efficiency
- Plug-and-play node in a datacenter network -- Big Data – Hadoop, Yarn, Spark, Kafka
- Native support of Google TensorFlow (initially)

Dataflow Processing Element (PE)

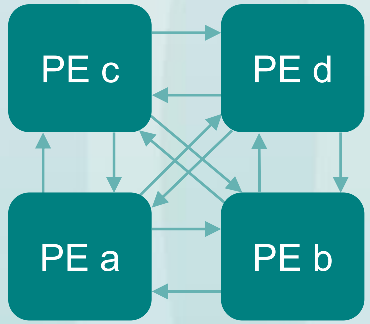


Processing Element (PE)

Pipelined 256-entry Instruction RAM /w ECC

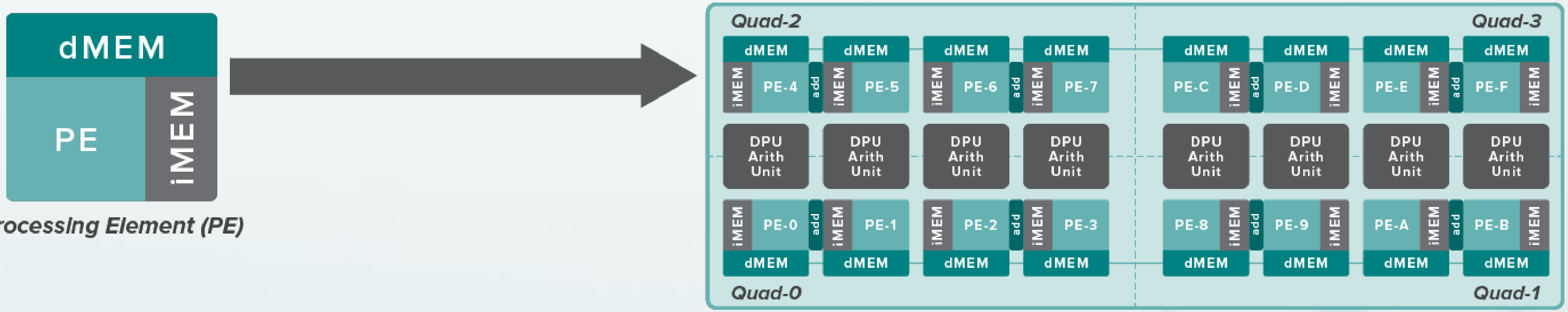


Pipelined 1KB Single Port Data RAM /w BIST & ECC



Quad of PEs are fully connected

Cluster of 16 Dataflow PEs



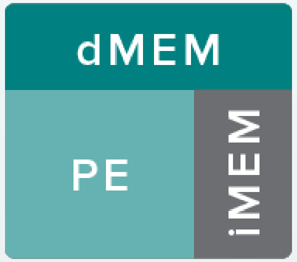
Processing Element (PE)

Cluster (16 PE's)

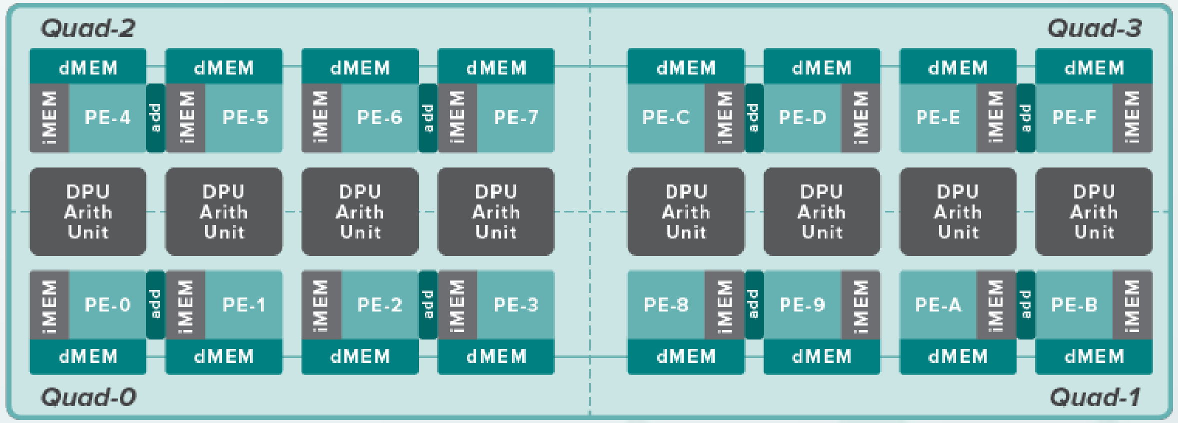
- *16 Processor CLUSTER: a full custom tiled GDSII block*
- **Fully-Connected PE Quads with fan-out**
- **8 DPU Arithmetic Units**
  - Per-cycle grouping into 8, 16, 24, 32, 64-b Operations
  - Pipelined MAC Units with (un)Signed Saturation
  - Support for floating point emulation
  - Barrel Shifter, Bit Processor
  - SIMD and MIMD instruction classes
  - Data driven
- ***16KB Data RAM***
- ***16 Instruction RAMs***
- ***Full custom semi-static digital circuits***
- ***Robust PVT insensitive operation***
  - Scalable to low voltages
  - No global signals, no global clocks

Wave Computing

Hybrid CGRA Architecture



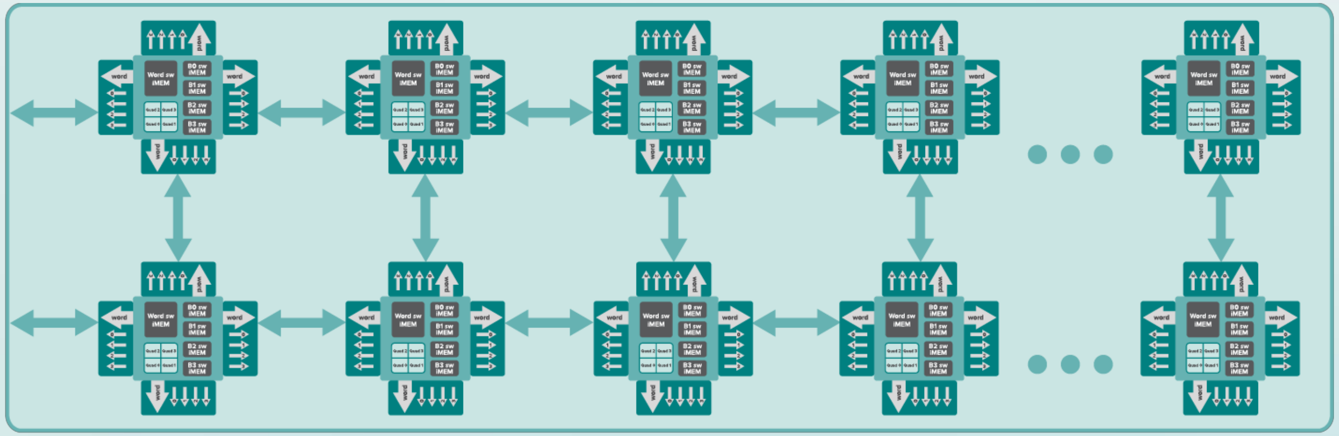
Processing Element (PE)



Cluster (16 PE's)

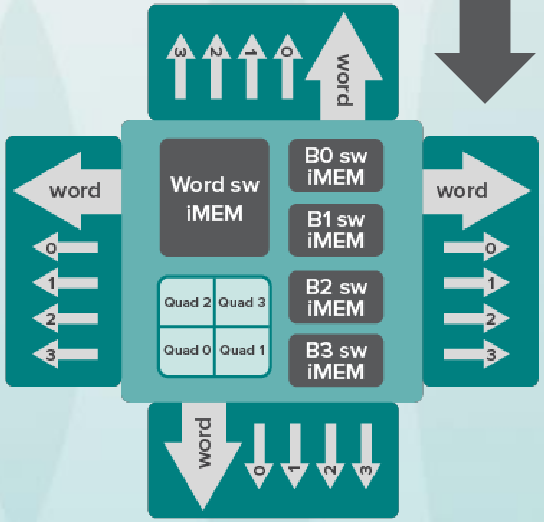
Each cluster has a pipelined instruction-driven word-level switch

Word switch supports fan-out and fan-in



2-D Connected Hybrid CGRA

"valid" and "invalid" data in the switch enables fan-in

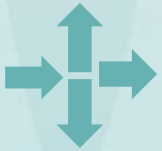


Each cluster has a 4 independent pipelined instruction driven byte-switches

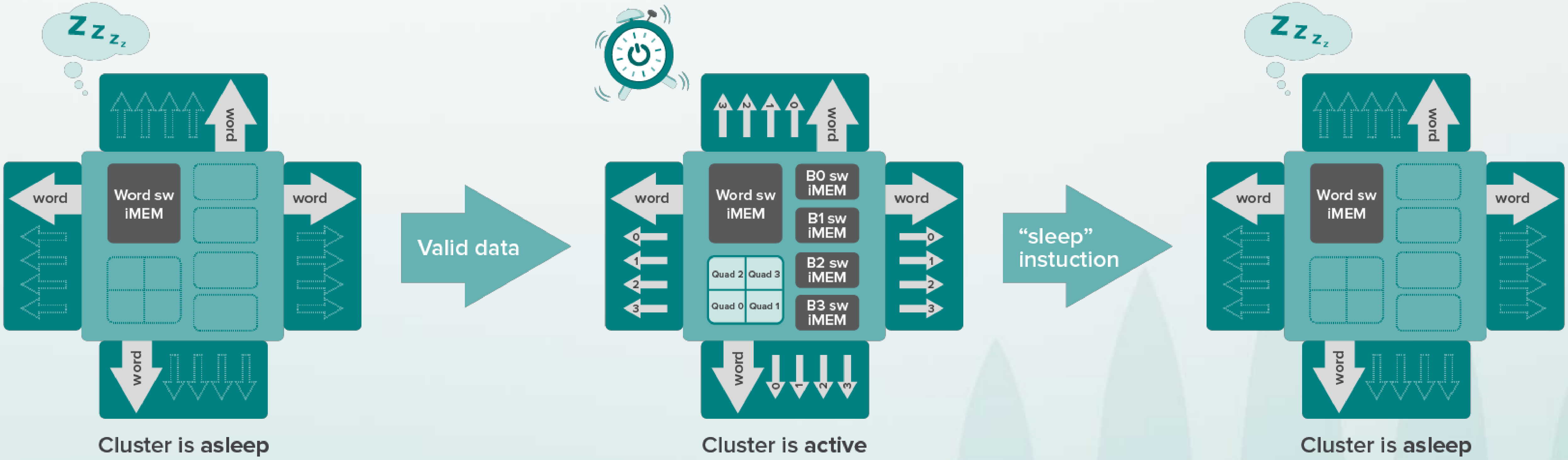
fan-in

All switches have registers for Router use to avoid congestion

fan-out



Data-Driven Power Management



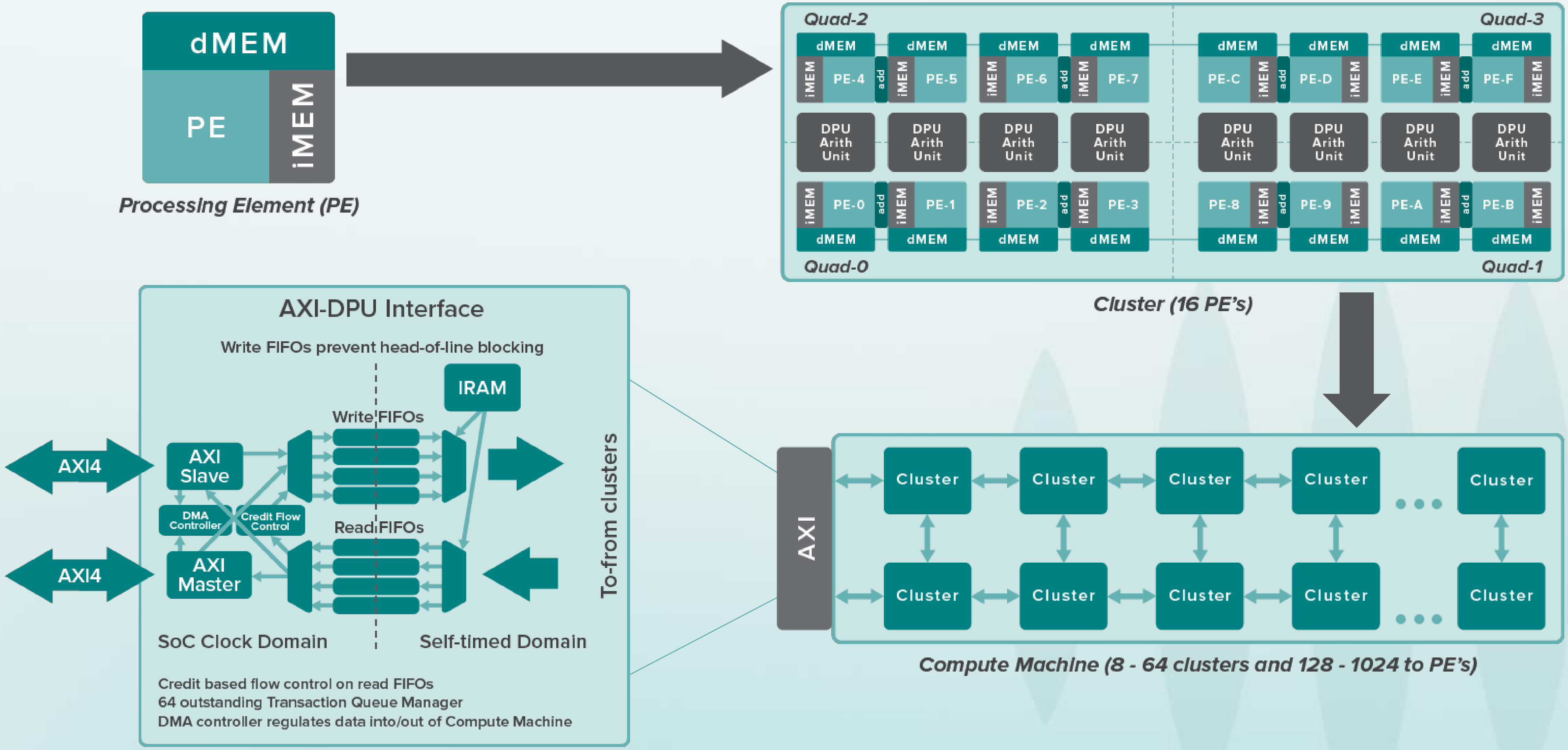
From Asleep to Active

- Word switch fabric remains active
- ***If*** valid data arrives at switch input ***AND*** switch executes instruction to send data to one of Quads ***THEN*** wake up PEs
- Copy PC from word switch to PE and byte switch iRAMs
- Send the incoming data to the PEs

From Active to Asleep

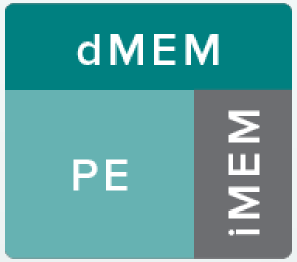
- A PE executes a "sleep" instruction
- All PE & byte switch execution is suspended
- PE can opt for fast wakeup or slow wakeup (deep sleep with lower power)

Compute Machine with AXI4 Interfaces

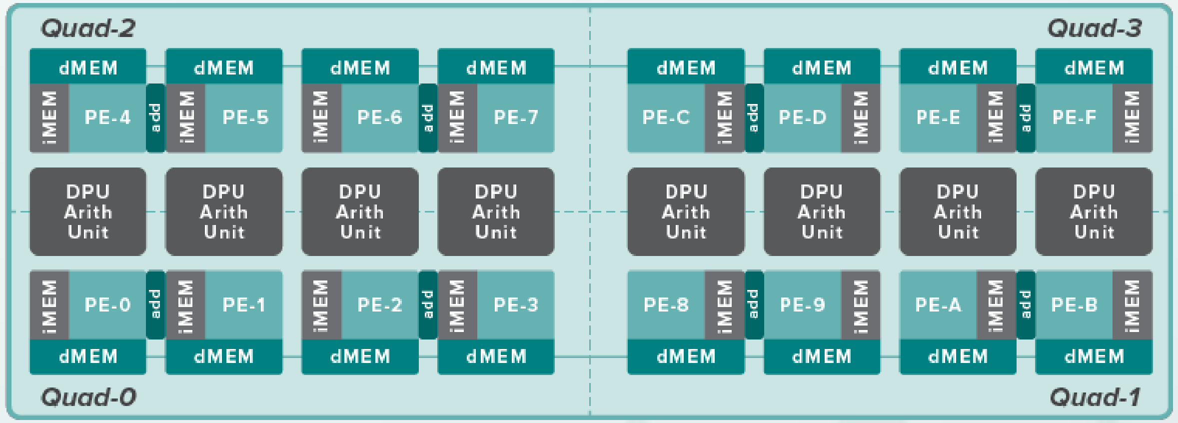


Wave Computing

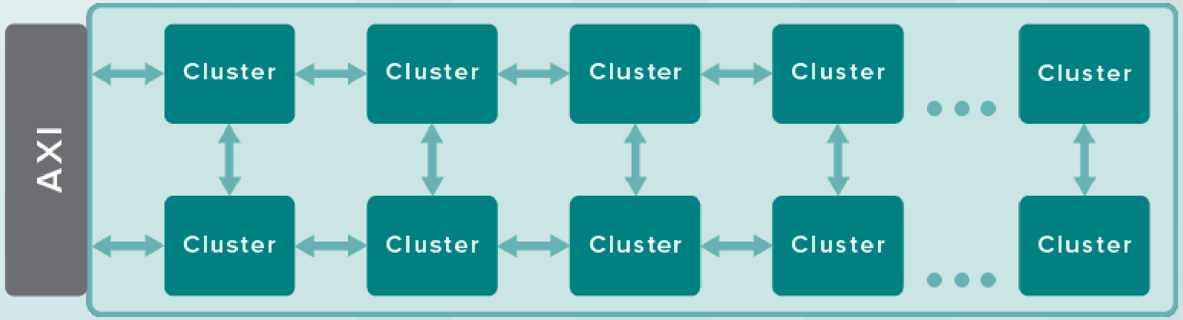
Wave DPU Hierarchy



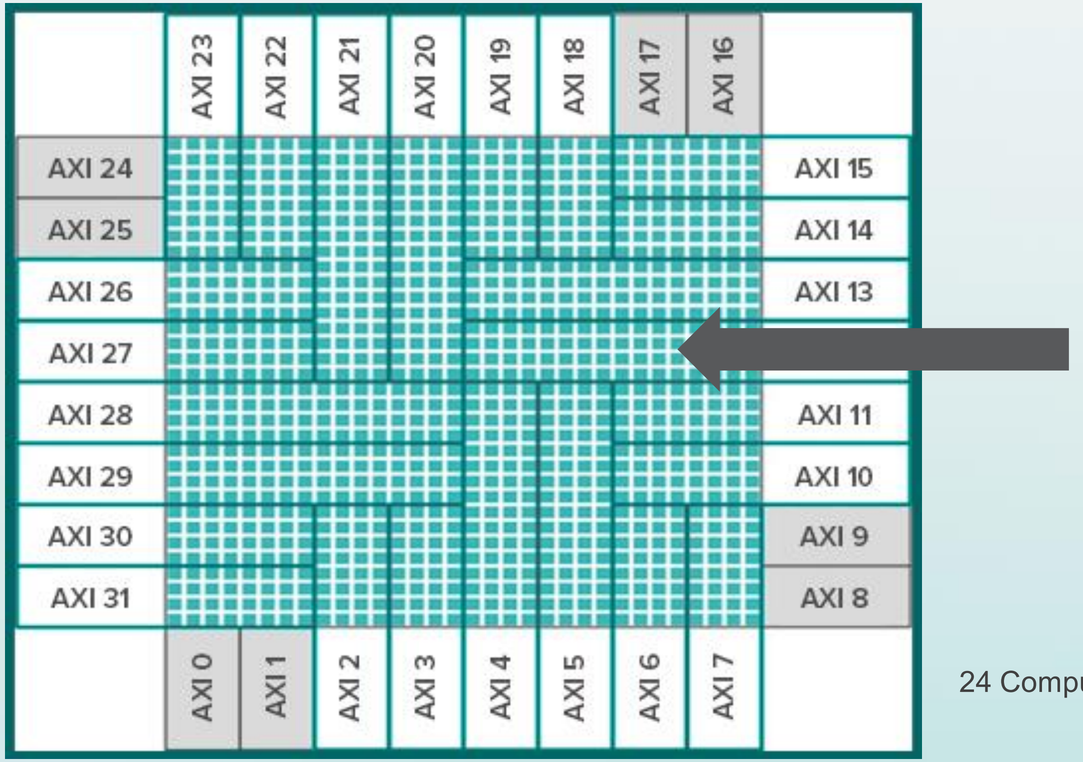
Processing Element (PE)



Cluster (16 PE's)

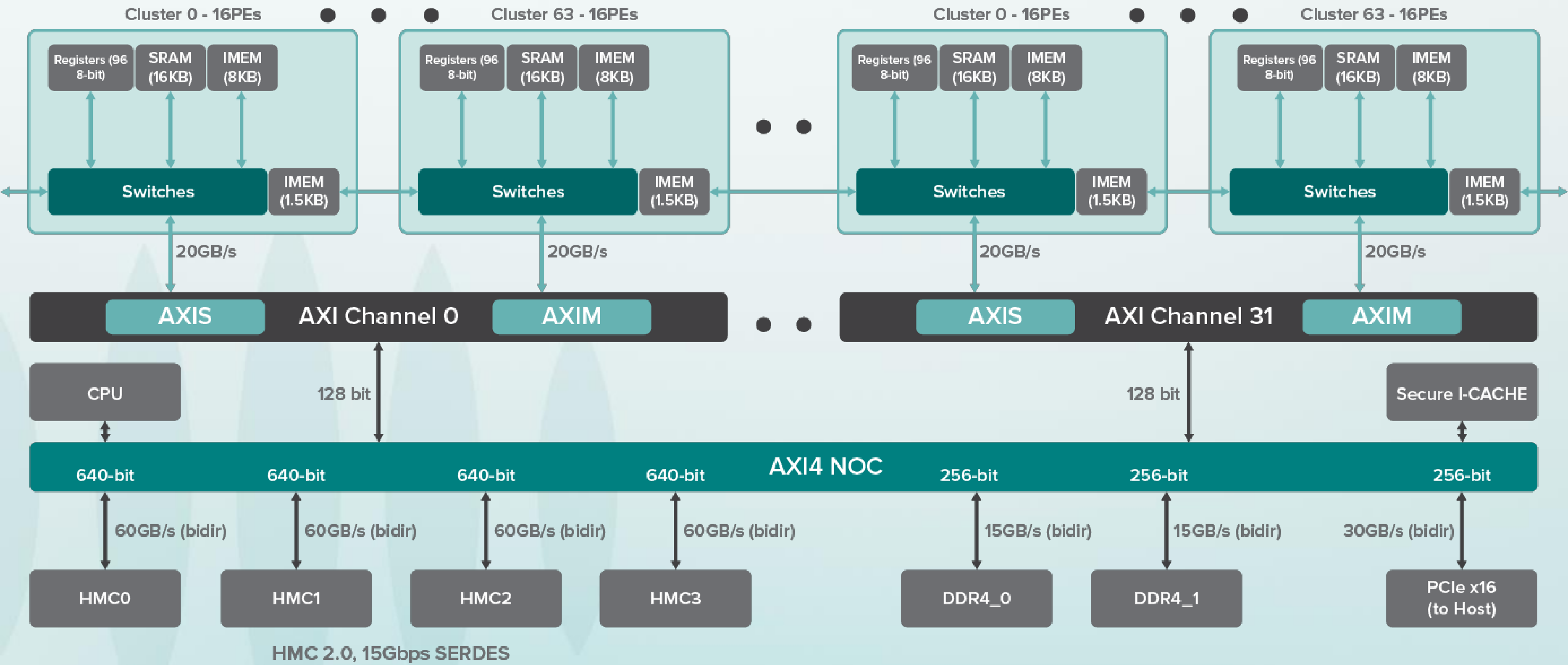


Compute Machine (8 - 64 clusters and 128 - 1024 to PE's)

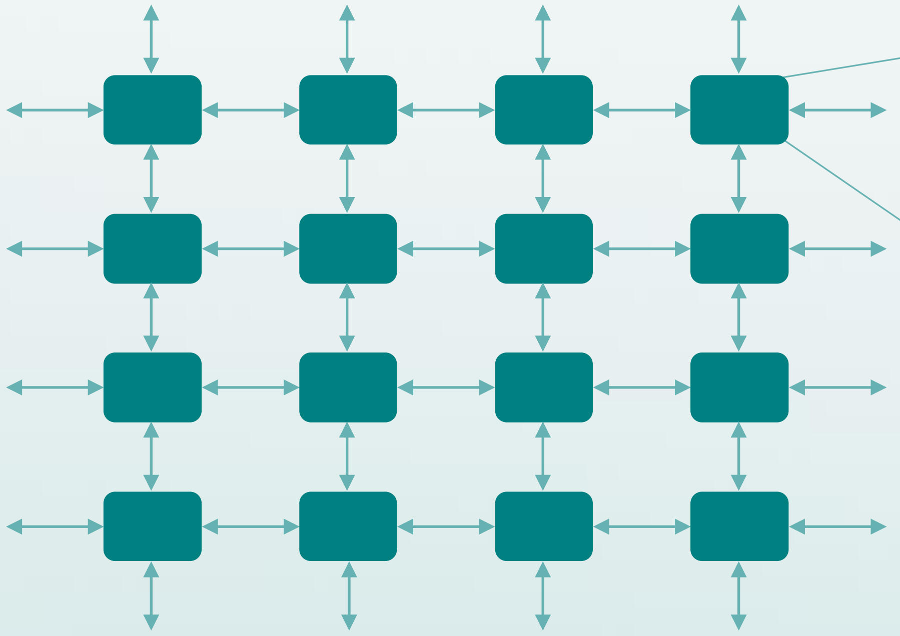


24 Compute Machines

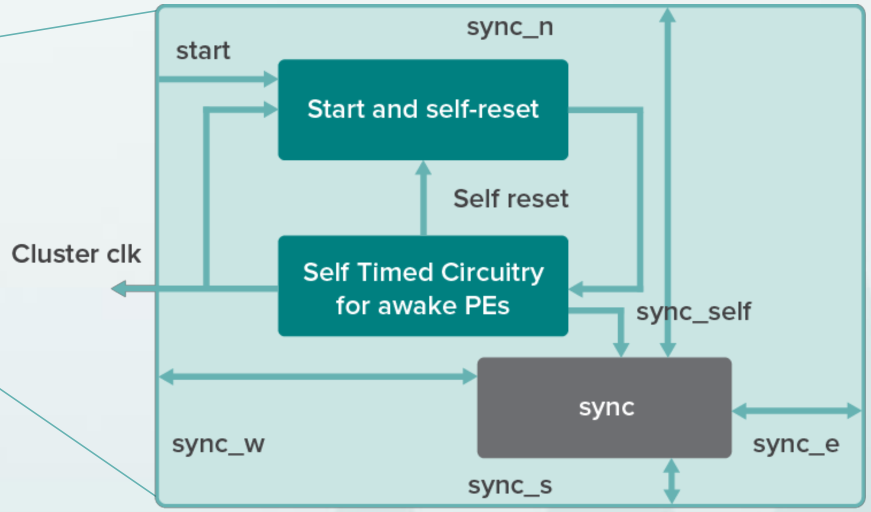
Wave DPU Memory Hierarchy



6-10 GHz Auto-Calibrated Clock Distribution



Clock Distribution and Generation Network across entire Fabric

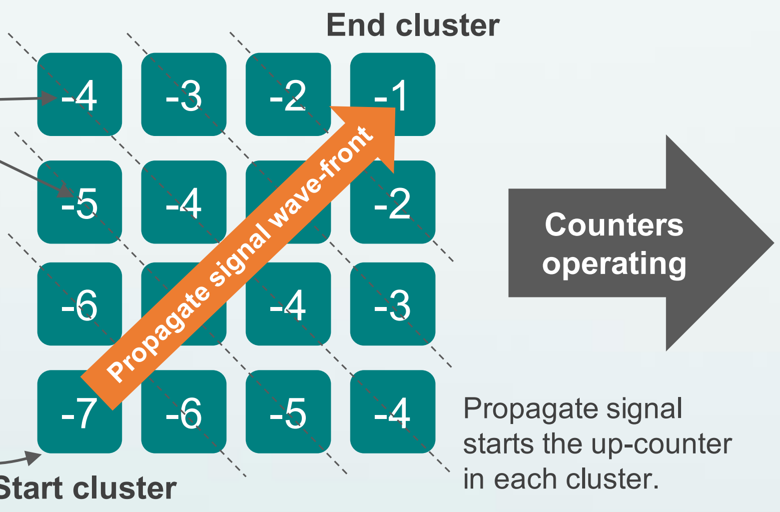


- Clock skew and jitter limit cycle time with traditional clock distribution
- Self-timed “done” signal from PEs if they are awake. Programmable tuning of margin.
- Synchronized with neighboring Clusters to minimize skew
- 1-sigma local mismatch ~1.3ps and global + local mismatch ~6ps at 140ps cycle time

Wave Computing

Reset, Configuration Modes

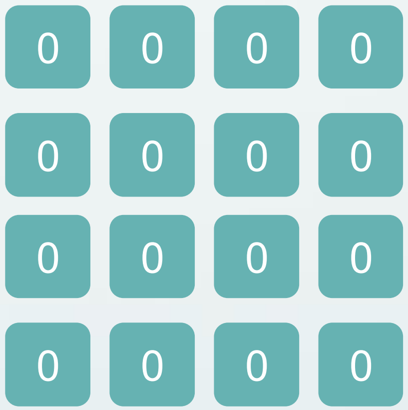
Up-counter in each cluster initialized to -(1+Manhattan distance from end cluster)



Propagate control signal from start cluster to end cluster. Advances 1 cluster per cycle.

Propagate signal starts the up-counter in each cluster.

All clusters running in-synch



When counter reaches 0, either:

- Reset the processors
- Suspend processors for configuration (at PC=0)
- Enable processors to execute (from PC=0)

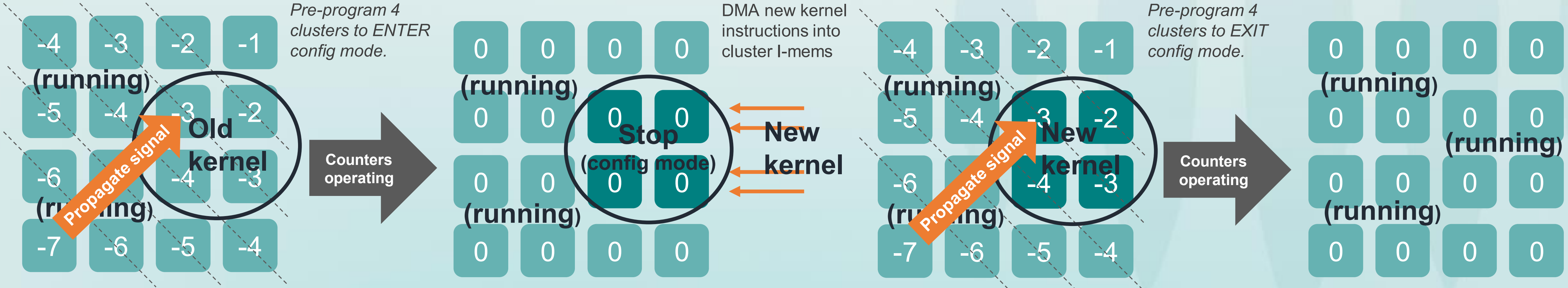
SW controls this process to manage surge current

Step 1
Propagate Signal

Step 2
Enter config mode

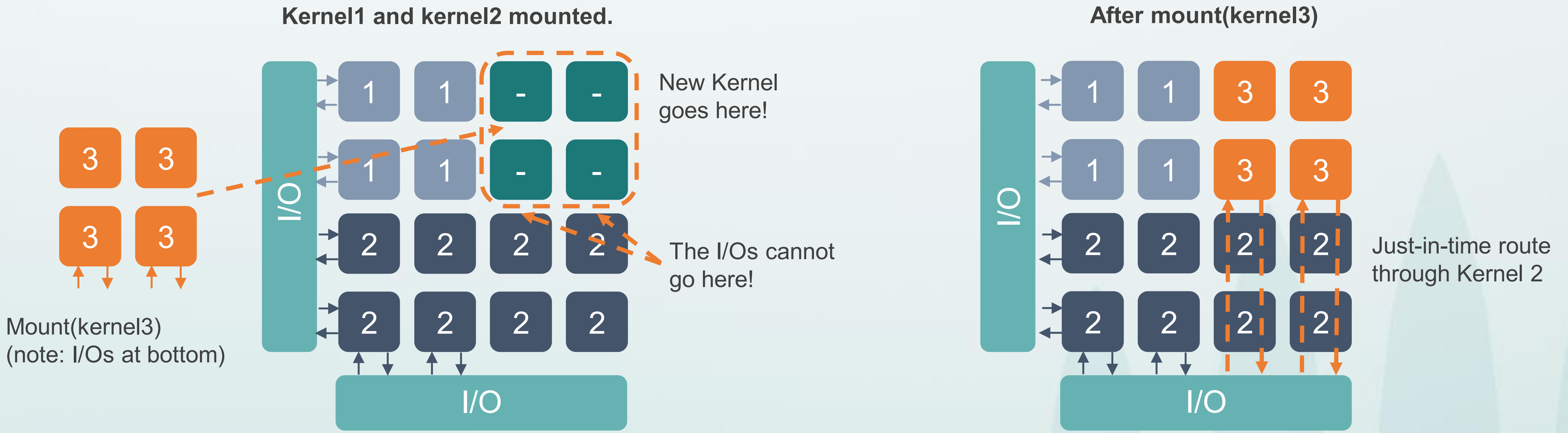
Step 3
Propagate Signal

Step 4
Exit config mode



Dynamic Reconfiguration

Runtime resource manager performing mount()



- Runtime resource manager in lightweight host.
- Mount(). Online placement algorithm with maxrects management of empty clusters.
- Uses “porosity map” for each kernel showing route-thru opportunities. (SDK provides this)
- Just-in-time Place & Route (using A*) of I/Os through other kernels without functional side-effects.
- Unmount(). Removes paths through other kernels.
- Machines are combined for mounting large kernels. Partitioned during unmount().
- Periodic garbage collection used for cleanup.
- Average mount time < 1ms

WaveFlow Software Stack

- DF agent partitioning
- DFG throughput optimization
- Runs on Session Host

WaveFlow Session Manager

- Resource Manager
- Monitors
- Drivers
- Runs on a Wave Deep Learning Computer

WaveFlow Execution Engine

- BLAS 1,2,3
- CONV2D
- SoftMax, etc.

WaveFlow Agent Library

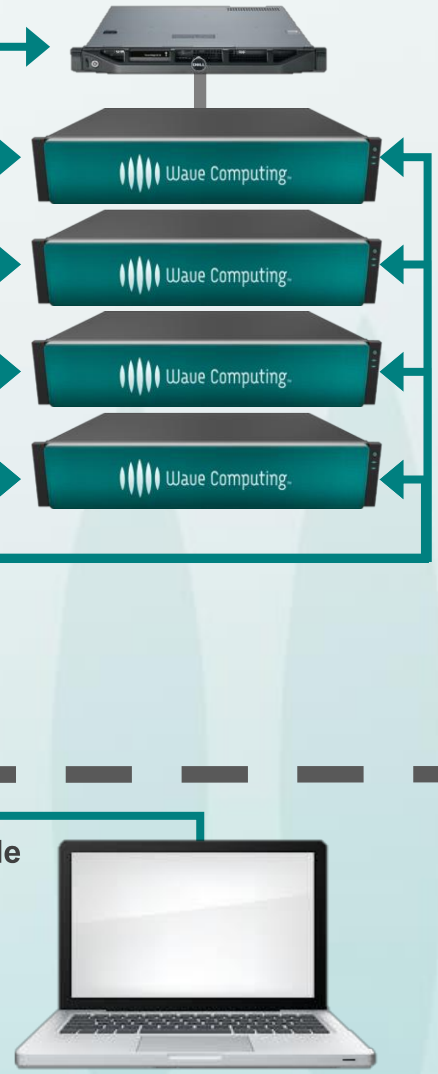
On line

Off line

- WFG Compiler
- WFG Linker
- WFG Simulator

WaveFlow SDK

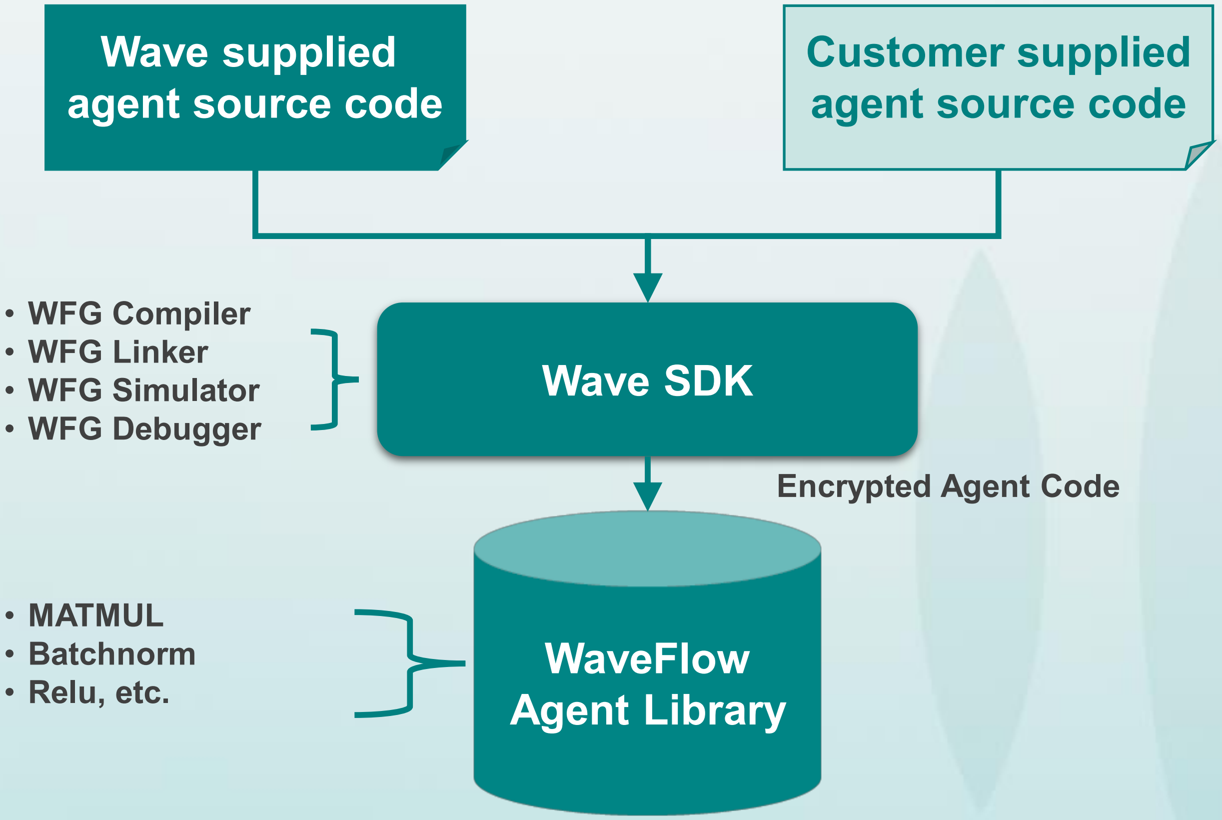
Encrypted Agent Code



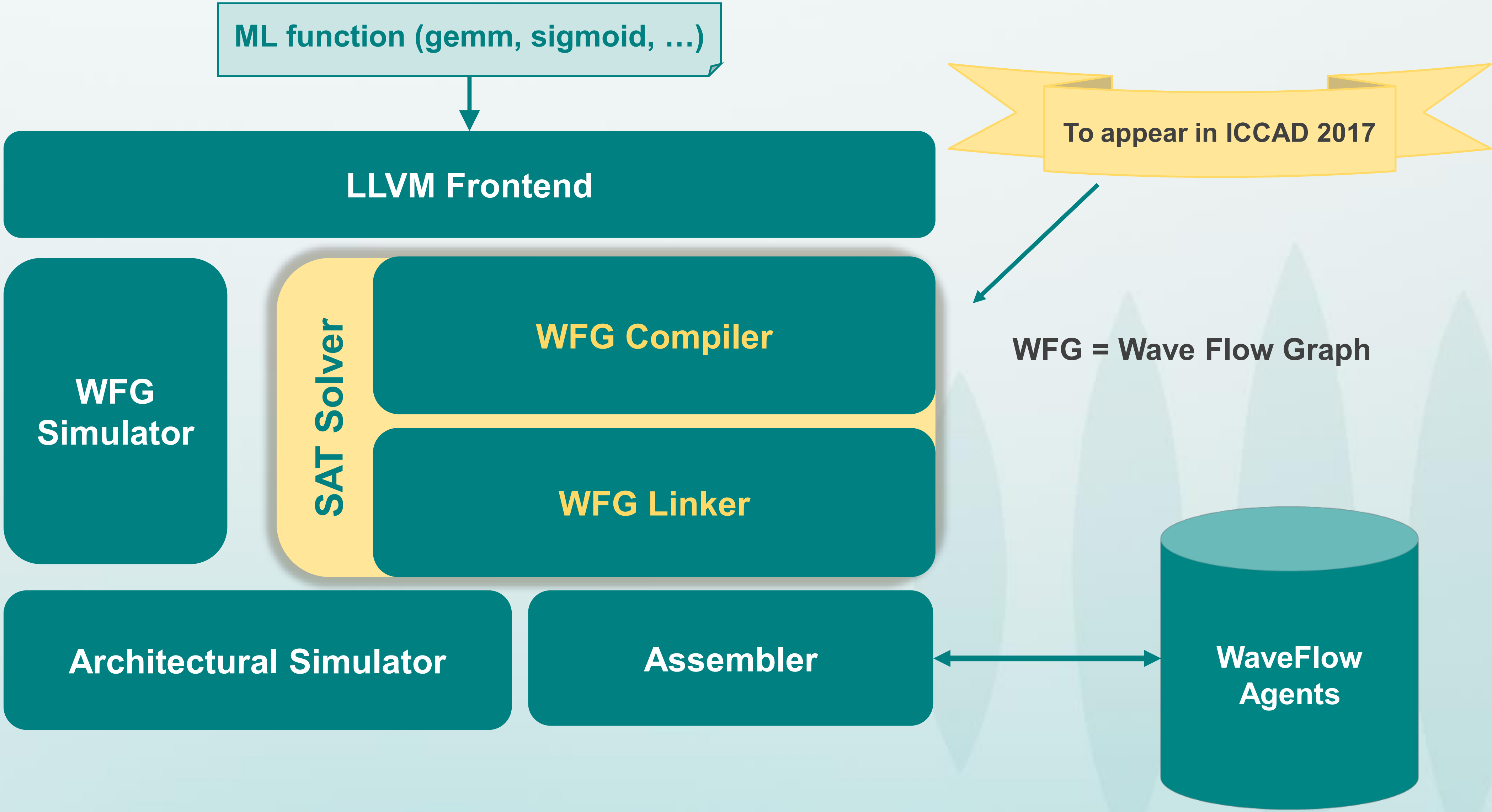
WaveFlow Agent Library

WaveFlow agents are pre compiled off-line using WaveFlow SDK

- Wave provides a complete agent library for TensorFlow
- Customer can create additional agents for differentiation



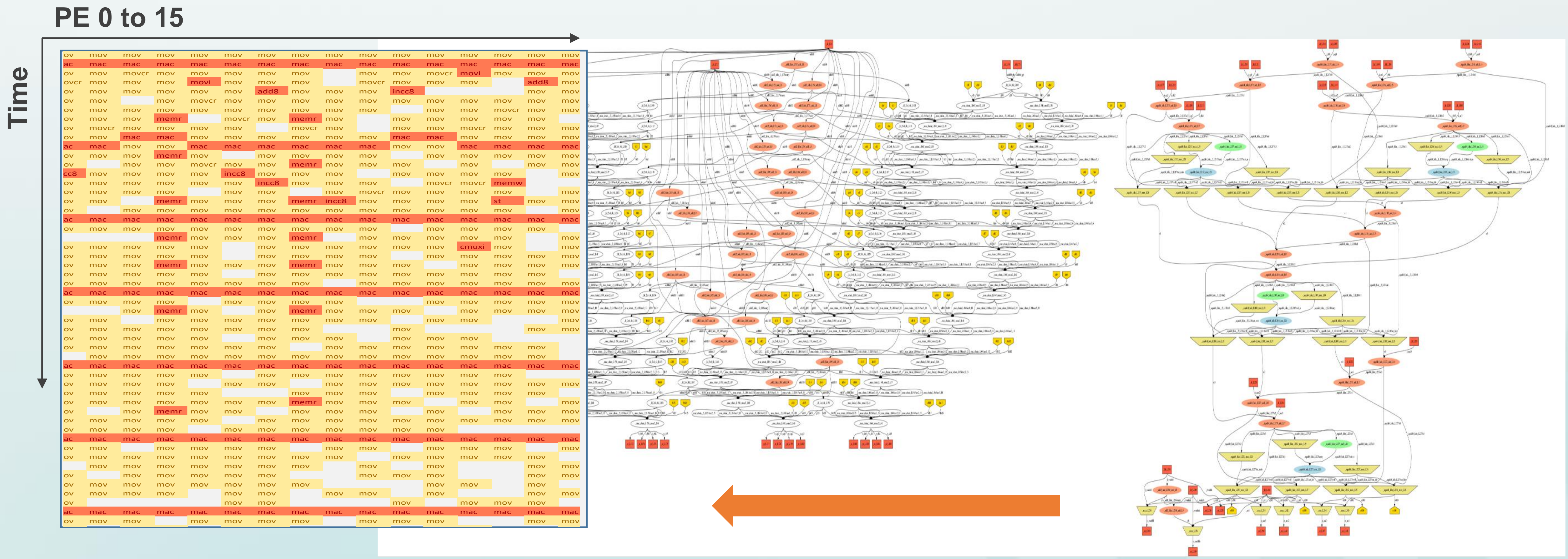
WaveFlow SDK



Wave SDK: Compiler Produces *Kernels*

Kernels are islands of machine code scheduled onto machine cycles

Example: Sum of Products on 16 PEs in a single cluster

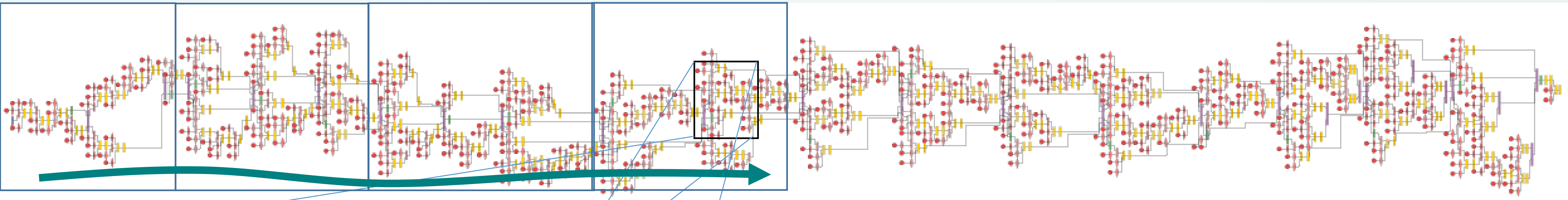


Sum of Products Kernel

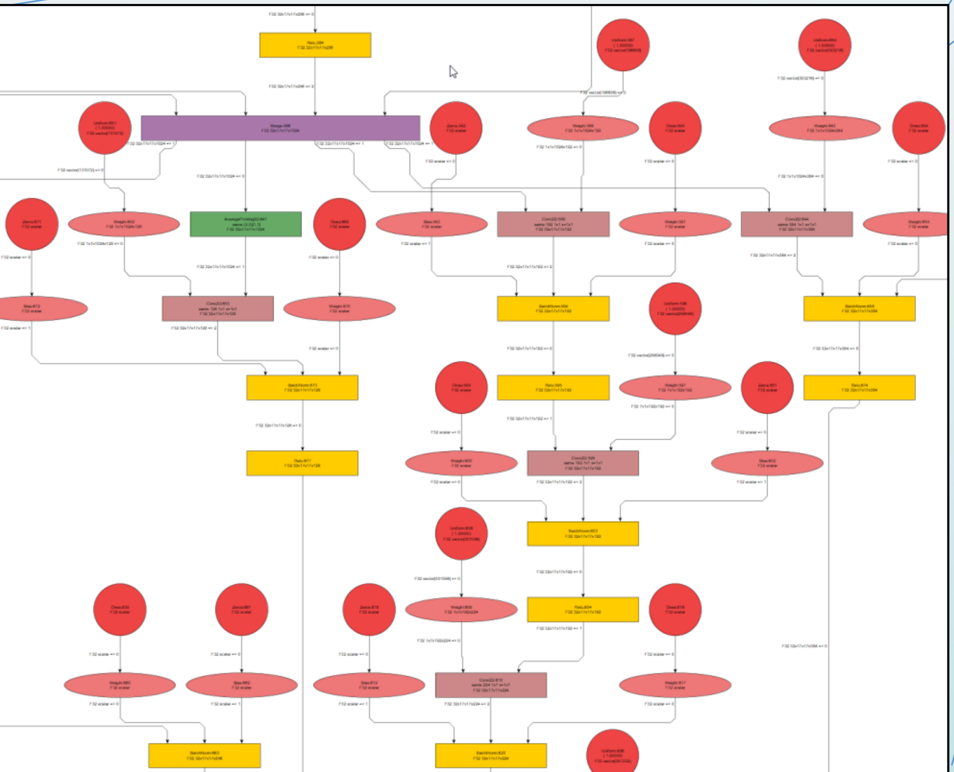
WFG of Sum of Products

Mapping Inception V4 to DPUs

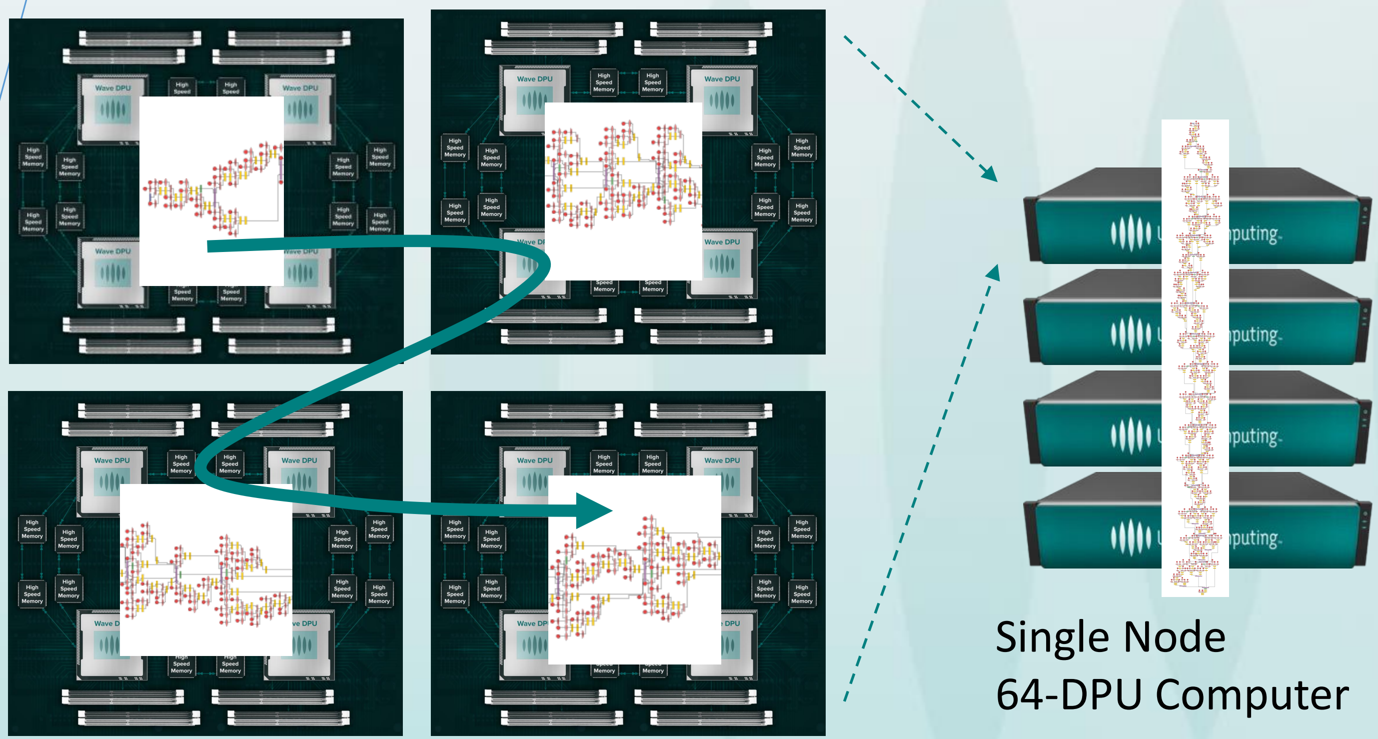
Inference Graph generated directly from Keras model by Wave Compiler



Wave Flow Graph Format



Session Manager Partitions & Maps to DPUs & Memory



Single Node
64-DPU Computer

Deep Neural Network Performance

Benchmarks on a single node 64-DPU Data Flow Computer

- ImageNet training, 90 epochs, 1.28M images, 224x224x3
- Seq2Seq training using parameters from https://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf *by I. Sutskever, O. Vinyals & Q. Le*

Network	Inferencing (Images/sec)	Training time
AlexNet	962,000	40 mins
GoogleNet	420,000	1 hour 45 mins
Squeezenet	75,000	3 hours
Seq2Seq	-	7 hours 15 min

Wave's Early Access Program

Wave is now accepting qualified customers to its Early Access Program (EAP)

Provides select companies access to a Wave machine learning computer for testing and benchmarking months before official system sales begin

For details about participation in the limited number of EAP positions, contact info@wavecomp.com