Portland State UNIVERSITY

Massively Parallel Adaptive Computing

Dan Hammerstrom
Electrical And Computer Engineering

Portland State UNIVERSITY

This talk:

1. Research Objectives
2. Motivation
3. An Example Computational Model – Bayesian Memory
4.
5. Workshop on Massively Parallel Adaptive Computing
6. DARPA SyNAPSE Program
7. ITRS – Emerging Research Devices / Architectures
8. Conclusion

1. Research Objectives

What Are We Doing?

- Studying various candidate modular building blocks that will allow us to create very large, scalable, structures for Intelligent Signal Processing (ISP – to be defined later)
- We want models that map cleanly to current and future hardware technologies
- Due to their massive parallelism and relatively low precision, neural models are appealing
  - They bring certain efficiencies to “inference”
  - They also allow for a wider range of architecture options and significant implementation optimizations

2. Motivation

Why Are We Doing It?

Intelligent Computing

- In spite of the transistor bounty of Moore's law, there is a large class of problems that computers still do not solve well
- These problems involve the transformation of data across the boundary between the real world and the digital world
  - "the digital seashore" Hiroshi Ishii, MIT Media Lab
- These "boundary" problems occur wherever a computer is sampling and acting on real world data
  - Which includes almost all embedded computing applications
- Our inability to adequately solve these problems constitutes a significant barrier to computer usage and to huge potential markets

- Examples include: computer vision, speech recognition, textual and image content recognition, robotic control, advanced automotive safety, intelligent power management, and data mining (making sense of massive quantities of seemingly unrelated data), ...
- These are difficult problems that require computers to find complex structures and relationships through space and time in massive quantities of low precision, ambiguous, noisy data
- Biological systems do this to various degrees – even "simple" insect brains!

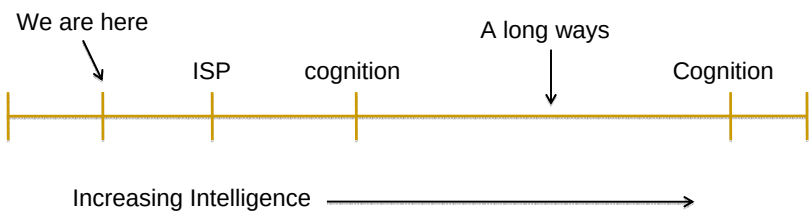
cognition vs. Cognition

- Most mammals have well developed neocortex
- This is particularly true of primates
- The cortices for these animals (excluding humans) perform fairly sophisticated "cognition" (with a small 'c')
  - Monkeys can do planning and learn complex social behaviors
- Yet Humans, on the other hand, have an additional capability for reasoning, planning, and, in particular, for language – "Cognition" (with a capital 'C')
- Many researchers confuse this distinction, often referring to cognition when they really mean Cognition and vice-versa

- Traditional Artificial Intelligence, among other things, primarily went after Cognition without having solved cognition
- We believe that we need to solve cognition before we can solve Cognition, and that we need to solve the ISP problem before we solve cognition!
- ISP is hard enough …

The Scope of The Project

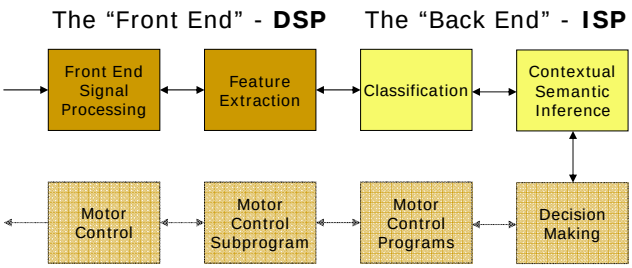
- One can conceptually think of this as a spectrum
- And though not universally accepted, it has been hypothesized that this spectrum is more or less continuous from one end to the other with no real "quantum" leaps anywhere along the way



Intelligent Signal Processing

- The term Intelligent Signal Processing (ISP) has been used to describe algorithms and techniques which involve the creation, efficient representation, and effective utilization of large complex models of semantic and syntactic relationships
- ISP augments and enhances existing Digital Signal Processing (DSP) by incorporating contextual and higher level knowledge of the application domain into the data transformation process

- Although an over-simplification, this flow diagram is useful in discussing the functional components of this problem domain
  - Real applications tend not to partition so cleanly
  - Notice the bidirectional arrows between each block – data flow both ways
- Most IC applications have components in each of these blocks



The "Front End"

- We understand the front end pretty well, it is the realm of traditional digital signal processing (DSP)
- Front end algorithms tend to involve applying the same computation over a large array of elements, they are data parallel and communication tends to be local
  - Most neuromorphic VLSI, for example, has been used primarily for this kind of front end processing
  - Another example of such an architecture is CNN (the Cellular Non-linear Network) developed by Chua, Roska et al.
    - A 2D array of programmable, "analogic" processors, with local 2D grid interconnect

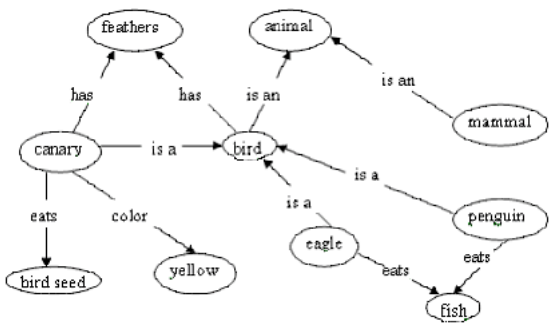
But Then There's The "Back-End" …

- In the early days of computing, "Artificial Intelligence" focused on the representation and use of contextual and semantic information
- Knowledge was generally represented by a set of rules
  - Logic operations (e.g., based on first order predicate calculus) were used to manipulate the rules and to "infer" the properties and state of an environment from input data
  - By using inferred state, logic operations also enabled planning and decision making algorithms
- However, these systems were "brittle," exhibiting no real flexibility, generalization, or graceful degradation
  - Often catastrophic failure resulted if a circumstance occurred that was not covered by a rule

- And they were unable to adapt dynamically (learn) within the context of a real world application
- ANNs (Artificial Neural Networks) extended computational intelligence in a number of important ways, primarily by adding the ability to incrementally learn or adapt to complex non-linear functions
  - They created a whole new set of very useful statistical and non-linear learning techniques
- However, ANNs generally focused on the lower levels of feature extraction and classification, and like AI before them, they have not scaled well
  - The lack of scaling also meant that traditional desktop computers (including clusters) had the necessary performance for applications and research,
  - This also more or less eliminated the need for specialized hardware

What Does The "Back End" Do?

- Simplistically it captures information on "high order" relationships of "abstract" entities
- Which can be represented by a graph structure
- These used to be "rules," but more recently they have become probabilistic relationships -> "Bayesian" Networks



- Inference is then performed over these structures
- This is not necessarily cognition, but is potentially an important building block

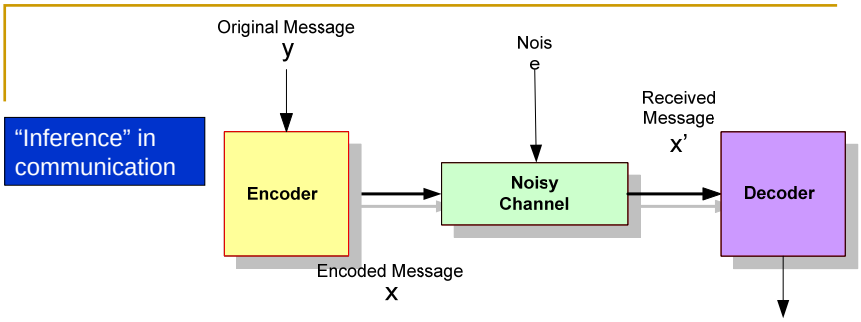
- Back-end algorithms have fundamentally different characteristics from Front-end algorithms
- For example, they are more sparsely connected and more sparsely activated
- The inference process requires more intercommunication in both directions
- These can be huge networks – consider a natural language
- It is most likely that mammalian neocortex uses distributed representations (more on this later) to reduce network size and scope, and also to make inference more efficient

- We now have **Bayesian networks**
  - These systems are far less brittle and they also more faithfully model aspects of animal behavior
  - Animals learn from their surroundings and appear to do a kind of probabilistic inference from learned knowledge as they interact with their environment
- However, learning is still difficult
  - Learning the network structure itself, is an NP-Hard problem
  - Though there are reasonably efficient ways to adjust the individual probabilities

Bayesian Networks

- Bayesian nets express the structured, graphical representations of probabilistic relationships between several random variables
- The graph structure is an explicit representation of conditional dependence (encoded by network edges)
  - And they allow arbitrary queries, that is, the setting any variables to known values and then inference the most likely values of
- Inference, however, is still compute intensive, and has also been shown to be NP-Hard

- One general technique for doing inference in Bayesian Networks is Bayesian Belief Propagation defined by Pearl
  - **Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,** by Judea Pearl, Morgan Kaufmann, 1988, **ISBN-10:** 1558604790
- A Bayesian network is generally much more compact than the full joint distribution
  - Some consider it an example of a general property of locally structured (sparse) systems, where each subcomponent interacts directly with only a limited number of other components (often “neighbors” according to some geometric metric)
- By reducing the order sufficiently, we reach an inference problem that is computable in a reasonable time
  - Though we may also have thrown away useful information



$$p(y \mid x') = \frac{p(x' \mid y)p(y)}{p(x')} = \frac{p(x' \mid y)p(y)}{\sum_x p(x' \mid x)p(x)}$$

The Inference Problem: Choose the most likely *y', based on P[y|x']*

We need to "infer" the most likely original message given what we received and our knowledge of the statistics of the messages being generated

Other Requirements: Scaling

- The scaling limitations of both symbolic and traditional neural network approaches constitute one of their biggest shortcomings
- It is very likely that sheer size is a major component of the "secret sauce of cognition"
  - **Consider the differences: hundreds of rules or thousands of nodes vs. millions / billions of neurons**
- Therefore we need models and hardware/software platforms that scale to very large sizes

Other Requirements: Learning / Adaptation

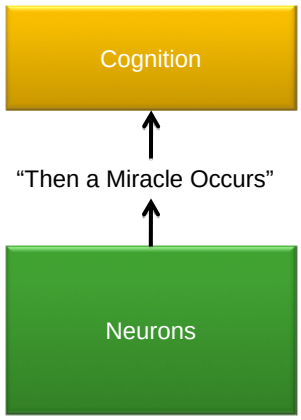
- Animals learn in real time from their surroundings and appear to do a kind of probabilistic inference from learned knowledge as they interact with their environment
- Therefore, another important characteristic of ISP is incremental, integrative adaptation / learning during system operation
- In fact, systems configuration becomes more a process of self-organization and adaptation rather than of programming
  - Principles of Organic Computing – von der Malsburg

3. An Example Computational Model – Bayesian Memory

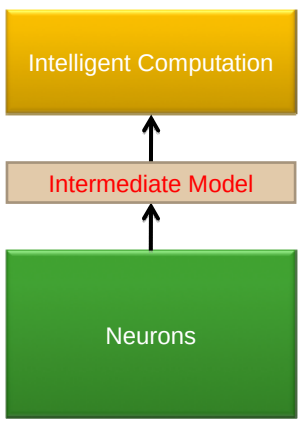
How Are We Doing It?

The "Gap"

- So do neural techniques lead to advanced ISP?
- Most Computational Neuroscience is weak in making the jump from spiking neurons with learning rules such as STDP to Cognition
- Even solutions to the more narrowly defined ISP back-end problem are not obvious



- One way to possibly bridge the gap is to define a computational model that "spans" the gap
- A candidate has been proposed by Albus and many others:
- The Cortical Computation Unit (CCU)



Albus: What is the path to success for reverse engineering the brain?

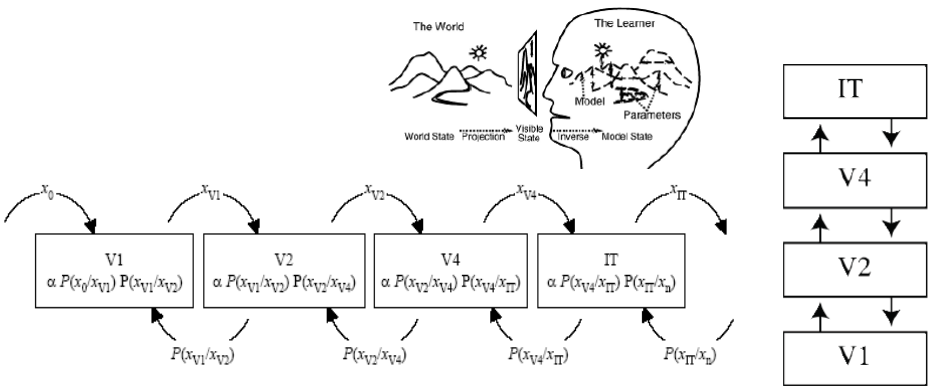
Pick the right level of resolution

- overall system level (central nervous system)
- a AI and Cognitive Neuroscience units (e.g., cortical regions)
- macro-computational units (e.g., cortical hypercolumns & loops)
- micro-computational units (e.g., cortical microcolumns & loops)
- CCUs
- neural clusters (e.g., spinal and midbrain sensory-motor nuclei)
- neurons (elemental computational units) – input/output functions
- synapses (electronic gates, memory elements) – synaptic phenomena
- r Mainstream Neuroscience & Neural Nets y) – molecular phenomena

Slide courtesy James Albus, "Reverse Engineering the Visual System"
From the PSU / Intel "Massively Parallel, Adaptive Computing" Workshop

Portland State UNIVERSITY

- Probabilistic inference at almost all levels of abstraction.
- Conditional probabilities/ priors - model the world
- Bi-Directional Belief propagation – visual cortex model



Lee and Mumford Visual cortex model

From *Big Brain* by Gary Lynch and Rick Granger (Palgrave McMillan 2008):

- "Although the specialized 'front end' circuits of the brain, with their point-to-point circuit designs, specialize in their own particular visual and auditory inputs, the rest of the brain converts these to random-access encodings in association areas throughout cortex. … these areas take initial sensory information and construct grammars
- "These are not grammars of linguistic elements, they are grammatical organizations (nested, hierarchical, sequences of categories) of percepts – visual, auditory, and other
- "Processing proceeds by incrementally assembling these constructs … these grammars generate successively larger 'proto-grammatical fragments,' eventually constituting full grammars"

- "They thus are not built in the manner of most hand-made grammars; they are statistically assembled, to come to exhibit rule-like behavior, of the kind expected for linguistic grammars
- "Proto-grammatical fragments capture regularities that are empirically found to suffice both for recognizing and generating grammatical sequences
- "Auditory pathways in our brains grew and lengthened building voice-sounds into words, words into phrases, phrases into sentences"

Distributed Representations

- **Perhaps the single most important implementation characteristic is the distributed data representation**
- There is reasonable agreement that neural structures represent data in a "distributed" manner, but there is less agreement on the details of these representations
- Different models and algorithms distribute their representations in different ways
  - *Neural Codes and Distributed Representations*, Eds. L. Abbott and T. Sejnowski, MIT Press 1999

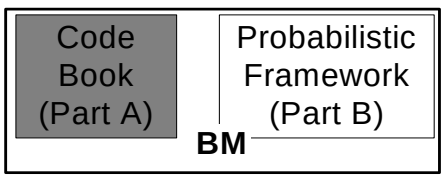
- Distributed representations allow for incremental, integrative adaptation or learning
- Computing with distributed representations is the computational equivalent of spread spectrum communication – spreading the computational and communication loads more evenly across the system
- One hypothesis is that distributing data is a kind of factorization allowing for much more efficient inference

- Our models are based on the ideas of the “Cortical Computation Unit,” where we assume a building block that corresponds roughly to a cortical column
- For this we have developed a “Bayesian Memory” (BM) module that we use as a CCU and which is connected into hierarchical arrays
  - Others have similar modular structures Lansner, Granger, Anderson, Hecht-Nielsen, George/Hawkins (Numenta)

- **We do not claim that this is a model of cortex or that it implements cognition (or Cognition)**
- **Our goal is to expand the existing capabilities of Intelligent Signal Processing (ISP) and explore more radical kinds of computer architectures**
- **The goal is advanced ISP available in inexpensive, hand-held devices!**

Bayesian Memory

- Code Book (CB) of vectors/patterns
  - Mechanism to compare, add and store vectors or similar vector quantization mechanism
- Probabilistic Framework
  - Mechanism to learn joint/conditional probabilities, the CPT
  - The CPT stores the probabilistic relation between the CB entries of the parent BM and the CB entries of the child BM
  - Bidirectional flow of probabilistic information (Bidirectional Belief Propagation – Pearl’s Algorithm BBP-PA)
  - A Bayesian Memory (BM) provides a maximum Entropy data reduction with sparse, distributed output representations



Bayesian Belief Propagation (Pearl)

- A node represents a variable and is part of a larger, acyclic graph
- Child regions Y_1 and Y_2, parent U

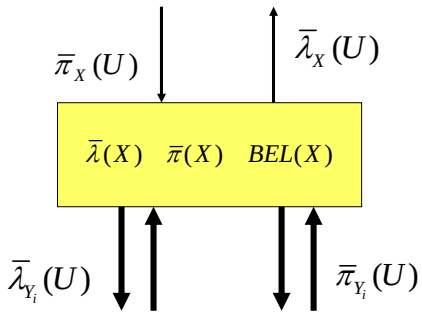
1) $\lambda(x_k) = \prod_j \lambda_{Y_j}(x_k)$

2) $\lambda(u_m) = \sum_x \lambda(x) P(x \mid u_m)$

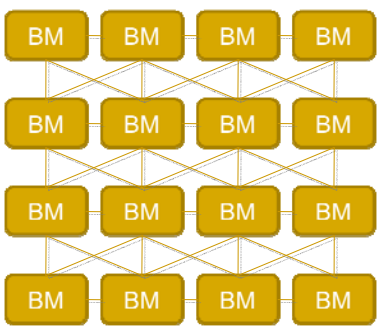
3) $\pi(x_k) = \sum_u P(x_k \mid u)\pi_X(u)$

4) $BEL(x_k) = \alpha\lambda(x_k)\pi(x_k)$

5) $BEL(x_k) = \alpha\pi(x_k)\prod_{i \neq j} \lambda_{Y_i}(x_k)$



- BM building block allows the creation of larger networks
- A BM sees only a subset of its input BMs and each BM's subset is slightly different – one way to distribute the representation
- One can implement exact BBP, but due to small granularity of BM modules, we are investigating various kinds of approximate inference



- The BM is a Bayesian Network, though using distributed representations, and so conceptually connects upward to more traditional Bayesian Networks
- Consequently we believe that the BM has the potential to implement complex varieties of Intelligent Signal Processing
- We also believe that the BM, in turn, can be implemented by a bidirectional associative array based on spiking neural models, and so conceptually connects downward to Computational Neuroscience
- The use of associative models creates an interesting range of architectural / hardware possibilities

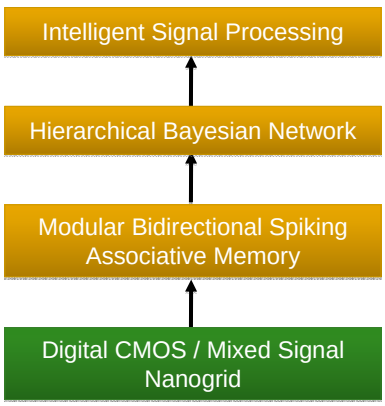
Spiking Associative Memory

- Neural systems capture and store spatial and temporal information, and then perform inference over such spatial/temporal information - however, adding temporal information to Bayesian structures is rarely straightforward
- One approach is to use more biologically plausible spiking models which inherently operate in the time domain
- Though more complex to use, spiking or pulse based models actually present an interesting opportunity at the hardware level:
- ***Computation proceeds by incremental change in response to spikes to a base-line state, where data are represented by the inter-pulse timing***

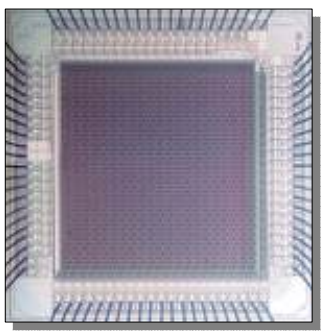
4. Hardware – Not Covered

Tuesday's talk: Hardware and Architecture For Efficient Inference

The "Big" Picture



- A hypothetical “cortical” processor
  - 22 nm, 8 metal CMOS / Nano-grid molecular arrays, 1 inch on a side, 10^{13} devices, 100 nm^2 CMOL memory cell
  - 10K processors fabricated, 11 peta-ops(10^{16})/sec
- These adapt rather than being programmed to perform real-time, adaptive Bayesian inference over very complex spatial and temporal knowledge structures
  - Implements arrays of Bayesian Memories (**FABA – Field Adaptable Bayesian memory Arrays**)



- Those of us working in this field believe that chips based on these models have the potential to be the microprocessor of the 21st century!

Portland State UNIVERSITY

5. Workshop on Massively Parallel Adaptive Computing

Portland State UNIVERSITY

- Held at Portland State University March 2-3, 2009
- Co-sponsored by:
  - Portland State
  - Intel
  - The National Science Foundation
  - The Office of Naval Research
- ~ 40 participants
- Presentations: http://www.technologydashboard.com/adaptivecomputing/

Speakers

- James Albus, “Reverse Engineering The Human Visual System”
- Misha Pavel, “Fusion-Based Robust Signal Processing by Humans and Machines”
- Rick Granger, “Nonstandard engineering principles of brain circuits”
- Jim Anderson, “What can you do with your brain-inspired computer now that you’ve built it?”
- Dileep George, “A mathematical canonical cortical circuit model, that can help build future-proof parallel architectures”
- Greg Hornby, “The ALPS-EA for Robust, Massively Parallel Optimization”
- Greg Snider, “Stable learning in networks of unreliable, memristive nanodevices”
- Karlheinz Meier, “VLSI Implementations of Very Large Scale Neuromorphic Circuits - Achievements, Challenges and Hopes”
- Kwabena Boahan, “Neurogrid: Emulating a million neurons in the cortex”

- Bruce Schachter, “Neuromorphic Target Cuer”
- Bob Thibadeau, “When the storage device becomes the computer”
- Pradeep Dubey, “Massive Data Computing”
- Craig Rasmussen, “PetaVision: A Software Architecture for Performing Petascale Simulations of Visual Cortex”

Breakout Discussions & Write-Up

- Programmability of Existing Approaches
- Scalable Algorithms and Applications
- Platform and Silicon Architectures
- Technology Gaps & Potential Solutions

Portland State UNIVERSITY

6. DARPA SyNAPSE Program

Maseeh College of Engineering
and Computer Science

DARPA DSO

DEFENSE SCIENCES OFFICE

Systems of **N**euromorphic **A**daptive **P**lastic **S**calable **E**lectronics

Bidder's Workshop and Teaming Meeting

March 4, 2008

Dr. Todd Hylton, Program Manager
DARPA DSO

DEFENSE SCIENCES OFFICE

Historical Evolution of Modern Electronics

Transistor → IC → µProcessor & memory → Programmable machines

60 years

- End of scaling
- Defect intolerant
- Architectural bottleneck
- Software limited
- *No path to biologically competitive intelligence*

DARPA SyNAPSE

Vision for the Future

Electronic Synapse → "Cortical" Microcircuit → "Cortex" Fabric → Intelligent machines

<<60 years

- Increased component density
- Increased component function
- Defect tolerant
- Neuromorphic information, learning, cognition, understanding architecture
- *Path to biologically competitive intelligence*

The SyNAPSE program seeks to extend the development of modern electronics into a new revolutionary new era using a similar paradigm.

DEFENSE SCIENCES OFFICE

Program Approach

System (SyNAPSE)

Modules (*e.g.* visual cortex)

Networks (*e.g.* cortical column)

Circuits (*e.g.* center-surround)

Components (*e.g.* synapse / neuron)

Materials (*e.g.* memristors)

Top-down (simulation)

Biological Scale Machine Intelligence

Bottom-up (devices)

Attack the problem "bottom-up" and "top-down" and force disciplinary integration with a common set of objectives.

Model

Make — **Measure**

Employ theoretical and empirical approaches constrained by practicality.

Architecture

Hardware — **Simulation**

Environment

Sponsor a suite of complementary capabilities to build, train, and evaluate devices.

DARPA DSO
DEFENSE SCIENCES OFFICE

Program Outline

	Phase 0	Phase 1	Phase 2	Phase 3	Phase 4
Hardware	Component synapse (and neuron) development	CMOS process and core circuit development	CMOS process integration	~10^6 neuron single chip implementation "Mouse" level	~10^8 neuron multi-chip robot at "Cat" level
Architecture & Tools	Microcircuit architecture development	System level architecture development	~10^6 neuron design for simulation and hardware layout	~10^8 neuron design for simulation and hardware layout	Comprehensive design capability
Emulation & Simulation	Preparatory studies only	Simulate large neural subsystem dynamics	"Mouse" level benchmark (~ 10^6 neuron)	"Cat" level benchmark (~ 10^8 neuron)	
Environment	Preparatory studies only	Build Sensory, Planning and Navigation environments "Small mammal" complexity	Add Audition, Proprioception and Survival "All mammal" complexity	Add Touch and Symbolic environments	Sustain

Program Phases 1-4 may be combined per the BAA instructions

Portland State UNIVERSITY

7. ITRS Emerging Research Devices / Architectures

ITRS - International Technology Roadmap for Semiconductors – Emerging Research Devices (ERD) & Architectures (ERA)

- In 2008 the NSF, the SRC, and Sandia Labs sponsored a workshop, "2020 Virtual Immersion Architectures" which explored architecture opportunities in a hypothetical virtual immersion application
- As a part of this effort, multi-core and, "morphic" architectures, were discussed as well as non-traditional computational models:
  - "Morphic architectures embrace a broad class of mixed-signal systems that focus on a particular application and draw inspiration for their structure from the application. In some cases, processing is carried out in the analog domain, offering orders-of-magnitude improvement in performance and power dissipation, albeit with reduced accuracy," Cavin, R., et al., *Emerging Research Architectures.* IEEE Computer, 2008. **41**(5): p. 33-37..

- The current plan for the ERA section of the ERD Chapter for the 2009 Roadmap is for 3 sections:
  - Memory architectures
  - Benchmarking of proposed devices based on the MIND, Mid-west Institute for Nanoelectronics Discovery, and
  - Inference architectures
- Workshop: Architectures for Post-CMOS Switches, August 18, Notre Dame
  - http://mind.nd.edu/

8. Conclusion

- We do not claim that by creating generic implementations of very large BMs duplicates the brain or, in and of itself, implements Cognition
- But we do believe that it (and related research projects) may take us further along in implementing better ISP for a range of applications
- And these models have significant promise in developing a set of hardware optimizations
- There is a strong sense by many that the field of computer architecture has stagnated
- **Radical new models of computation create opportunities for radical new architectures**

email: strom@ece.pdx.edu
my page: http://www.ece.pdx.edu/~strom