Neuromorphic Computing:
The journalistic approach

David J. Mountain
Senior Technical Director
Advanced Computing Systems Research Program

Outline

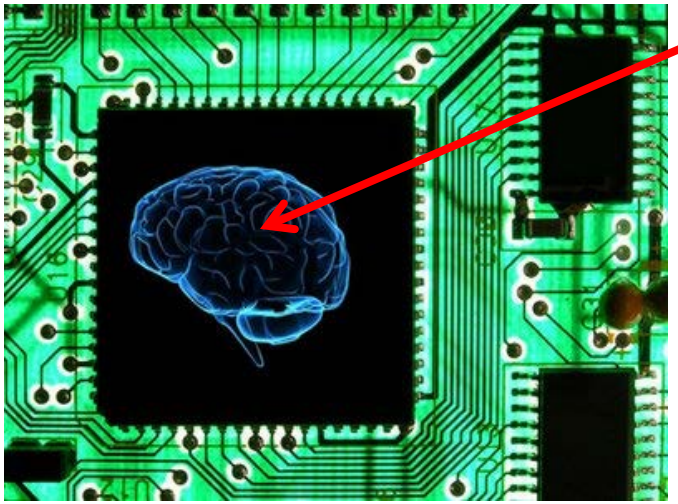
- What
- Why and when
- Who and where
- How


RESEARCH DIRECTORATE
ADVANCING INTELLIGENCE THROUGH SCIENCE

What

Neuromorphic Computing

- The integration of algorithms, architectures, and technologies, informed by neuroscience, to create new computational approaches.



Silicon Brain not required

Image: zmescience.com

Types of NMC

- Neuromimicry #1 -- Building systems that faithfully characterize or replicate ‘brain-like’ functionality to better understand the brain from a science/medical perspective

Examples:
Human Brain Project in Europe
BRAIN initiative
Allen Institute for Brain Science

- Neuromimicry #2 -- Building systems that faithfully replicate ‘brain-like’ functionality to achieve ‘brain-like’ computing or capability

Examples:
Early Carver Mead work in visual and auditory IC designs
DARPA SYNAPSE (IBM TrueNorth)

- Neuromorphic engineering -- Utilizing available algorithms, architectures, and technologies (or developing new ones) to build computing systems that are optimal based on our current understanding of neuroscience, in order to provide computing capabilities ill-served by traditional models of computing

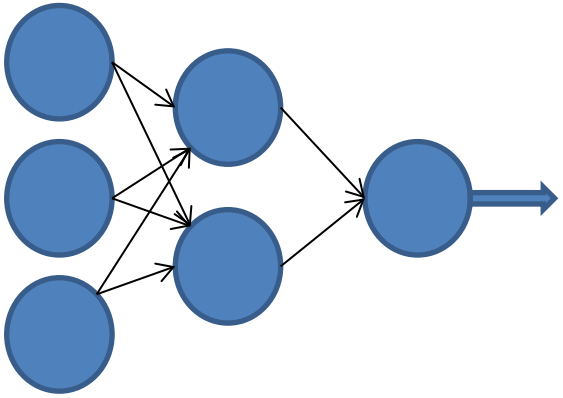
Examples:

Numerous – nVidia, Intel, Google, etc. There is a lot of work going on here. Our program is focused here.

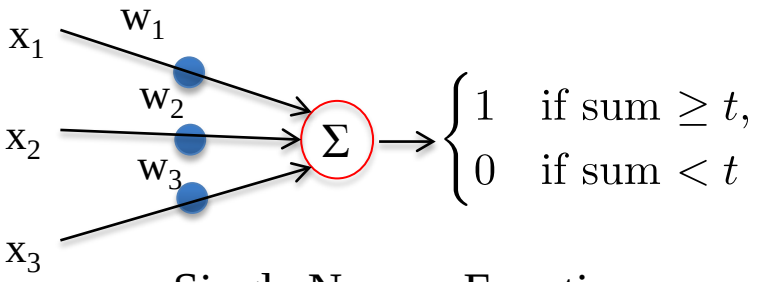
- Neural computing -- Iterative neuroscience-computer science explorations to develop theories of computation based on brain functionality

Examples:
IARPA MICrONs (Machine Intelligence from Cortical Networks)
Sandia HAANA (Hardware Acceleration of Adaptive Neural Algorithms)
Allen Institute for Brain Science

Neural Networks – A typical NMC approach

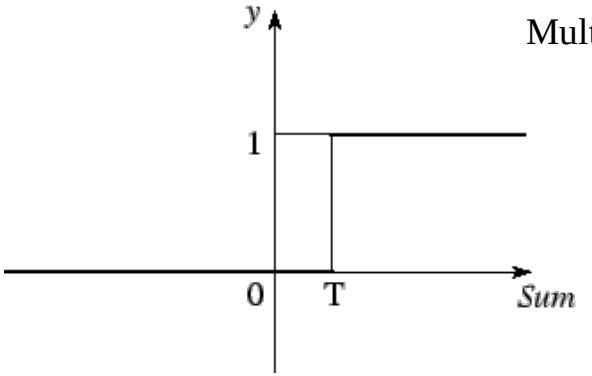


Feed-Forward Neural Network



Single Neuron Equation

Multiply accumulate (MACC) with an activation function



Activation Function

The devil is in the details

- How do you set up the network?

Fully connected, recurrent, etc.

- What data do you use?

Supervised or unsupervised approaches

- What is your learning algorithm?

Backpropagation, CLA, Reinforcement approaches (Bayesian, decision trees)

- How do your neurons function?

Simple MACC, leaky-integrate and fire, convolutions, winner take all, spike-timing

- What is your architecture?

CPU/GPU, analog, spiking event representation, hybrid, etc.

The Laboratory for Physical Sciences

Example: Our approach

- Architectures that scale to handle real applications
  - Ohmic Weave
- Methodologies and algorithms for designing/programming these systems
  - Loom
- Experience & experiments with applications to guide architectures and methodologies
  - Malware Detection

Implementing neurons using physics

I = V/R

G = 1/R

I = VG

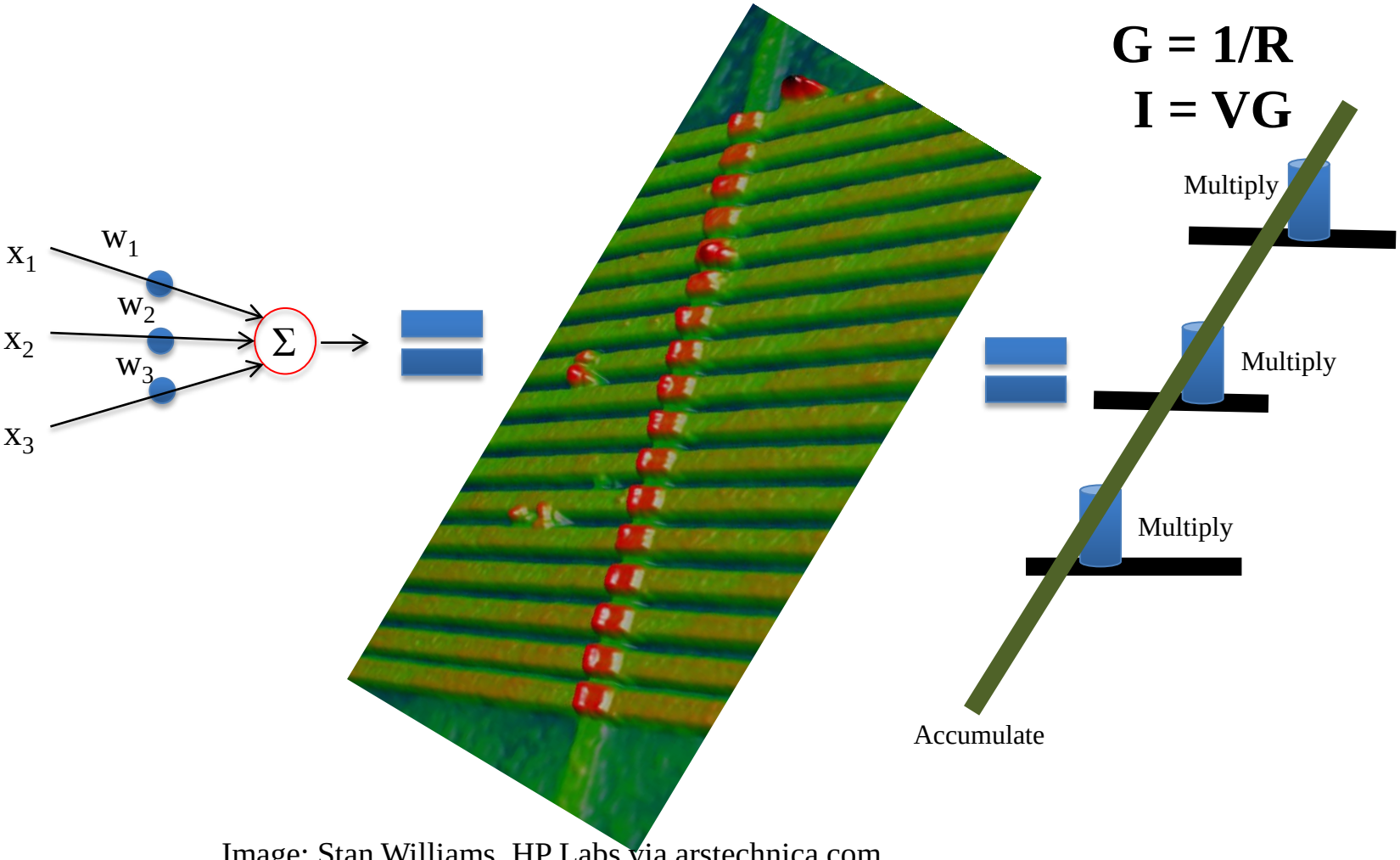
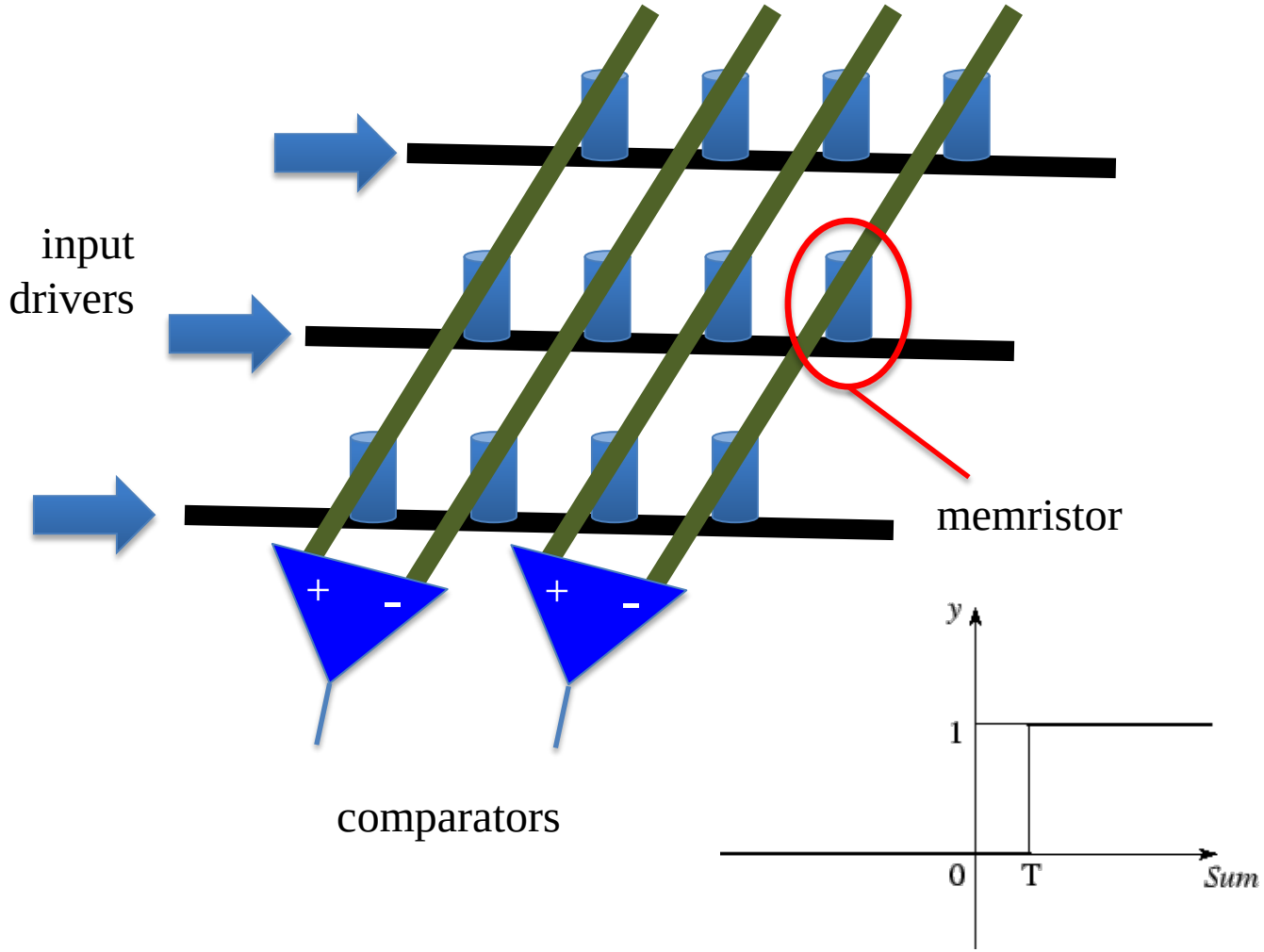


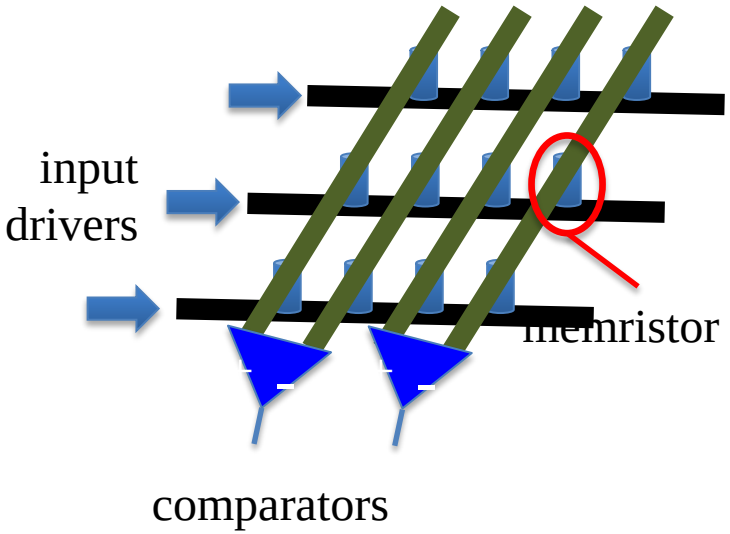
Image: Stan Williams, HP Labs via arstechnica.com

3x4 Crossbar = 2 Neurons

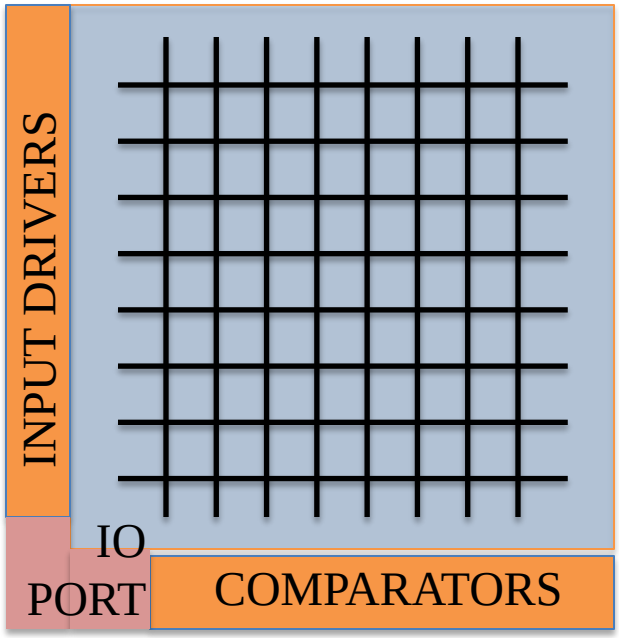


Ohmic Weave: Single Tile

256 axons, 128 neurons, 65536 synapses



256x256 memristor crossbar

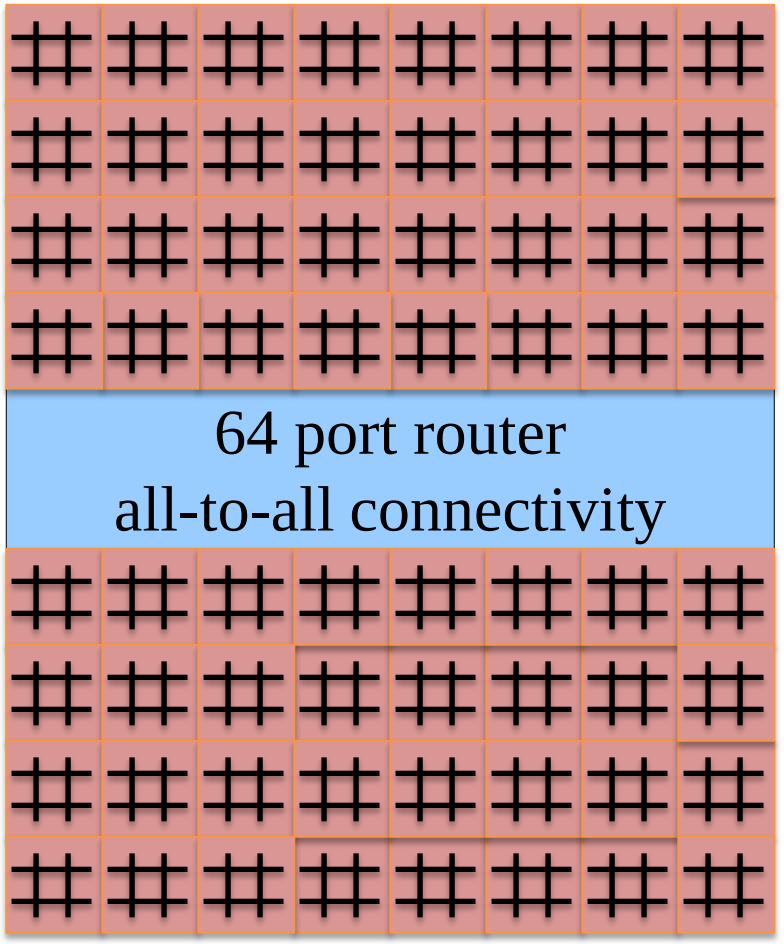


128 differential comparators

All inputs and all outputs are sent to a central router

Ohmic Weave: 64 Tile General Purpose Processor*

16k axons
8k neurons
4M synapses



*56 Tera synaptic ops per watt (TSOPS/W), 1.1 TSOPS/mm^2

The Laboratory for Physical Sciences

Tools, Methodologies, Algorithms

- Loom – Ohmic Weave design tool
  - Python classes with C, Cuda extensions
  - Enables exploration of design trade-offs
    - Weights with limited precision and ranges
    - Neural network topologies (layers, neurons per layer)
    - Connectivity pruning
- Simulates Ohmic Weave designs on CPUs, GPUs
    - Debug with full view of internal state

Methodology: Block Based Design

- Decompose the problem into blocks
  - Much like block based CMOS design
  - Can pull blocks from a “circuit library”
- Loom can compose blocks into a single larger network
  - Will optimize by removing unused neurons and connections
  - Compresses to minimum number of layers
  - Handles recurrence/loops

Digital Hierarchical Neural Nets

- Digital functions must be 100% correct
- Divide and conquer by partitioning
  - 64 inputs = 1.6×10^{19} training vectors
  - 4 x 16 inputs = 2.56×10^{5} training vectors
- Reduce the training set size
  - But train to 100% accuracy
  - The logic truth table becomes the training set
  - The training data encompasses all possible data

Training

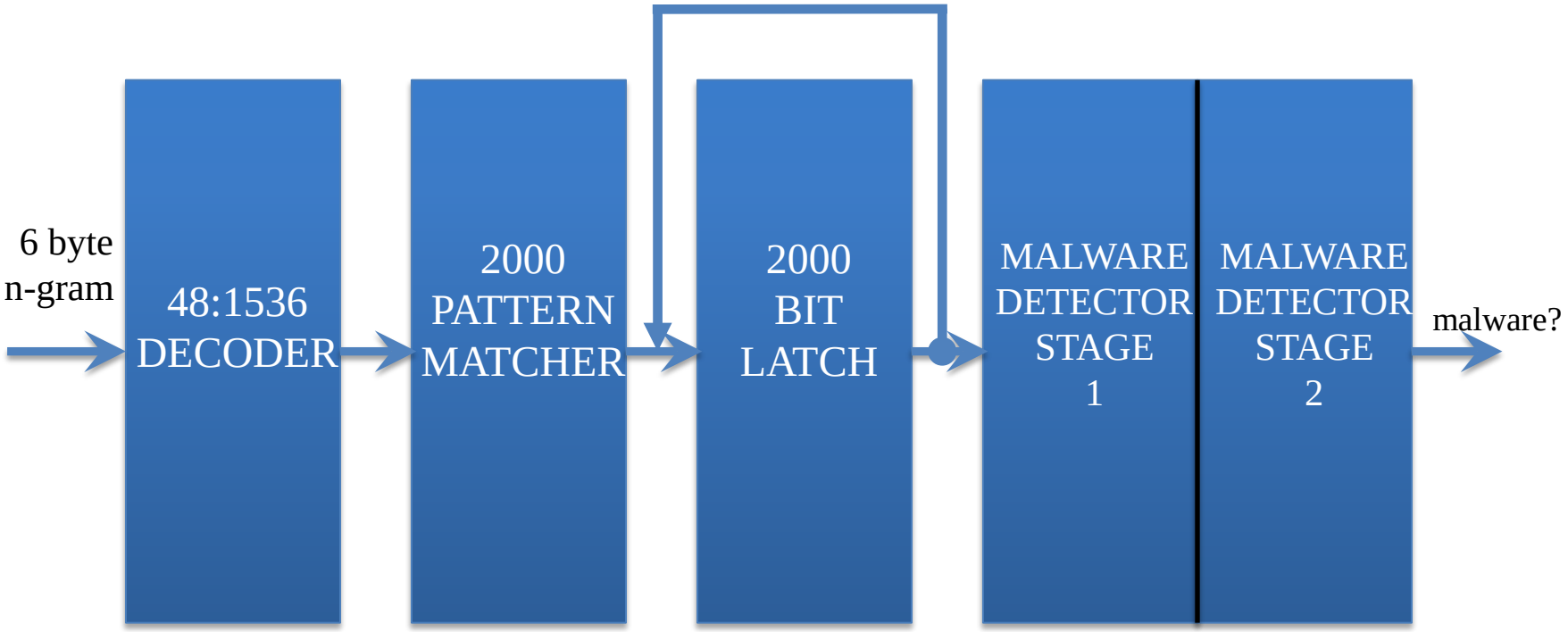
- Loom can train blocks given a training set or truth table
  - Uses the Concurrent Learning Algorithm*
  - Can train for exact logic or for inexact classifiers

*M. McLean, “Concurrent Learning Algorithm and the Importance Map,” Network Science and Cybersecurity, ed. by R. Pino, Springer 2014, vol. 55, pp. 239-250.

Application: Malware Detection

- Classifies files as malware (e.g. virus) or benign
  - Looks at the file in 6 byte *n-grams* at a time
  - Matches 2000 critical n-grams, notes their presence in a 2000 bit latch
  - Uses a neural network classifier to decide if pattern in latch is malware

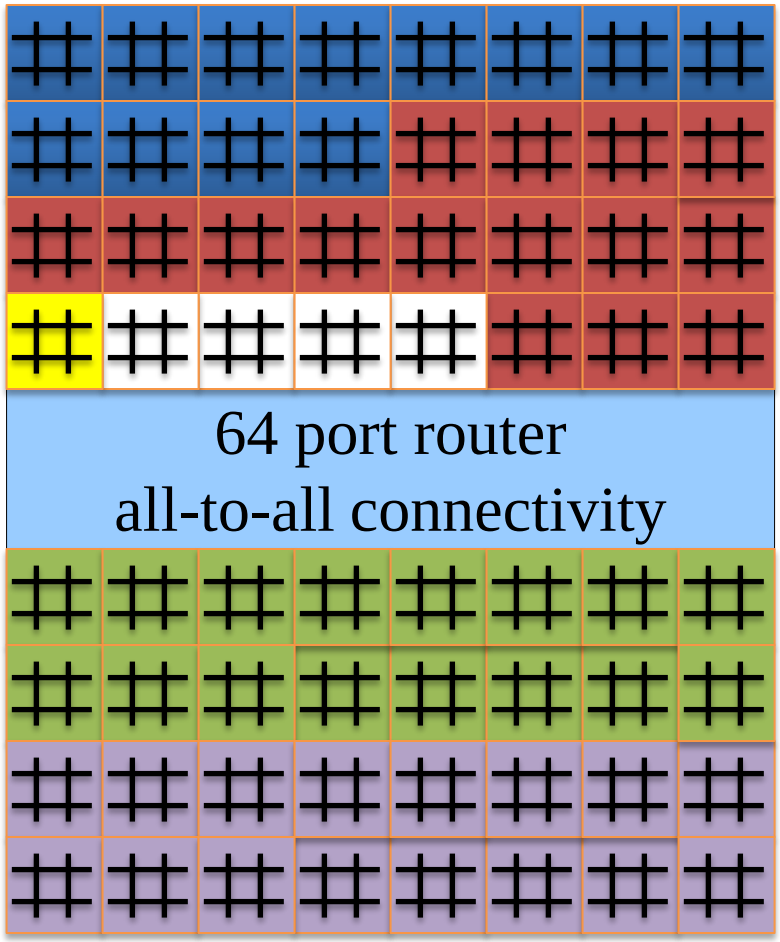
Conceptual Diagram

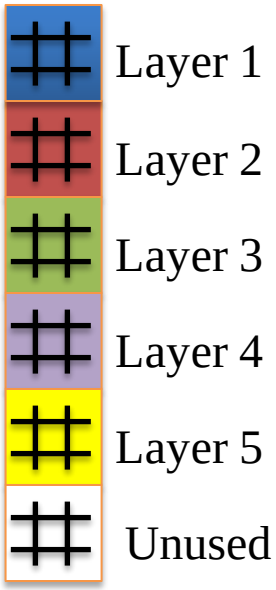


About 4007 instances of 5 unique blocks

About 5800 neurons in 5 layers

Mapped to Ohmic Weave using Loom





Malware detection using neural nets: General purpose Ohmic Weave vs CPU

CPU: 6 Core Intel Core i7 3930K; Throughput is 1.92 Gbps
Ohmic Weave requires 1 neural net to match that throughput

Numbers shown below are for a somewhat 'optimistic' Ohmic Weave implementation

Function	Area (mm^2)	Power (mW)
Row Drivers	0.046	31.3
Memristive Array	0.510	71.7
Comparators	0.083	107.5
Router	0.544	50.0
Total	1.183	260.5

338x Improvement in Area
307x Improvement in Power

Why and when (or why now?)

Lots of reasons

- Moore's Law

Proliferated the amount and type of data that can be created and managed in digital form, enabling 'non-scientific applications' to provide value – business driver

Created extremely low cost digital devices that need to be easy to use, and created the desire to use them for 'non-scientific applications' – consumer driver

Enabled high speed networks and critical functions to be controlled from a distance simultaneously and cheaply – national security driver

It's ending – new opportunities for creativity in models of computing and architecture

- NSCI
- OSTP grand challenge

National level interest and $$

OSTP Grand Challenge

Create a new type of computer that can proactively interpret and learn from data, solve unfamiliar problems using what it has learned, and operate with the energy efficiency of the human brain.

Examples

- Sensory system based applications – robotics of various types

Image processing, pattern recognition, speech recognition

- Trend identification, prediction – analytics

Multimodal inputs, high bandwidth, time sensitivity, anomaly detection

- Decision making – human intelligence augmentation

Model development, context awareness

Cognitive Cybersecurity

Scales with technology, not humans

Able to rapidly deal with changes

Level 1 – prevent (minimize) unauthorized access

Level 2 – identify anomalous behavior

Level 3 – contextual analysis for adversary intent

Who and where

Good things to look into – think interdisciplinary efforts

- NICE (Neuro-inspired Computational Elements workshop)

Run by Brad Aimone at Sandia, in California this year (Mar 7-9, 2016)

- IJCNN (International Joint Conference on Neural Networks) or other major conferences

This year you can spend time in Vancouver, Canada! (July 25-29, 2016)

- IARPA MICrONS program

Awards to Harvard, Baylor, Allen Institute, Princeton, and Carnegie-Mellon as primes

They will probably be starting more projects in NMC

- Neuromorphic computing forecast by RD at NSA

Come see me about this one

- China is becoming very active in this area

China Brain-Inspired Computing Research (CBICR) program at Tsinghua University


RESEARCH DIRECTORATE
ADVANCING INTELLIGENCE THROUGH SCIENCE

How

Key challenges – at the high level*

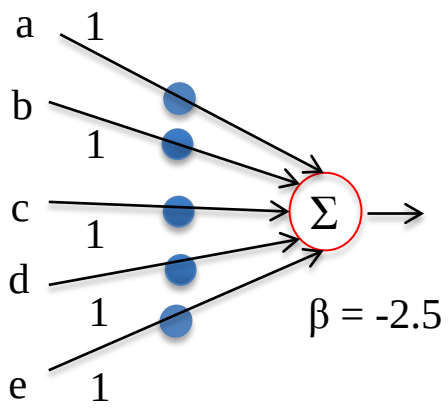
- "In particular, if one can identify a set of computational operations that a hardware system performs well, a directed abstraction from a complex biological system so as to emphasize those optimal operations and capture the desired function would be ideal."

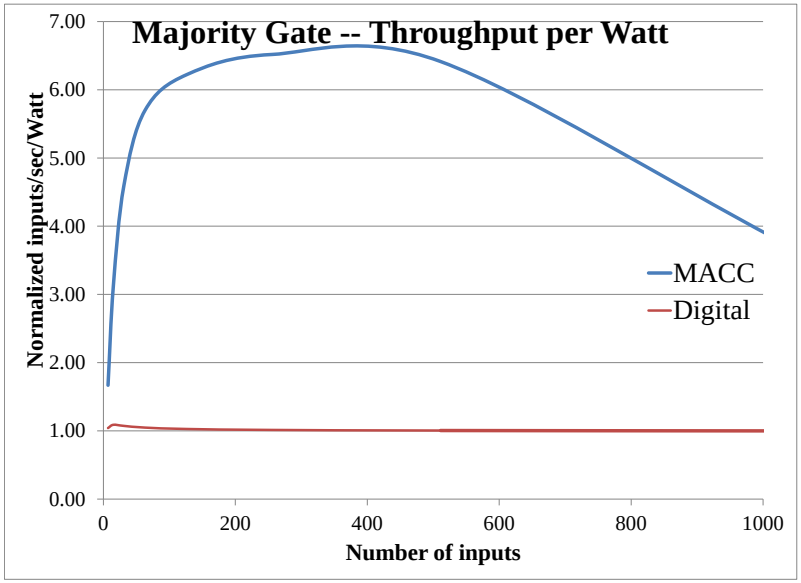
- Neural theory, neural-inspired computing algorithms and architectures, and novel electronics capabilities

*Brad Aimone et al, Sandia NMC forecast for RD

Different computational primitives will become the common case: Majority function example

Digital implementations are relatively inefficient for large numbers of inputs; MACC-centric design appears to have a large sweet spot





Key challenges – one level down

- One shot learning, unsupervised learning, concept drift
- Data reduction techniques
- Building and curating credible and available data sets
- Comprehensive modeling and simulation environments for iterative algorithm-architecture-technology co-design
- Creating and maintaining multi-disciplinary teams
- Brain imaging techniques that facilitate neural computing goals
- Metrics for evaluating ‘brain-like’ systems

A way to think about metrics*

HIGH 10. The computer decides everything, acts autonomously, ignoring the human.

9. informs the human only if it, the computer, decides to

8. informs the human only if asked, or

7. executes automatically, then necessarily informs the human, and

6. allows the human a restricted time to veto before automatic execution, or

5. executes that suggestion if the human approves, or

4. suggests one alternative

3. narrows the selection down to a few, or

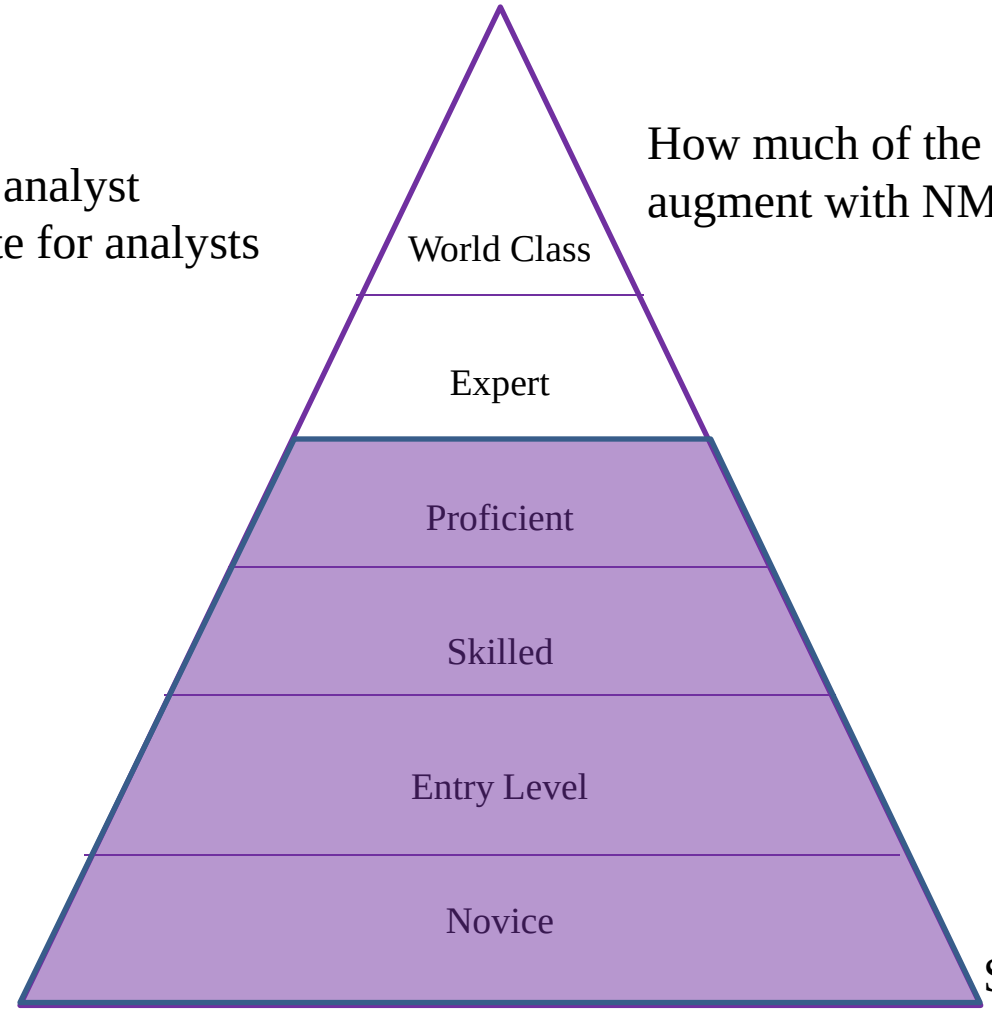
2. The computer offers a complete set of decision/action alternatives, or

LOW 1. The computer offers no assistance: human must take all decisions and actions.

*A Model for Types and Levels of Human Interaction with Automation
Raja Parasuraman, Thomas B. Sheridan, and Christopher D. Wickens
IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART A: SYSTEMS AND HUMANS, VOL. 30, NO. 3, MAY 2000

Another version of a metric

Ops/analyst
Ops/trained analyst
Learning rate for analysts
Scaling rate



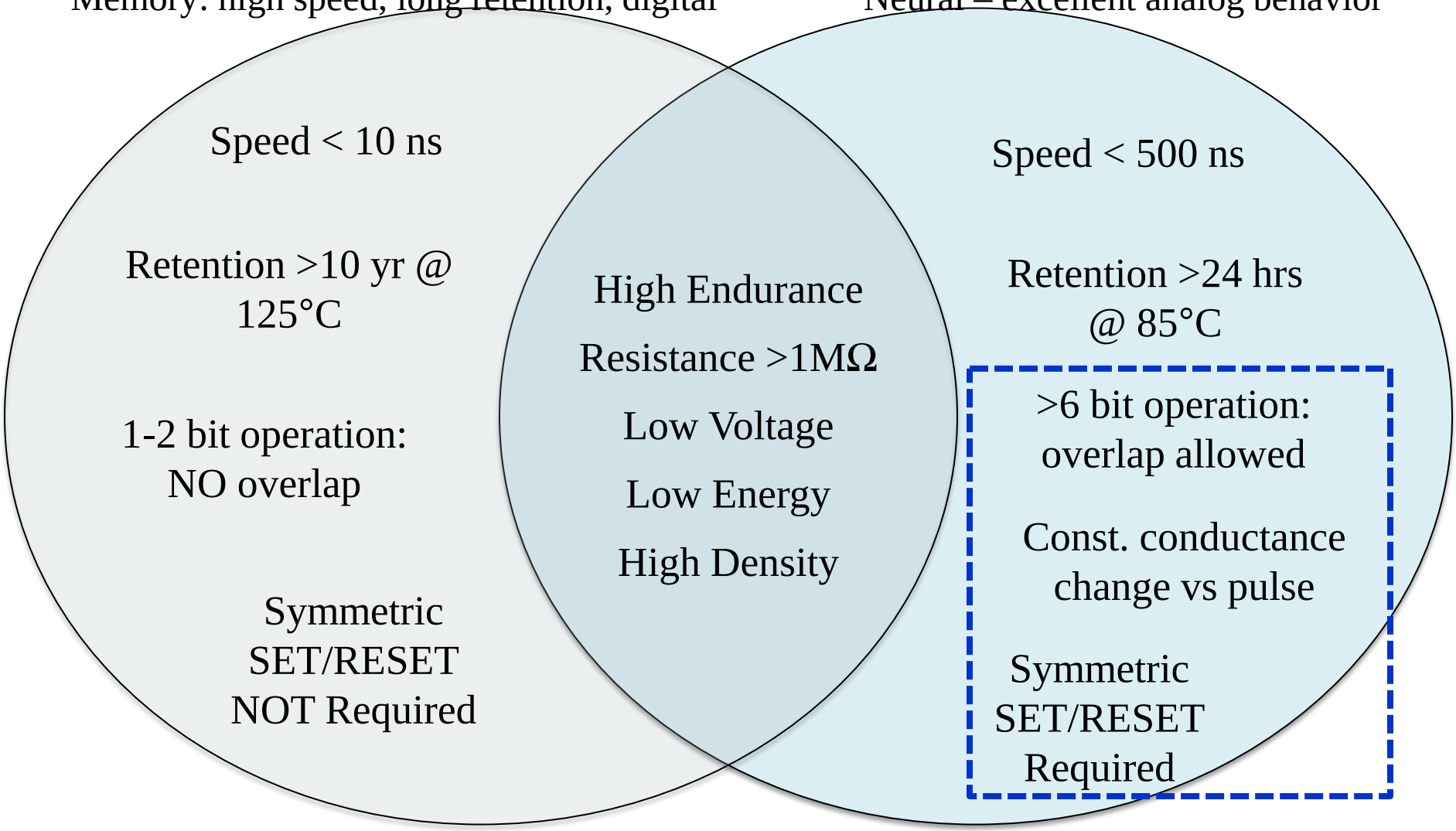
How much of the pyramid can you augment with NMC?

Skill level pyramid

Novel electronics challenges

- Interconnect density
- Controlled growth of interconnects
- Access devices (monolithic 3D)
- Devices to vastly improve comparators and on-chip programming circuits
- Efficient STDP/Spiking device – superconducting electronics?
- Efficient analog comms (or efficient transducers to optical, magnetic, etc.); includes sensors
- Memristors optimized for NMC

Memristor characteristics of value*



*Courtesy of Dr. Matthew Marinella, Sandia National Labs

Acknowledgments to the NMC team that is making all this happen

Chris Krieger
Mark McLean

along with a substantial number of academic, national lab and industry partners


RESEARCH DIRECTORATE


LPS
The Laboratory for Physical Sciences


IBM


Sandia
National
Laboratories


UNIVERSITY of
DAYTON


DARPA


AIR FORCE RESEARCH LABORATORY


UNIVERSITY OF
PITTSBURGH
VERITAS
1787
VIRTUS


UNIVERSITY OF
MICHIGAN


ASU
ARIZONA STATE
UNIVERSITY


Hewlett Packard
Enterprise


IARPA
BE THE FUTURE

The Laboratory for Physical Sciences

Contact Information

- Laboratory for Physical Sciences
  - Advanced Computing Systems Research Program

  - Christopher D. Krieger
    - krieger@lps.umd.edu
  - David Mountain
    - davidjmountain@ieee.org
  - Mark McLean
    - mrmclea@lps.umd.edu


RESEARCH DIRECTORATE
ADVANCING INTELLIGENCE THROUGH SCIENCE

Questions?