Neuromorphic Computing

Rick Stevens and Ian Foster

CMSC 35350

Ryerson 276, 9:30-10:50, Tuesday and Thursday

https://uchicago-cs.github.io/cmsc35350/

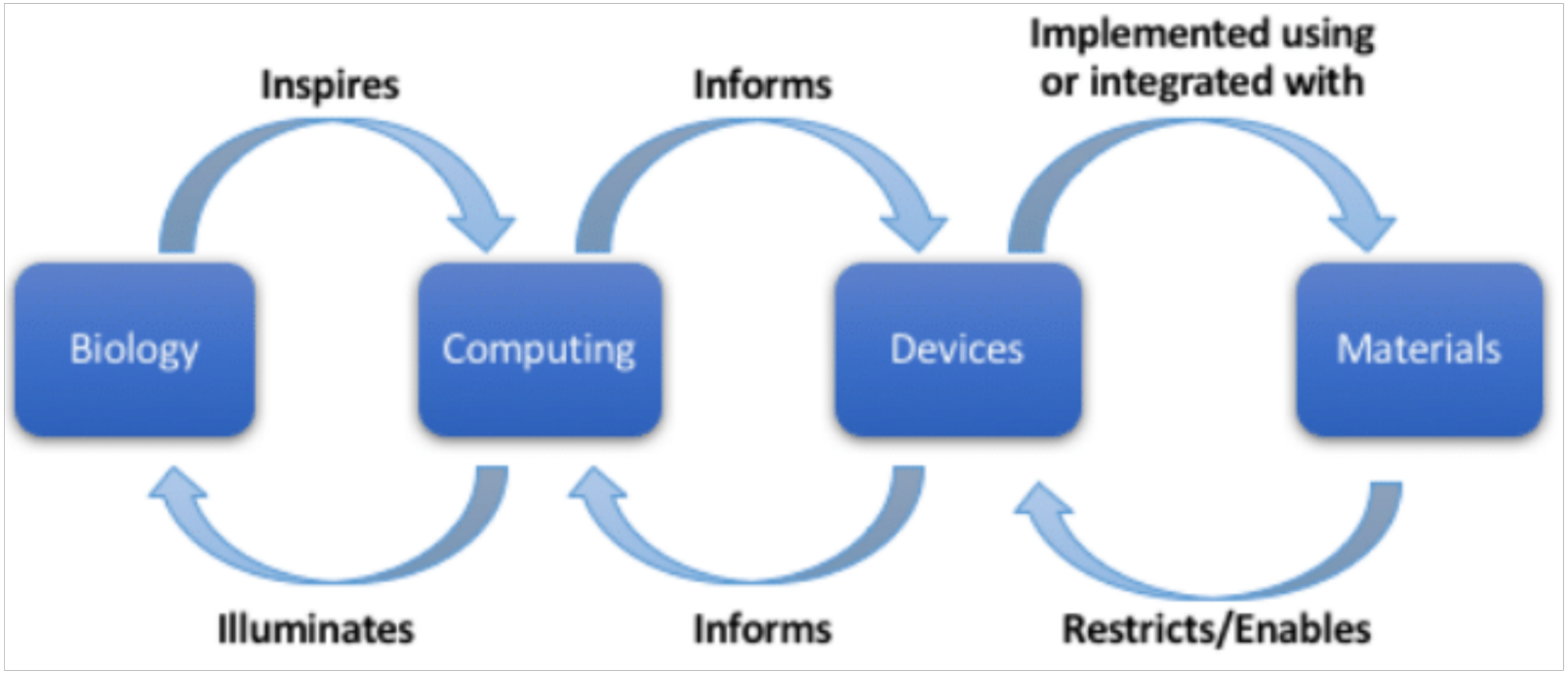
deep learning
Search term

neuromorphic
Search term

+ Add comparison

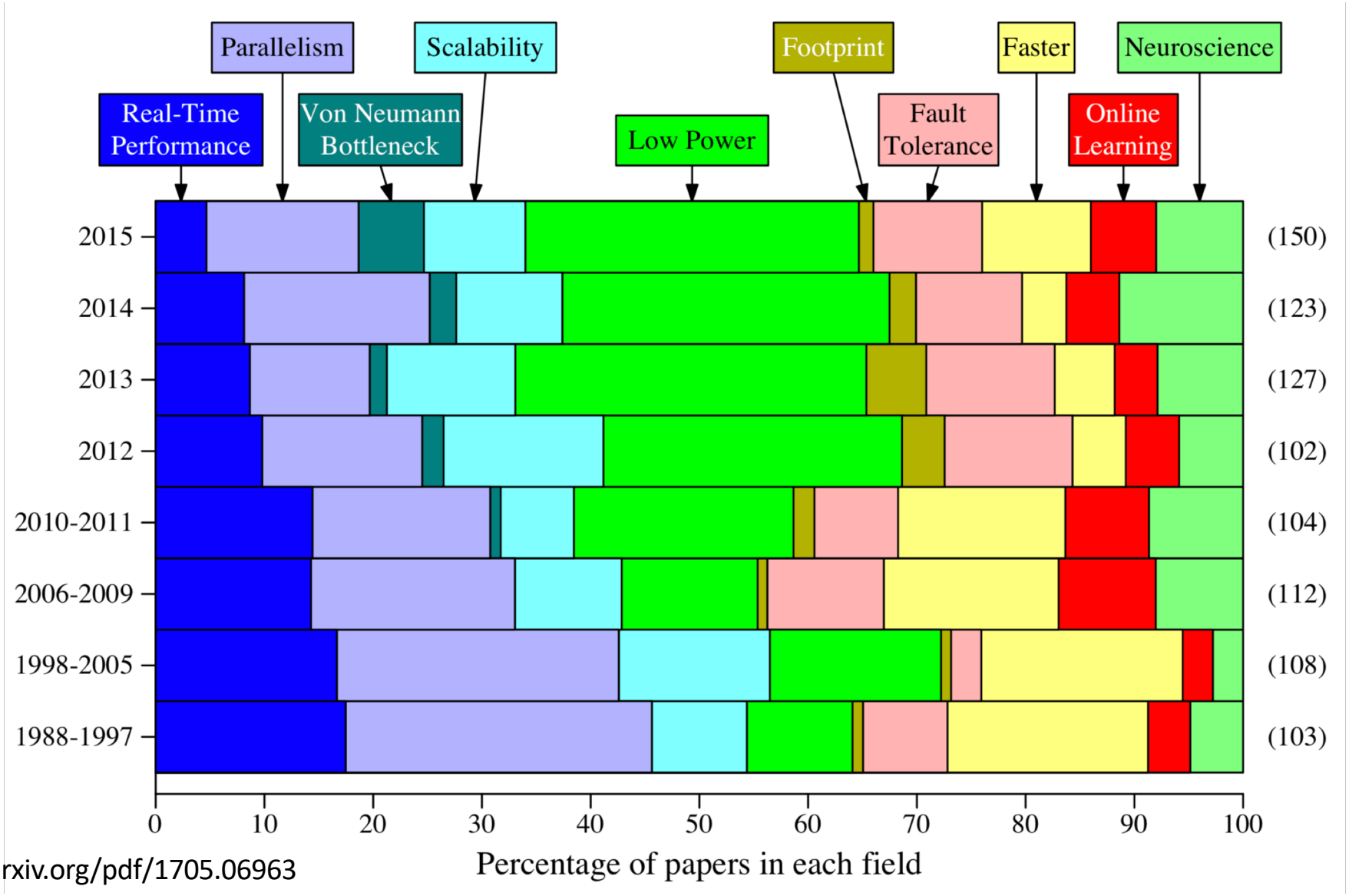
United States | Past 12 months | All categories | Web Search

Interest over time



THIS paper provides a comprehensive survey of the neuromorphic computing field, reviewing over 3,000 papers from a 35-year time span looking primarily at the motivations, neuron/synapse models, algorithms and learning, applications, advancements in hardware, and briefly touching on materials and supporting systems.

https://arxiv.org/pdf/1705.06963



https://arxiv.org/pdf/1705.06963

What is neuromorphic computing?

Transferring aspects of structure and function from biological substrates to electronic circuits

Structure
Cells – Networks – Connections

Function
Local Processing – Communication – Learning

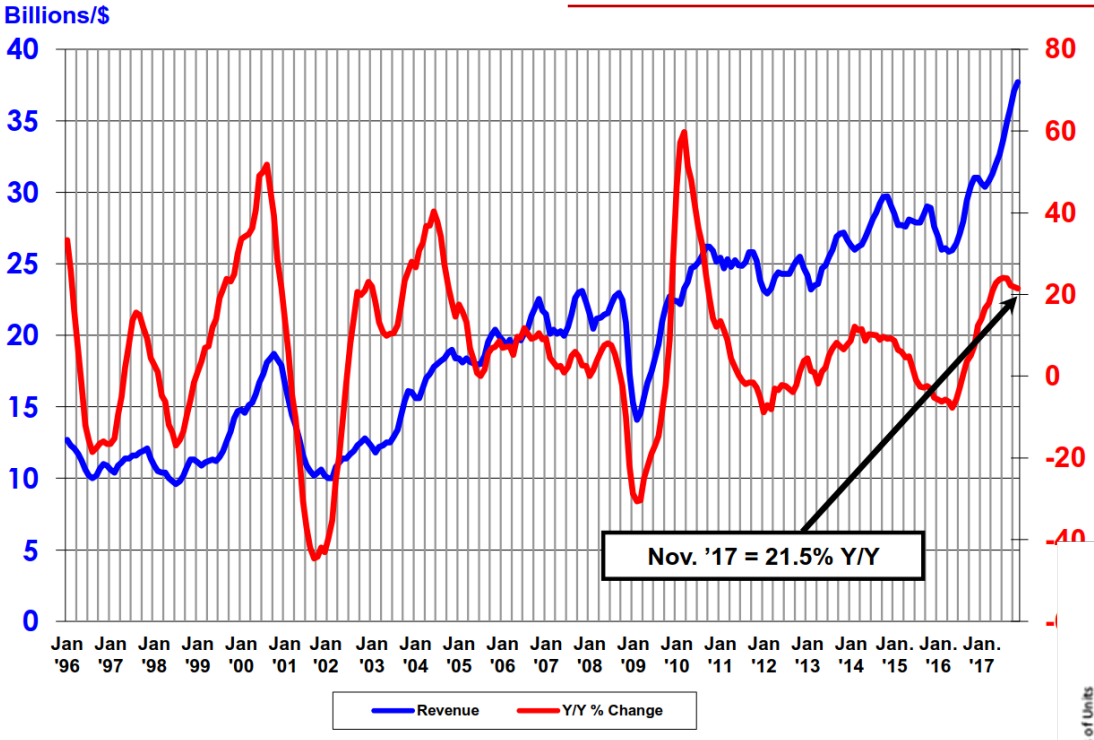
Why neuromorphic computing?

Modelling neural circuits to advance brain science

Applying brain-like principles to cognitive computing

Achieving energy efficiency, speed, robustness, ability to learn

General Semiconductor Market



Source: WSTS

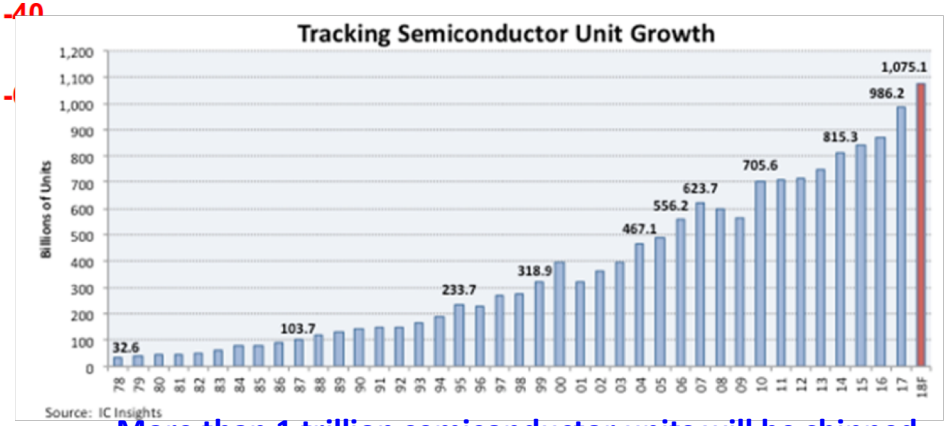
Total IT spending expected in 2018: 3.7 trillion $, 4.5% growth rate

Wafer supply companies have increased prices in 2017 and will increase by 20% in 2018 and further in 2019

Large jump (21.5%) in revenue and growth rate in 2017 which will partly continue in 2018
Major reason: RAM price increase by 120%

Total revenue per year has now crossed 400 B

Biggest semiconductor company is now Samsung (61 B$), followed by Intel (58 B$)
10 companies make >50% of revenues



More than 1 trillion semiconductor units will be shipped in 2018 (12% are microprocessors, DRAM, NAND, etc)

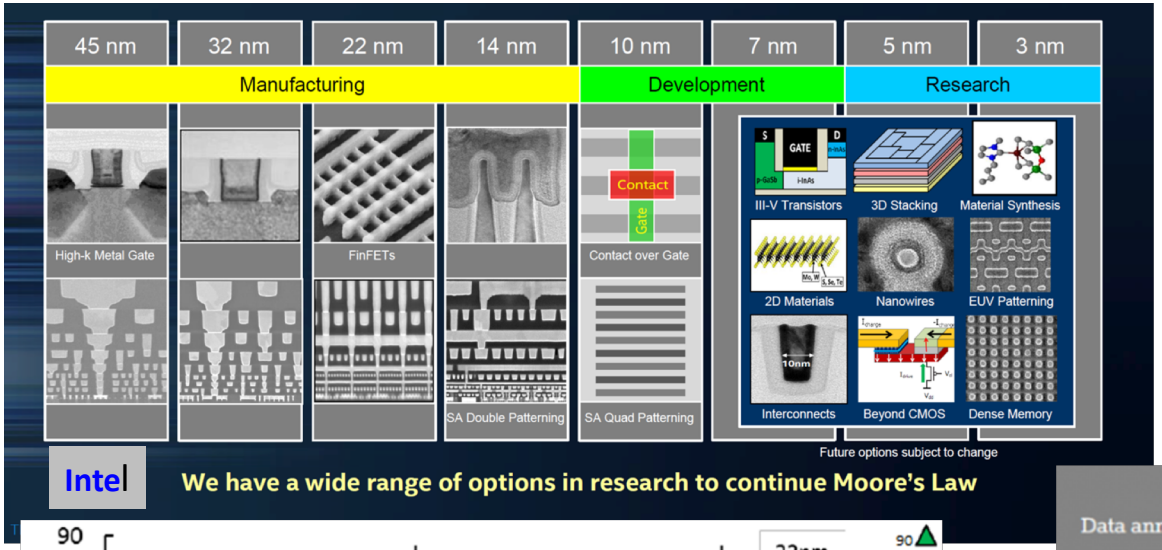
Processor Technology

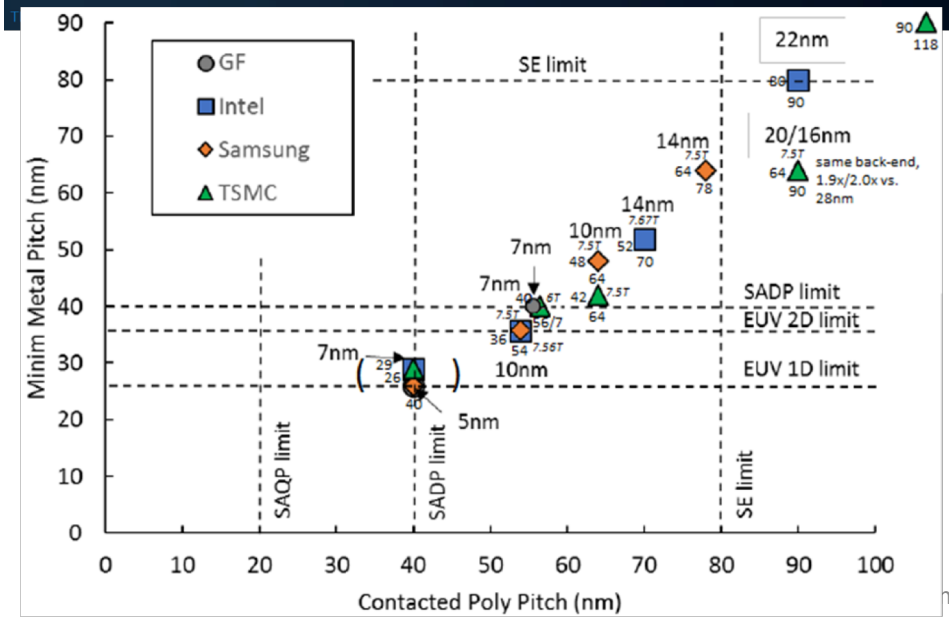
Intel has problems with their 10nm process

TSMC building fab 18 for their 5nm process,
Will be finished in 2020; 950000 m2 for 17 B$

There is no norm for the process names:
10 nm Intel compares to a 7 nm Samsung/TSMC process

Below 7nm new technologies are needed
(nanowires, non-silicon materials), very expensive





Industry FinFET Lithography Roadmap, HVM Start
Data announced by companies during conference calls, press briefings and in press releases

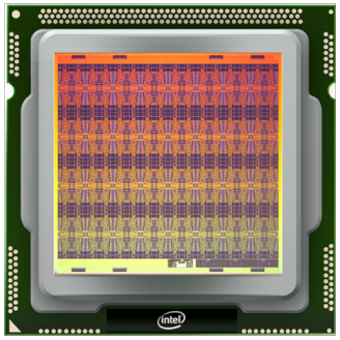
	2016	2017		2018		2019		2020		2021
		1H	2H	1H	2H	1H	2H	1H	2H	
GlobalFoundries	14LPP				7nm DUV	7nm with EUV*				
Intel	14 nm 14 nm+		14 nm++ 10 nm			10 nm+ 10 nm++				
Samsung	14LPP 14LPC	10LPE		10LPP		8LPP 10LPU	7LPP		6 nm* (?)	
SMIC	28 nm**	14 nm in development								
TSMC	CLN16FF+ CLN16FFC	CLN10FF CLN16FFC		CLN7FF CLN12FFC		CLN12FFC/ CLN12ULP	CLN7FF+		5 nm* (?)	
UMC	28 nm**		14nm	no data						

*Exact timing not announced
**Planar

New Processor Architectures

There is a plethora of new processor designs, all with a focus on Machine Learning:

- **Intel: Mobileye EyeQ5 (vision processing, autonomous cars), Nervana Neural Network Processor, Movidius MyriadX VPU**
- **ARM: Project Trillium, Machine Learning processor, Object Detection processor**
- **Graphcore IPU (Intelligent Processing Unit)**
- **Google second generation of Tensor Processing Unit TPU**
- **NeuPro AI processor from CEVA**
- **Neuromorphic chips from IBM (TrueNorth, 64 M neurons + 16 B synapsis) and Intel (Loihi, 130 K neurons + 130 M synapsis)**
- **Nvidia is enhancing their graphics cards, Titan V (110 Tflops Deep Learning), Xavier (SoC, 20 TOPS, vision accelerator)**



All high-end smartphones are integrating AI chip enhancements (Qualcomm-neural processing engine, Apple- A11 Bionic chip, etc.)
The market for these special chips will reach 5-10 B$ in 2022

The keyword is LOCAL data processing also major impact on IoT
→ much less network, cloud storage and cloud processing needed

Table A1. Selected component technology vectors that have a high probability of deployment in ten years

(* denotes more speculative deployment within this timeframe)

Component technology vector	Time-frame to first commercial products	Approach to achieving and retaining competitive advantage
Neuromorphic Computing	Available now	Continued R&D into new architectures coupled with 3D technologies and new materials, Deep Learning accelerators (for mobile and data center applications), and applications for true brain-inspired computing
Photonics	Available now	Foundries for tools and materials R&D; integrate photonics with CMOS and other materials
Sensors	Available now	Foundries for tools and materials R&D; integrate new types/classes of sensors with CMOS and other materials
CMOS (sub 7nm node size or new 3D structures)*	Advances in thermal management available with new process nodes	Deep understanding of transistor physics and chipset architecture and related design know-how; foundries and labs for transistor and materials R&D
Magnetics	1-2 years (MRAM as eFlash), 3 years (as DRAM), 5-7 years (as SRAM)	Foundries for tools and materials R&D; integrate magnetics with CMOS and other materials
3D	2-3 years (wafer-to-wafer stacking), 4-5 years (die-to-wafer stacking), 5-7 (Monolithic 3D)	Deep understanding of applications space and benefits associated use of 3D technologies and design know-how; foundries for tools and materials R&D; design automation tool R&D
Data-flow based architectures	3-4 years	Continued architecture R&D, coupled with materials, integration, and manufacturing; build an ecosystem for solutions using data-flow based architectures
Ultra-high performance wireless systems	3 years (5G), 10-12 years (6G)	Continued R&D in new materials and processes, antenna design advances, chipset manufacturing, and integration
Advanced non-volatile memory as	5+ years	Deep understanding of applications space and chipset architectures

Motivation: The Case for Neuromorphic Computing

Problem Statement:

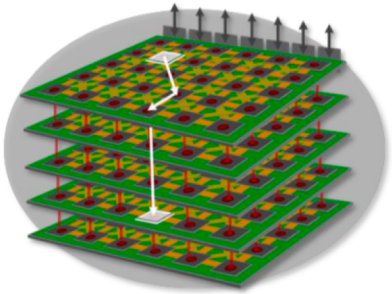
Emerging computing workloads demand *intelligent behaviors* that we do not know how to deliver *efficiently* with today's algorithms and computing architectures.

Examples:

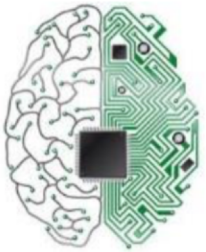
- Online and lifelong learning
- Learning without cloud assistance
- Learning with sparse supervision
- Understanding spatiotemporal data
- Probabilistic inference and learning
- Sparse coding/optimization
- Nonlinear adaptive control
- Pattern matching with high occlusion
- SLAM and path planning

Potential Future Product Applications

Robotics



HPC Systems



Neuroprosthetics

Smart Glasses

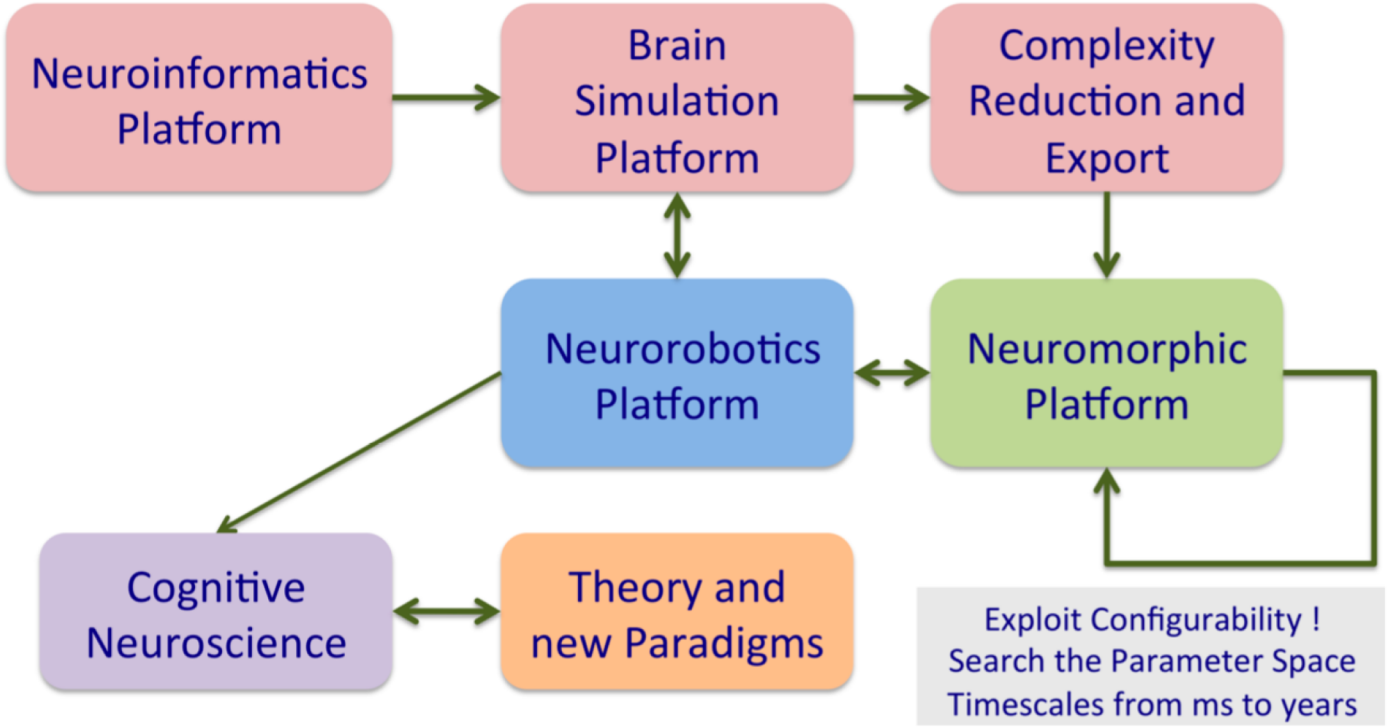
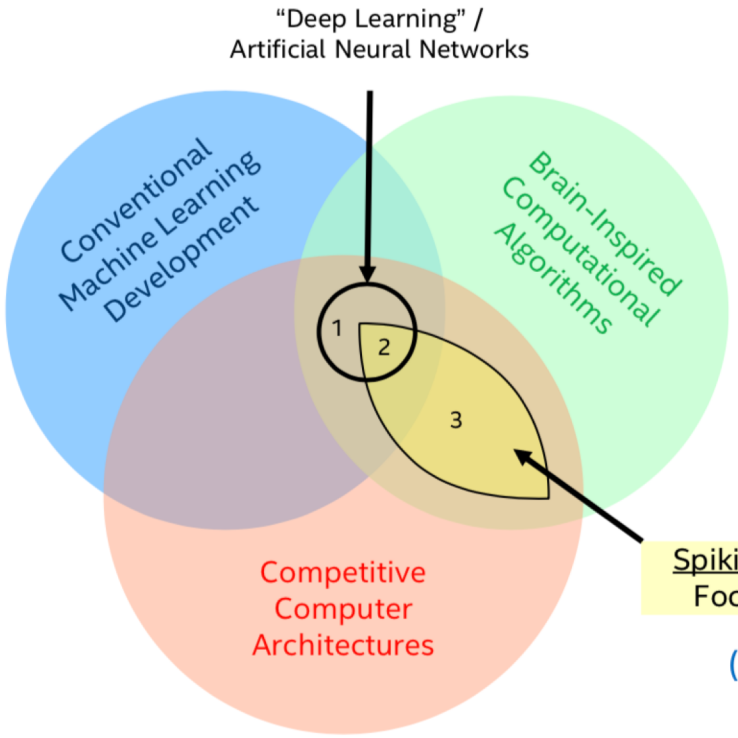


Figure .4: Integration of the Neuromorphic Computing Platform into the HBP Platform Ecosystem

Solution Exploration Space



Research Goals:

- **Broad class** of brain-inspired computation
- **Efficient** hardware implementations
- **Scalable** from small to large problems and systems

(We hope this is a non-empty class!)

The Engineering Perspective

- Nature has come up with something amazing. Let's copy it…
- Not so simple – very different design regimes
- Yet objectives and constraints are largely the same…
  - Energy minimization
  - Fast response time
  - Cheap to produce

Need to understand and apply the basic principles, *adapting for differences*

Status today:

	Nature	Silicon	Ratio
Neuron density[1]	100k/mm^2	5k/mm^2	20x
Synaptic area[1]	0.001 um^2	0.4 um^2[2]	400x
Synaptic Op Energy	~2 fJ	~4 pJ	2000x

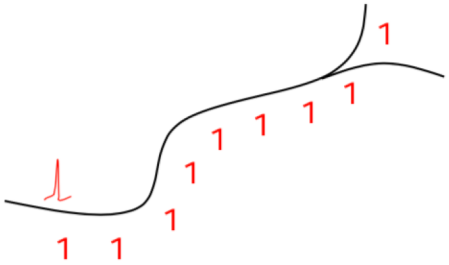
[1] Planar neocortex [2] ~5b SRAM

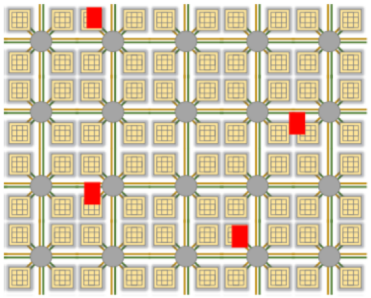
But…

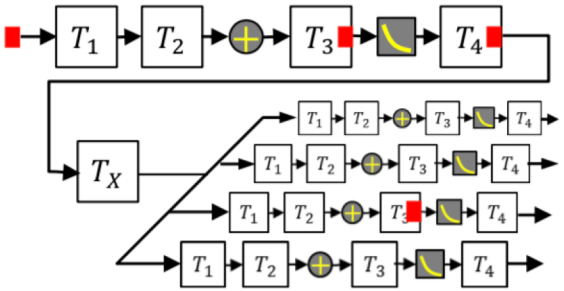
Max firing rate	100 Hz	1 GHz	10,000,000x
Synaptic error rate	75%	0%	∞

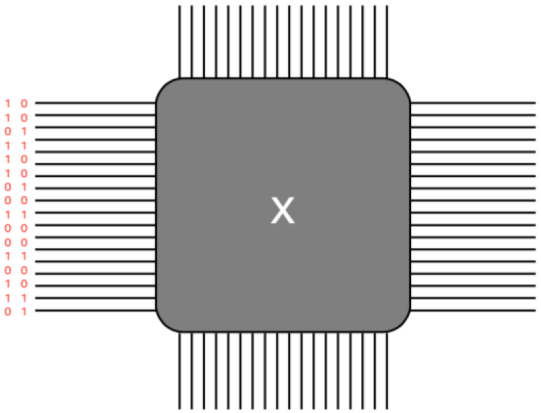
Nature	Silicon
Autonomous self-assembly	Fabricated manufacturing
Per-instance variability desired	Variability causes brittle failures
lasticity over lifetime	Must support rapid reprogramming
rministic operation	Deterministic operation desired

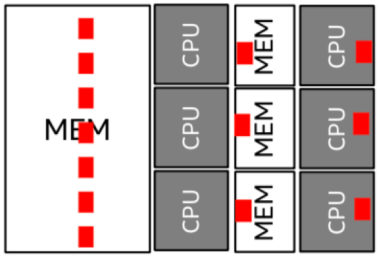
Are Spiking Architectures Efficient?

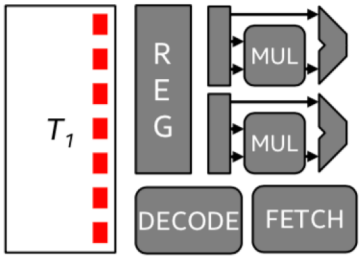




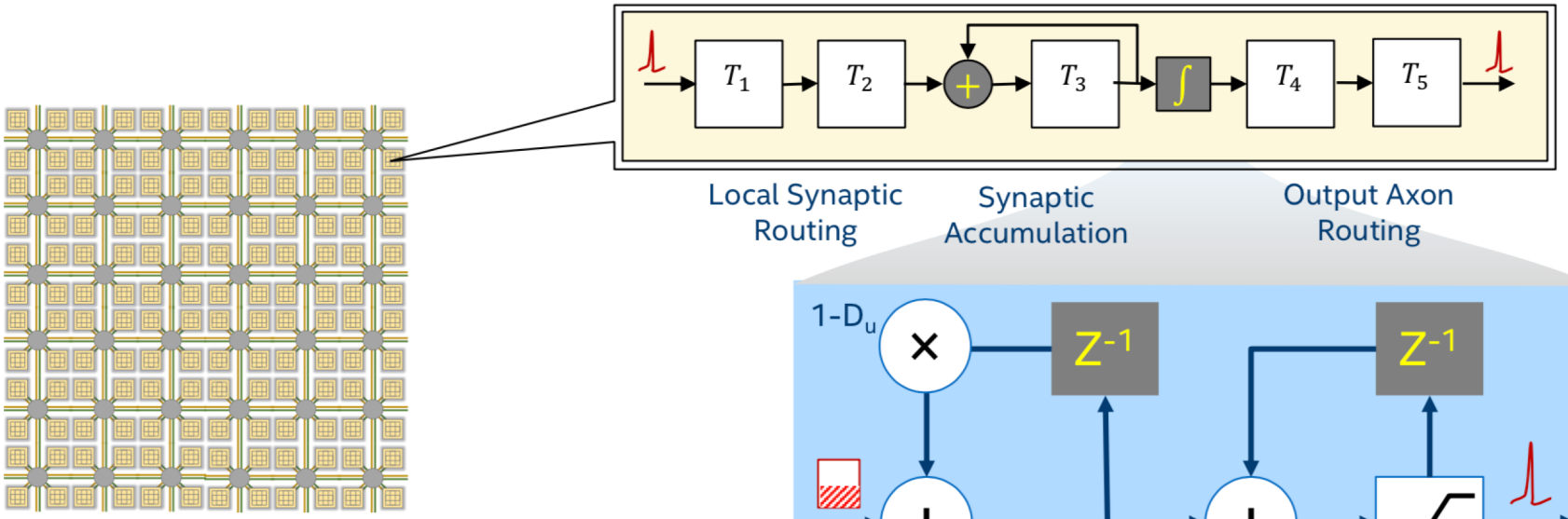






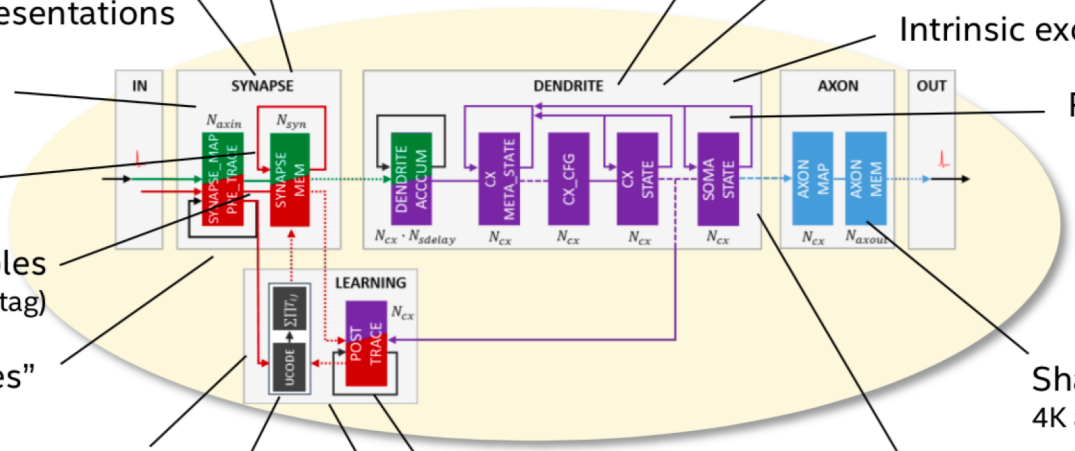


What this gives us… a baseline SNN architecture



2D Mesh
Packetized spikes
High fanout required
Low overhead synchronization

Neuromorphic Core Architecture



All synaptic connections pooled
128KB shared memory

Sparse, *dense*, and *hierarchical*
Synaptic mapping representations

Synaptic delays

Synaptic eligibility traces

Flexible 3-tuple synaptic variables
(1-9b weight, 0-6b delay, 0-8b tag)

Graded "reward spikes"

Flexible synaptic plasticity with microcode-programmable rules

Sum-of-products rule semantics

Filtered spike train traces

Plasticity rules target any synaptic variable

Discrete time LIF neuron model (CUBA)

Multi-compartment dendritic trees
up to 1K compartments

Intrinsic excitability homeostasis

Random noise sources

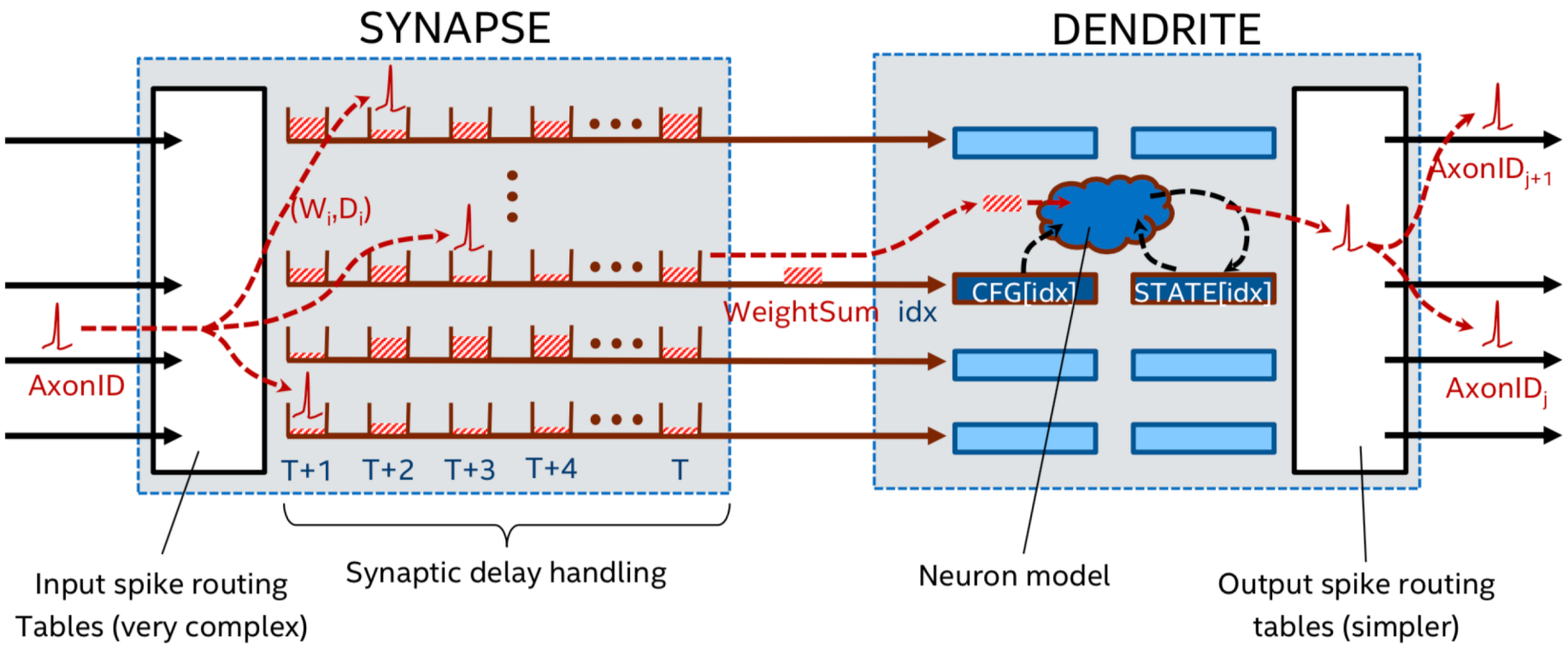
Shared output routing table
4K axon routes

Axon delays

Refractory delays (+ random)

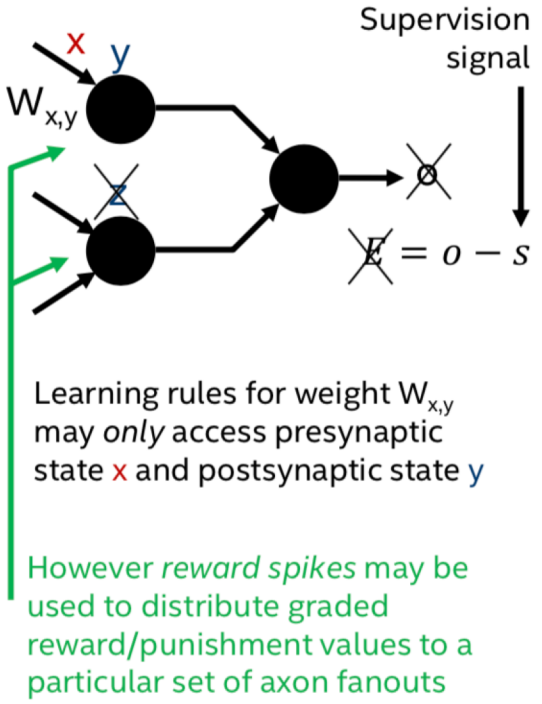
Basic Core Operation (Non-Learning)

(Time multiplexing illustrated unrolled in space)



Learning with Synaptic Plasticity

- **Local learning rules** – essential property for efficient scalability
  Compatible with biological plausibility
- Should be derived by **optimizing an emergent statistical objective**
  Too much directionless experimentation otherwise
- Plasticity on **wide range of time scales** is needed
  Delayed reward/punishment responses, eligibility traces



Learning Rule Examples

Pairwise STDP:

$$W(t+1) = W(t) - A\_x_0(t)y_1(t) + A_+x_1(t)y_0(t)$$

Triplet STDP with heterosynaptic decay:

$$W(t+1) = W(t) - A\_x_0(t)y_1(t) + A_+x_1(t)y_0(t)y_2(t) - B \cdot W(t) \cdot y_3(t)$$

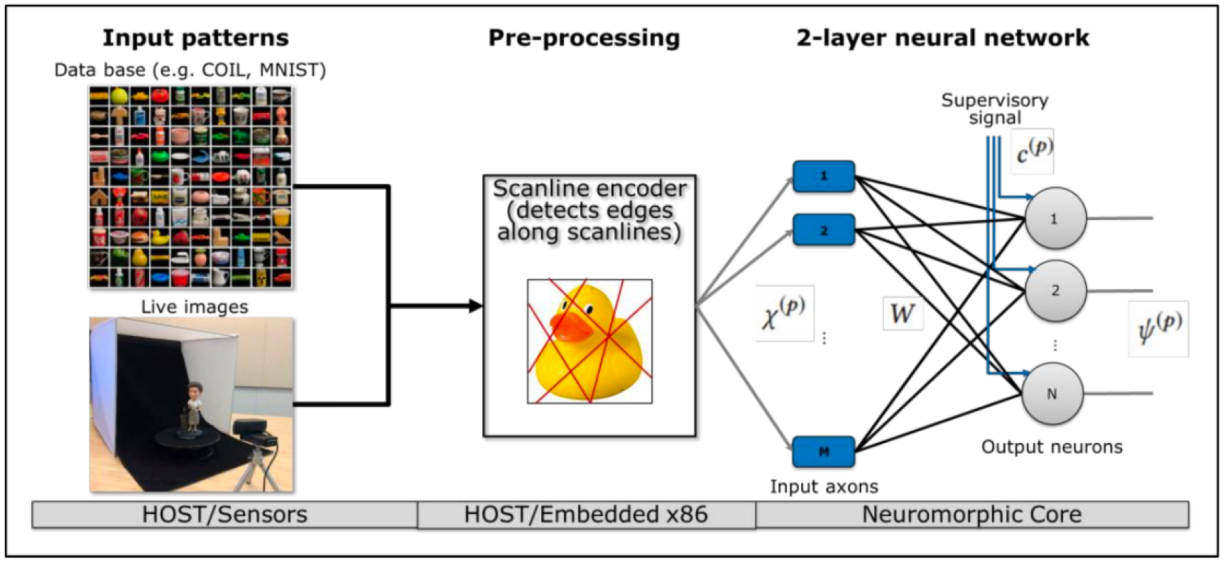
Delay STDP:

$$D(t+1) = D(t) - A\_x_0(t)(127 - y_1(t)) + A_+(127 - x_1(t))y_0(t)$$

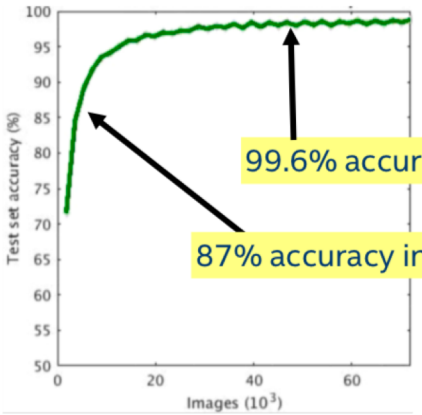
Encoding	*Term* (T_{ij})	Bits	Description
0	$x_0 + C$	5b (U)	Presynaptic spike count
1	$x_1 + C$	7b (U)	1st presynaptic trace
2	$x_2 + C$	7b (U)	2nd presynaptic trace
3	$y_0 + C$	5b (U)	Postsynaptic spike count
4	$y_1 + C$	7b (U)	1st postsynaptic trace
5	$y_2 + C$	7b (U)	2nd postsynaptic trace
6	$y_3 + C$	7b (U)	3rd postsynaptic trace
7	$r_0 + C$	1b (U)	Reward spike
8	$r_1 + C$	8b (S)	Reward trace
9	wgt+C	9b (S)	Synaptic weight
10	dly+C	6b (U)	Synaptic delay
11	tag+C	9b (S)	Synaptic tag
12	sgn(wgt+C)	1b (S)	Sign of case 9 (±1)
13	sgn(dly+C)	1b (S)	Sign of case 10 (±1)
14	sgn(tag+C)	1b (S)	Sign of case 11 (±1)
15	C	8b (S)	Constant term. (Variant 1)
15	$S_m \cdot 2^{S_e}$	4b (S)	Scaling term. 4b mantissa, 4b exponent. (Variant 2)

TABLE 1: Learning rule product terms

Our “Hello World” Application: Supervised Learning for Object Recognition



Performance on COIL20 data set



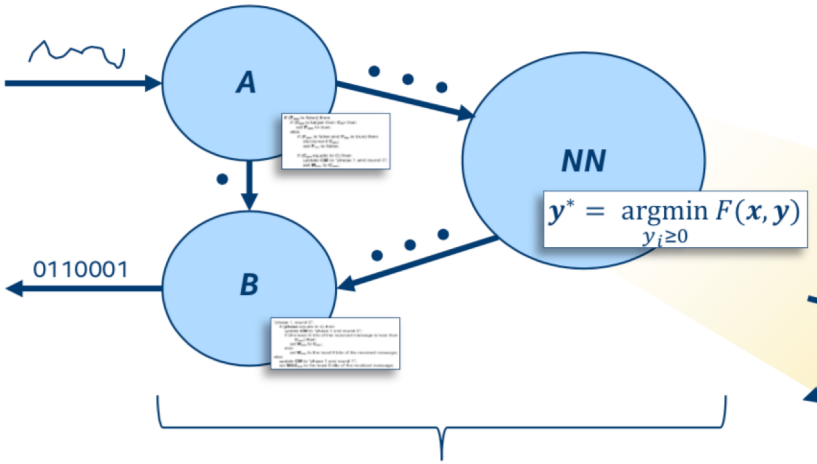
	Training	Inference
Active energy per image (total)	553 uJ	128 uJ
Neuromorphic energy	322 uJ	13 uJ
Processing time per image	7.5 ms	1.8 ms
Chip power	74 mW	73 mW
Neuromorphic power	43 mW	7.4 mW

Resource Utilization	Count	Utilization
Neurons	20	0.02%
Synapses	38400	0.28%
SNN Cores	1	078%

S-STDP rule:

$$W_{i,j}(t) = W_{i,j}(t-1) + \eta \cdot \left(u_K \cdot \delta_{i,C(p)} - y_{i,0}\right) \cdot x_{j,1}$$

Up to the 10,000 foot view



Modules w/ behavioral models

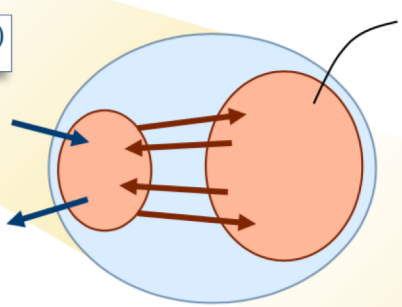
A, B: Sequential processes conventionally coded and run on conventional CPUs

NN: Neural network module

- Hierarchically specified
- Mathematical behavioral model
- May include conventional helper code ("snips")

The Nx System Framework

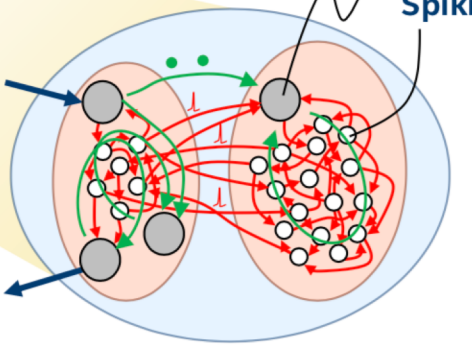
- Heterogeneous hierarchical parallel system
- Event-driven communication over channels
- Localized state
- Models describe *emergent behavior*



SNN specification

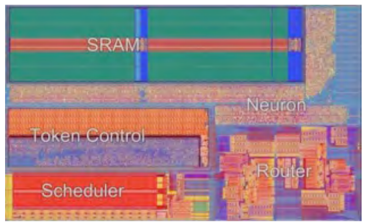
Snip
(Sequential neural interfacing process)

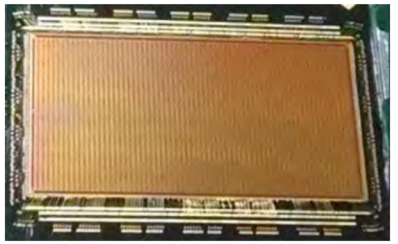
Spiking neuron

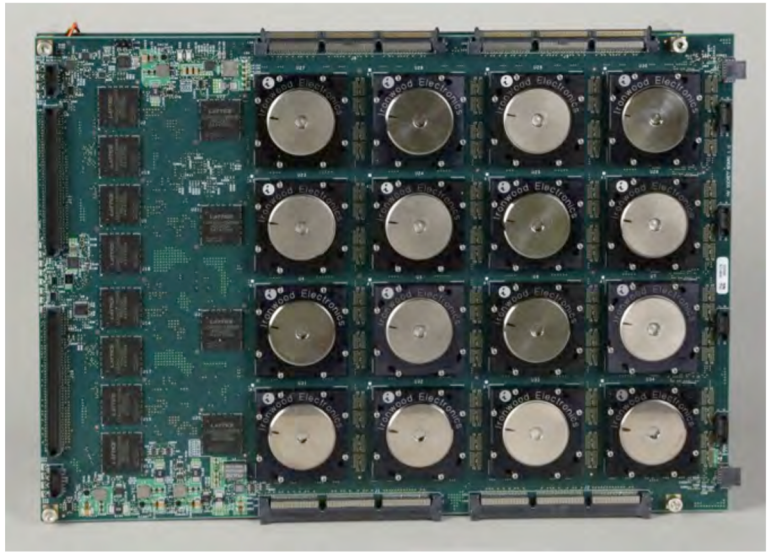


IBM TrueNorth

- 4,096 digital neurosynaptic cores
  - one million configurable neurons
  - 256 million programmable synapses
  - ~70mW
  - over 400 Mbits of embedded SRAM
  - 5.4 billion transistors
- 16 TrueNorth Chips assembled into a 4x4 mesh
  - 16 million neurons and 4 billion synapses.

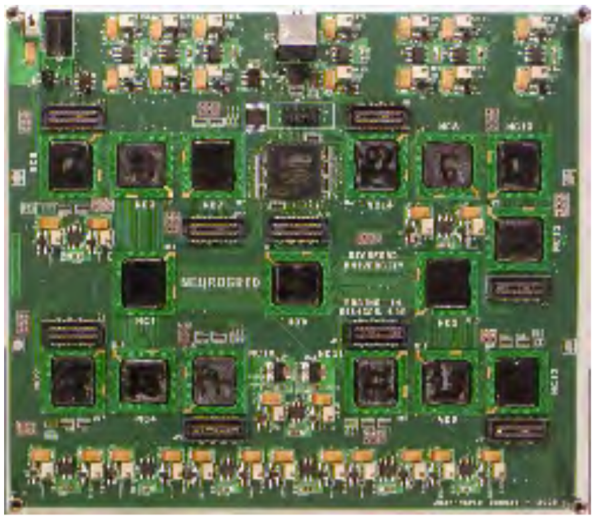






Stanford Neurogrid

- Neurocore Chip
  - 65k neurons
  - each with two compartments and a set of configurable silicon ion channels
- Sixteen Neurocores are assembled on a board
  - million-neuron Neurogrid



HBP Neuromorphic Computing Concepts

MANY-CORE NUMERICAL MODEL SYSTEM

0.5 – 1 Million ARM processors – address-based, small packet, asynchronous communication – running at real-time

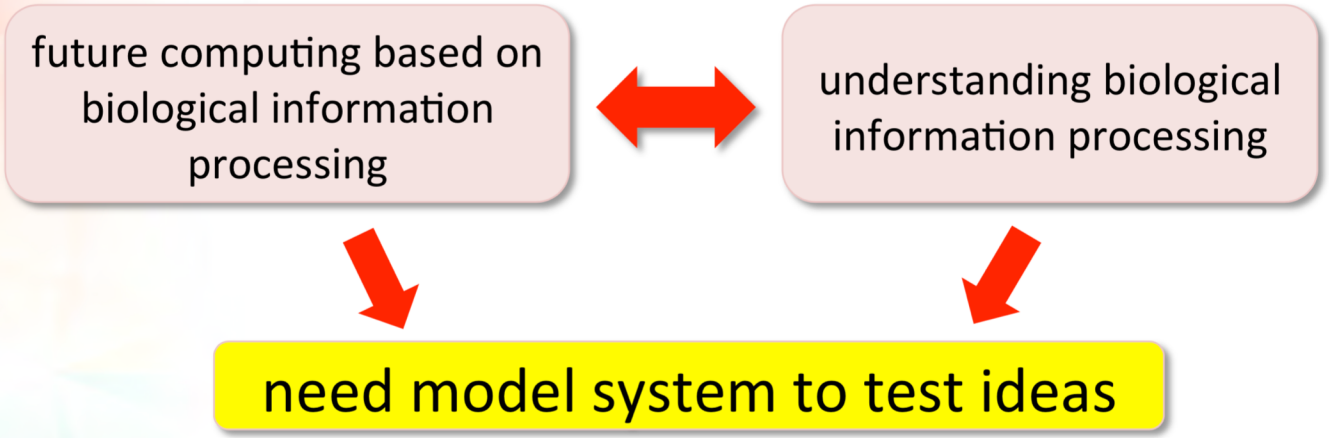
Location : Manchester (UK)

PHYSICAL MODEL SYSTEM

Local analog computing with 4 Million neurons and 50 Million synapses – binary, asynchronous communication – emulation speed is x 10 000 real-time

Location : Heidelberg (Germany)

Motivation and Approaches

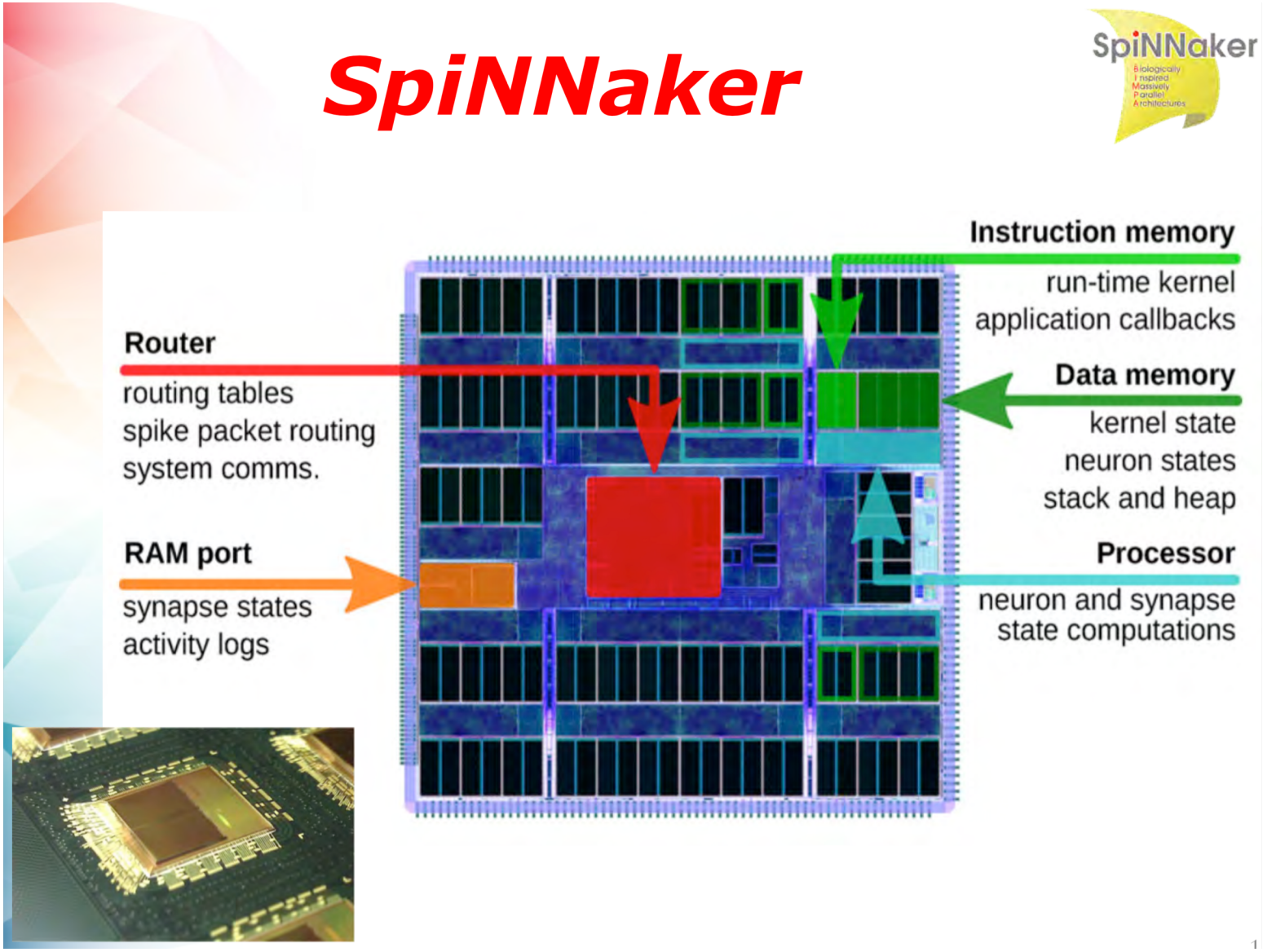


Two fundamentally different modeling approaches:

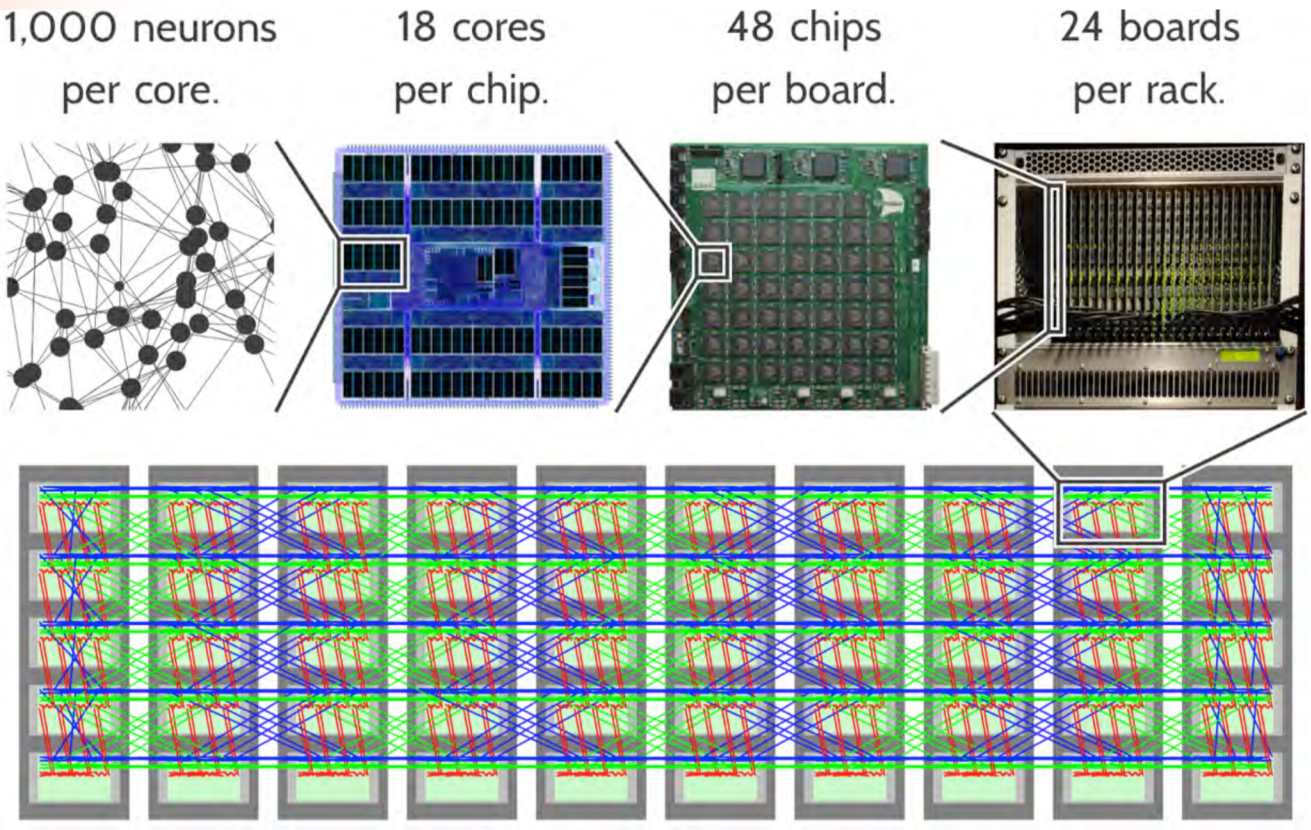
- **NUMERICAL MODEL (SpiNNaker)**
  represents model parameters as binary numbers
- **PHYSICAL MODEL (BrainScaleS)**
  represents model parameters as physical quantities
  → **voltage, current, charge** (like the biological brain)

can be combined to form a hybrid system

SpiNNaker


Instruction memory
run-time kernel
application callbacks
Router
routing tables
spike packet routing
system comms.
Data memory
kernel state
neuron states
stack and heap
RAM port
synapse states
activity logs
Processor
neuron and synapse
state computations

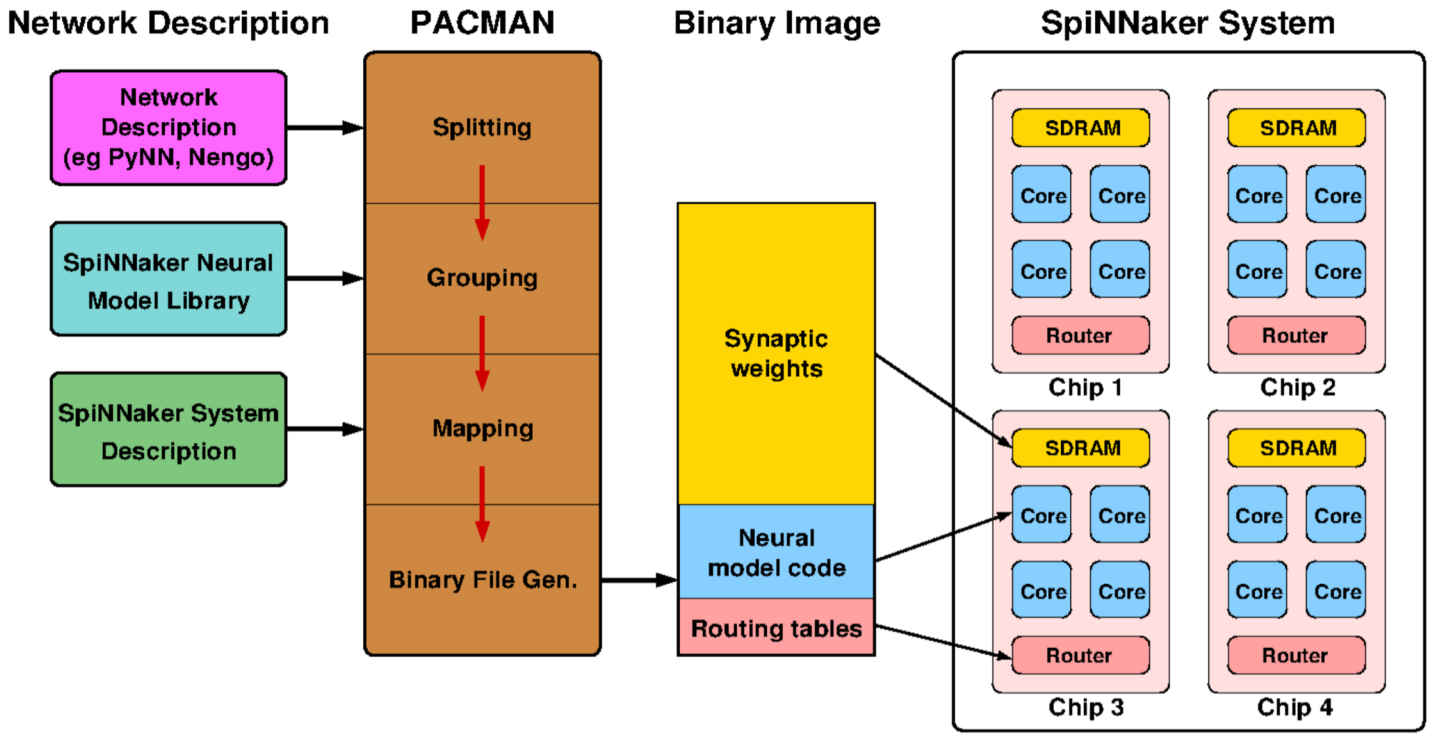
SpiNNaker


1,000 neurons per core.
18 cores per chip.
48 chips per board.
24 boards per rack.
5 racks per cabinet, 10 cabinets.
100kW

SpiNNaker

- HBP platform
  - 500,000 cores
  - 6 cabinets (including server)
- Launch
  - 30 March 2016

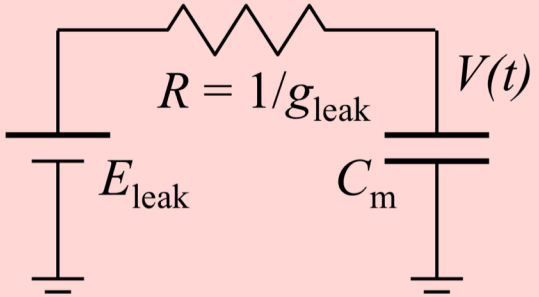
Problem mapping



Physical Model System

Continuous Time Integrating Neural Cell Membrane

$$C_m \frac{dV}{dt} = -g_{leak}\left(V - E_{leak}\right)$$



	ΔV [V]	g_{leak} [S]	C_m [F]	$(g\Delta V)/C$ [V/s]
Biology(*)	10^{-2}	10^{-8}	10^{-10}	10^{0}
VLSI	10^{-1}	10^{-6}	10^{-13}	10^{6}

Inherent speed gap:

10^6 Volt/second

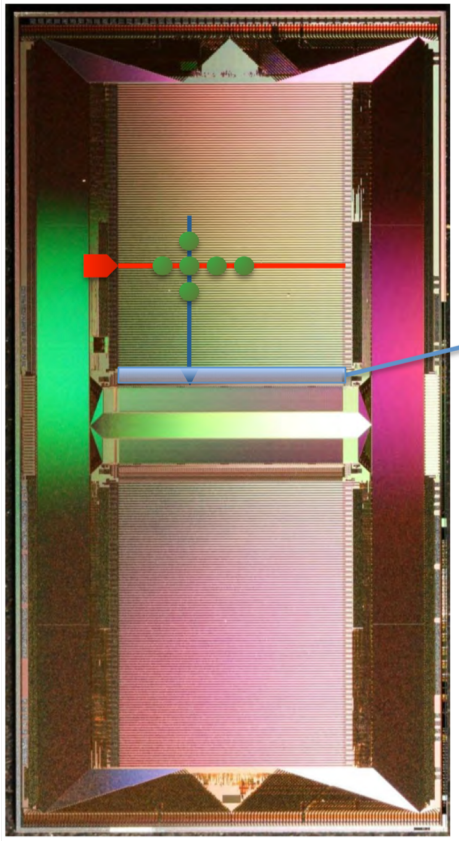
→ accelerated neuron model

(*) Brette/Gerstner, J. Neurophysiology, 2005

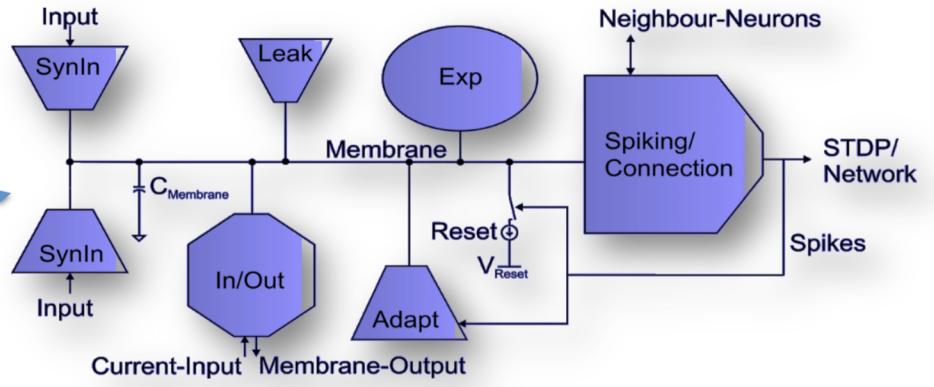
„Time“ is imposed by internal physics, not by external control

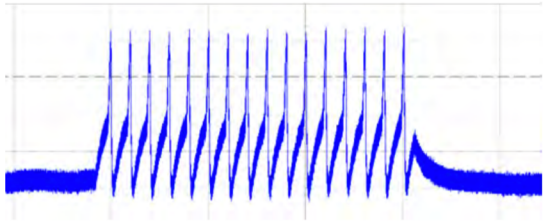
Structure of a BrainScaleS Chip

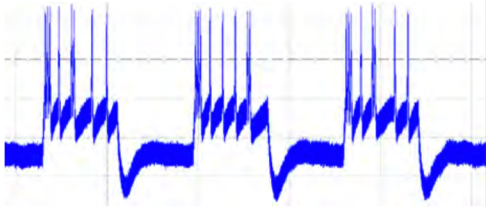
photograph of the
neuromorphic chip HICANN

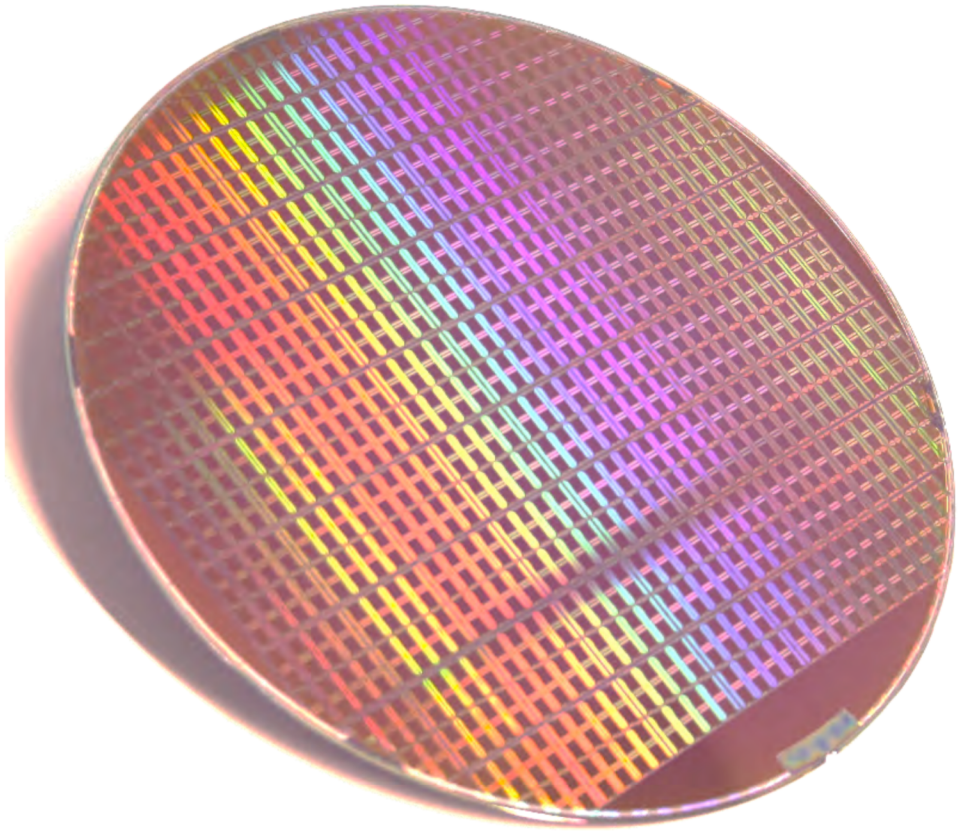


512 AdEx neurons
112650 programmable dynamic synapses
Short-term and long-term synaptic plasticity
10000x acceleration wrt. biological real-time







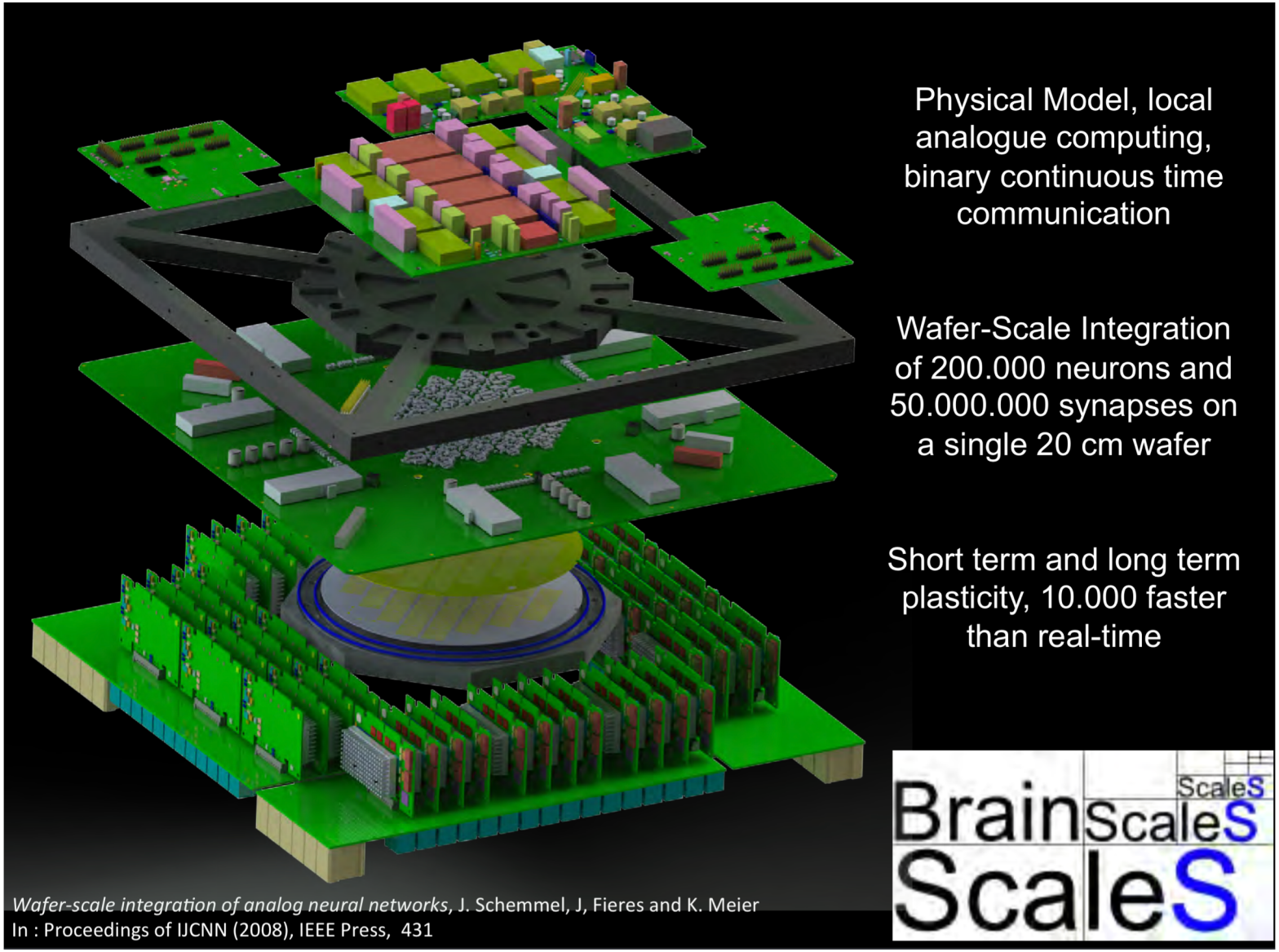


BrainScaleS

Central element

20cm silicon wafer with 450 neuromorphic Microchips

- analog, time-continuous operation
- 10000 faster than biological real time
- 50M synapse and 230k neuron circuits operate in parallel
- high-density on-wafer interconnects manufactured by multi-layer wafer-level metallization



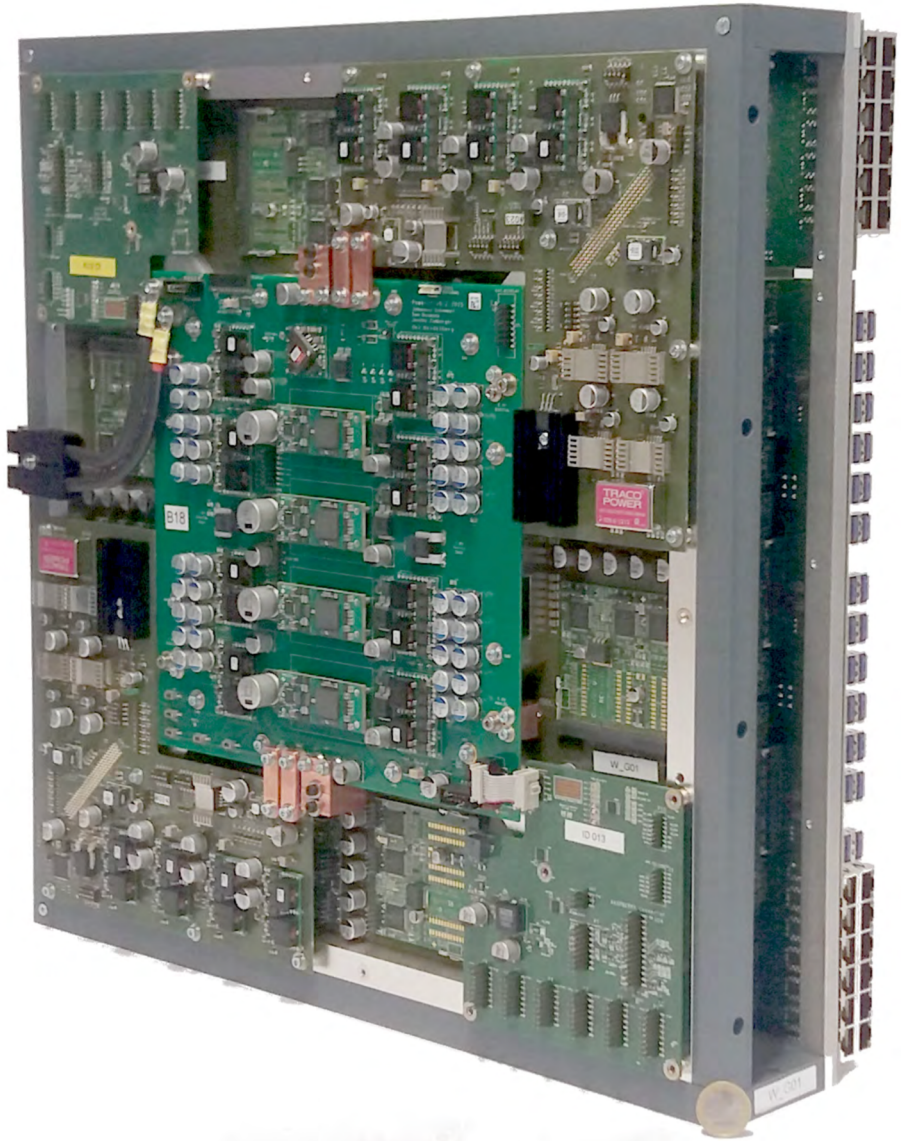
Physical Model, local analogue computing, binary continuous time communication

Wafer-Scale Integration of 200.000 neurons and 50.000.000 synapses on a single 20 cm wafer

Short term and long term plasticity, 10.000 faster than real-time

Wafer-scale integration of analog neural networks, J. Schemmel, J, Fieres and K. Meier
In : Proceedings of IJCNN (2008), IEEE Press, 431

BrainScaleS

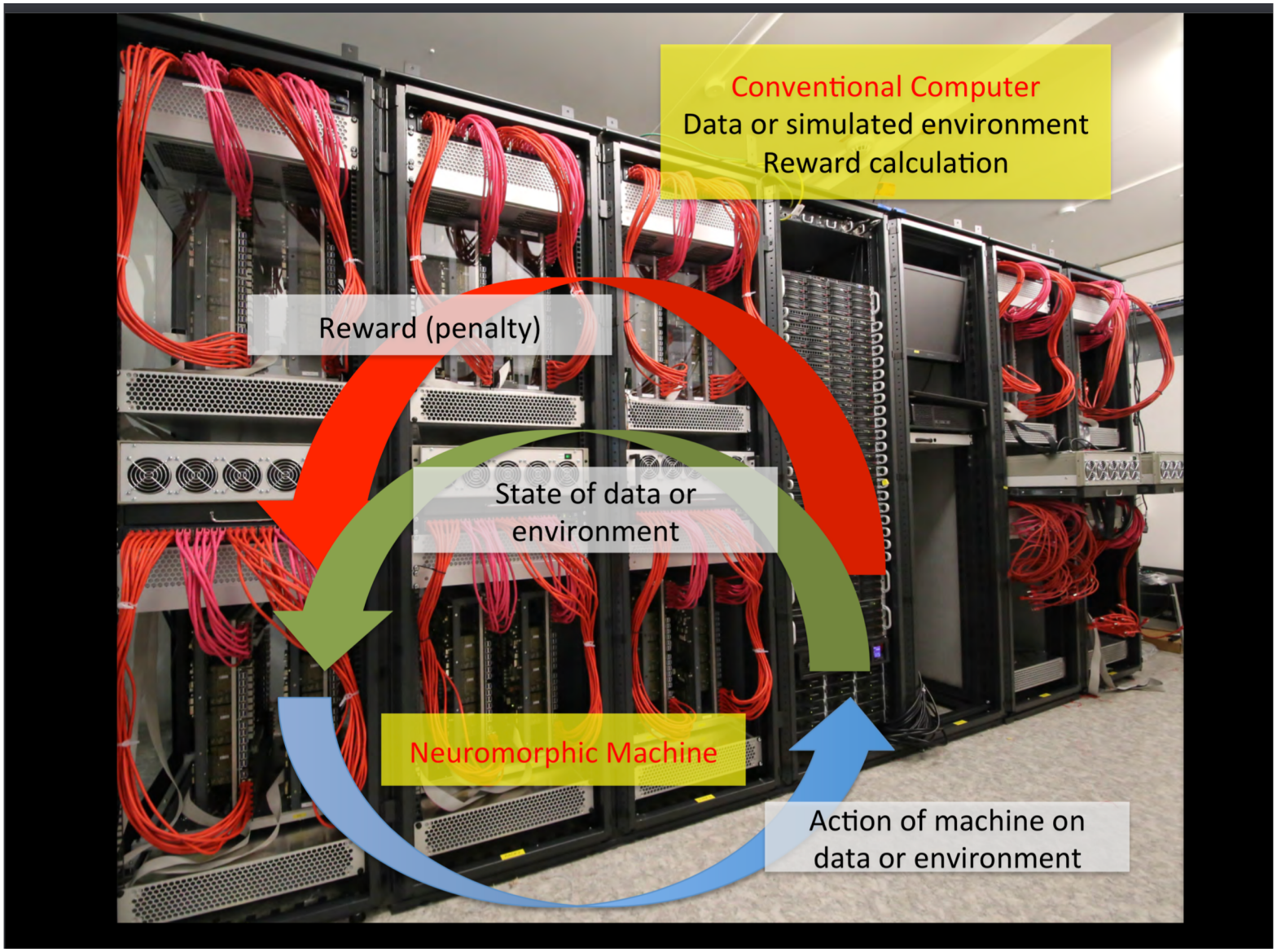


BrainScaleS
ScaleS

Physical Model, local analogue computing, binary continuous time communication

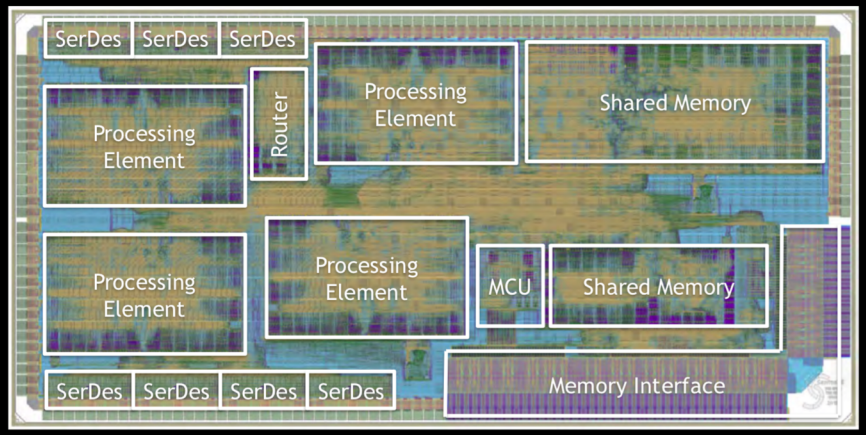
Wafer-Scale Integration of 200.000 neurons and 50.000.000 synapses on a single 20 cm wafer

Short term and long term plasticity, 10.000 faster than real-time


Conventional Computer
Data or simulated environment
Reward calculation
Reward (penalty)
State of data or environment
Neuromorphic Machine
Action of machine on data or environment

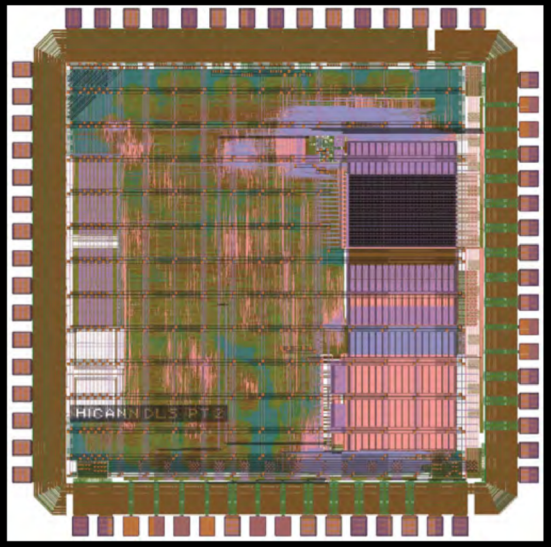
SpiNNaker-2

Power Management
Floating point precision
True random numbers



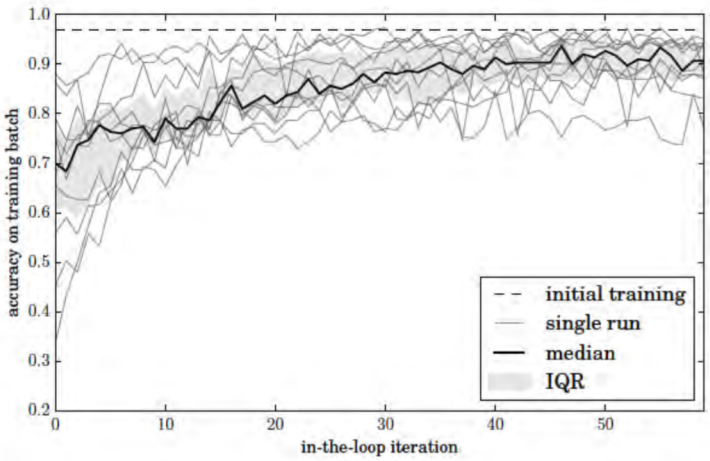
BrainScales-2

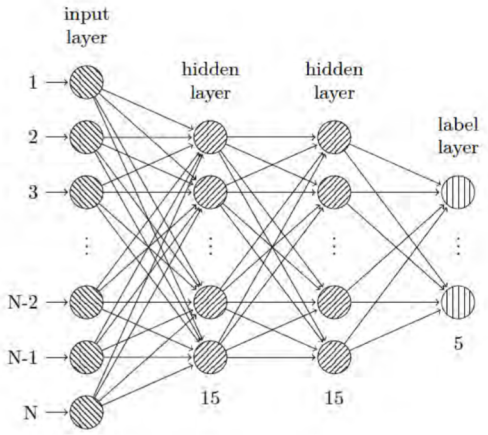
PowerPC plasticity processors
Improved parameter storage
Active dendrites / compartments

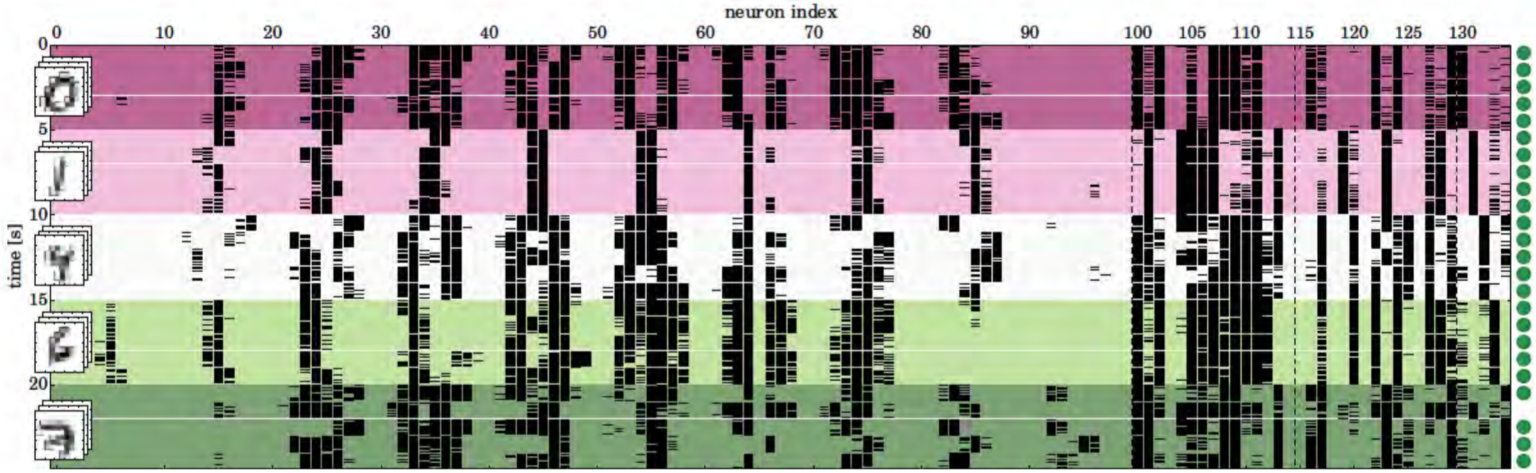


Ongoing 2nd generation prototype development
Stromg emphasis on learning capabilities

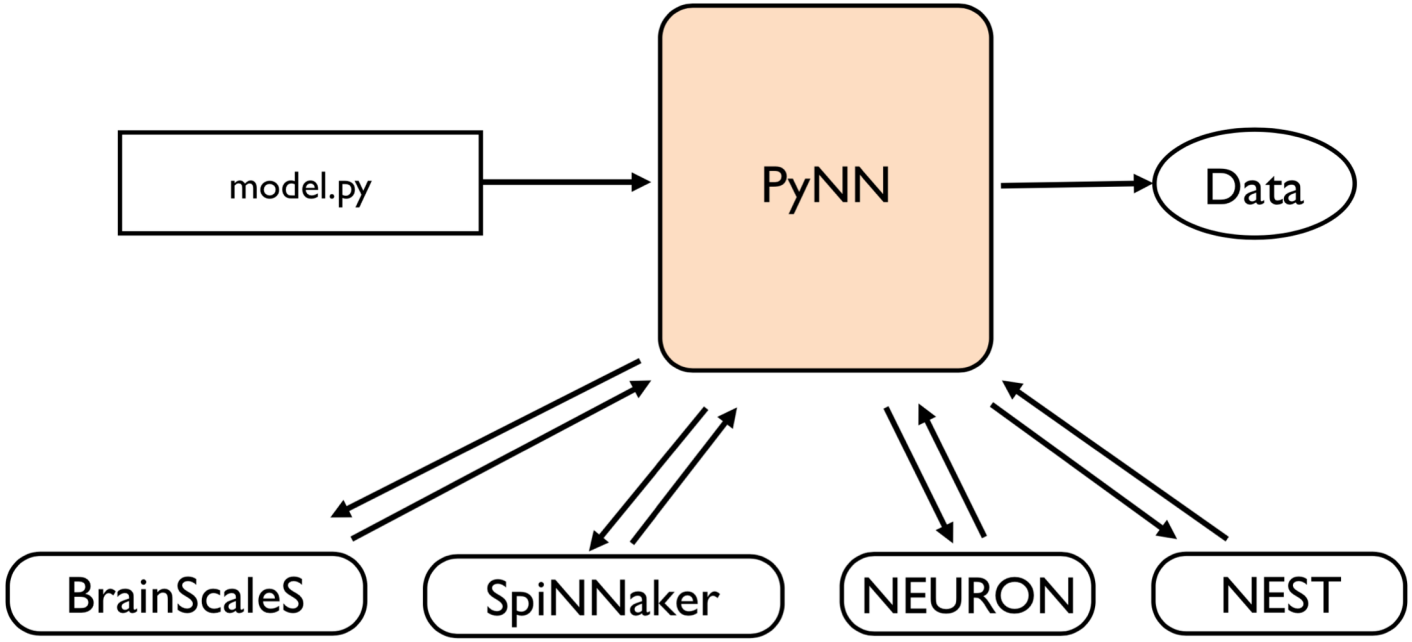
TimeScales	Nature + Real-time	Simulation	Accelerated Model
Causality Detection	10^{-4} s	0.1 s	10^{-8} s
Synaptic Plasticity	1 s	1000 s	10^{-4} s
Learning	Day	1000 Days	10 s
Development	Year	1000 Years	3000 s
12 Orders of Magnitude			
Evolution	> Millenia	> 1000 Millenia	> Months
> 15 Orders of Magnitude			







In-the-loop learning : 2 hidden layer feed forward spiking network on a physical substrate for MNIST data classification (submitted to IJCNN 2017)


model.py
PyNN
Data
BrainScaleS
SpiNNaker
NEURON
NEST

frontiers in
NEUROINFORMATICS

ORIGINAL RESEARCH ARTICLE
published: 27 January 2009
doi: 10.3389/neuro.11.011.2008

PyNN: a common interface for neuronal network simulators

Andrew P. Davison[1]*, Daniel Brüderle[2], Jochen Eppler[3,4], Jens Kremkow[5,6], Eilif Muller[7], Dejan Pecevski[8], Laurent Perrinet[6] and Pierre Yger[1]

[1] *Unité de Neurosciences Intégratives et Computationelles, CNRS, Gif sur Yvette, France*
[2] *Kirchhoff Institute for Physics, University of Heidelberg, Heidelberg, Germany*
[3] *Honda Research Institute Europe GmbH, Offenbach, Germany*
[4] *Berstein Center for Computational Neuroscience, Albert-Ludwigs-University, Freiburg, Germany*
[5] *Neurobiology and Biophysics, Institute of Biology III, Albert-Ludwigs-University, Freiburg, Germany*
[6] *Institut de Neurosciences Cognitives de la Méditerranée, CNRS, Marseille, France*
[7] *Laboratory of Computational Neuroscience, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*
[8] *Institute for Theoretical Computer Science, Graz University of Technology, Graz, Austria*

Edited by:
Rolf Kötter, Radboud University Nijmegen, The Netherlands

Reviewed by:
Graham Cummins, Montana State University, USA
Fred Howell, Textensor Limited, UK

****Correspondence:***
Andrew Davison, UNIC, Bât. 32/33, CNRS, 1 Avenue de la Terrasse, 91198 Gif sur Yvette, France.
e-mail: andrew.davison@unic.cnrs-gif.fr

Computational neuroscience has produced a diversity of software for simulations of networks of spiking neurons, with both negative and positive consequences. On the one hand, each simulator uses its own programming or configuration language, leading to considerable difficulty in porting models from one simulator to another. This impedes communication between investigators and makes it harder to reproduce and build on the work of others. On the other hand, simulation results can be cross-checked between different simulators, giving greater confidence in their correctness, and each simulator has different optimizations, so the most appropriate simulator can be chosen for a given modelling task. A common programming interface to multiple simulators would reduce or eliminate the problems of simulator diversity while retaining the benefits. PyNN is such an interface, making it possible to write a simulation script once, using the Python programming language, and run it without modification on any supported simulator (currently NEURON, NEST, PCSIM, Brian and the Heidelberg VLSI neuromorphic hardware). PyNN increases the productivity of neuronal network modelling by providing high-level abstraction, by promoting code sharing and reuse, and by providing a foundation for simulator-agnostic analysis, visualization and data-management tools. PyNN increases the reliability of modelling studies by making it much easier to check results on multiple simulators. PyNN is open-source software and is available from http://neuralensemble.org/PyNN.

Keywords: Python, interoperability, large-scale models, simulation, parallel computing, reproducibility, computational neuroscience, translation

INTERFACE FOCUS

rsfs.royalsocietypublishing.org

Research

Cite this article: Hopkins M, Pineda-García G, Bogdan PA, Furber SB. 2018 Spiking neural networks for computer vision. *Interface Focus* **8**: 20180007.
http://dx.doi.org/10.1098/rsfs.2018.0007

Accepted: 2 May 2018

One contribution of 12 to a theme issue 'Understanding images in biological and computer vision'.

Subject Areas:
computational biology

Spiking neural networks for computer vision

Michael Hopkins, Garibaldi Pineda-García, Petruţ A. Bogdan and Steve B. Furber

School of Computer Science, The University of Manchester, Oxford Road, Manchester M13 9PL, UK

GP-G, 0000-0002-6550-6016; PAB, 0000-0001-5535-7865; SBF, 0000-0002-6524-3367

State-of-the-art computer vision systems use frame-based cameras that sample the visual scene as a series of high-resolution images. These are then processed using convolutional neural networks using neurons with continuous outputs. Biological vision systems use a quite different approach, where the eyes (cameras) sample the visual scene continuously, often with a non-uniform resolution, and generate neural spike events in response to changes in the scene. The resulting spatio-temporal patterns of events are then processed through networks of spiking neurons. Such event-based processing offers advantages in terms of focusing constrained resources on the most salient features of the perceived scene, and those advantages should also accrue to engineered vision systems based upon similar principles. Event-based vision sensors, and event-based processing exemplified by the SpiNNaker (Spiking Neural Network Architecture) machine, can be used to model the biological vision pathway at various levels of detail. Here we use this approach to explore structural synaptic plasticity as a possible mechanism whereby biological vision systems may learn the statistics of their inputs without supervision, pointing the way to engineered vision systems with similar online learning capabilities.

```
input_list = KernelConnectionList(input_kernel, ...)
lateral_list = KernelConnectionList(lateral_kernel, ...)

evs = EVSObject(...)
bipolar_cells = pynn.Population(...)
inter_cells = pynn.Population(...)
ganglion_cells = pynn.Population(...)

pynn.Projection(evs, bipolar_cells, pynn.FromListConnector(input_list),
      receptor_type='excitatory')
pynn.Projection(bipolar_cells, inter_cells,
      pynn.OneToOneConnector(), pynn.StaticSynapse(...),
      receptor_type='excitatory')
pynn.Projection(bipolar_cells, ganglion_cells,
      pynn.OneToOneConnector(), pynn.StaticSynapse(...),
      receptor_type='excitatory')
pynn.Projection(inter_cells, ganglion_cells,
      pynn.FromListConnector(lateral_list),
      receptor_type='inhibitory')
```

Figure 8. Multi-scale image representation PyNN code. (Online version in colour.)

Table 1. The parameters used in the simulations presented throughout this section.

wiring	inputs	membrane	STDP[a]
$N_{layer} = 28 \times 28$	$f_{mean} = 5$ Hz	$v_{rest} = -70$ mV	$A_+ = 0.1$
$S_{max} = 96$	$t_{stim} = 200$ ms	$e_{ext} = 0$ mV	$B = 1.2$
$\sigma_{form\text{-}ff} = 2.5$	—	$v_{thr} = -54$ mV	$\tau_+ = 20$ ms
$\sigma_{form\text{-}lat} = 2$	—	$g_{max} = 0.1$ nS	$\tau_- = 20$ ms
$p_{form\text{-}ff} = 0.16$	—	$\tau_m = 20$ ms	—
$p_{form\text{-}lat} = 1$	—	$\tau_{ex} = 5$ ms	—
$p_{elim\text{-}dep} = 0.0245$	—	—	—
$p_{elim\text{-}pot} = 1.36 \times 10^{-4}$	—	—	—
$f_{rew} = 10$ kHz	—	—	—

[a]The STDP parameters are only used when synaptic plasticity is used in conjunction with the rewiring.

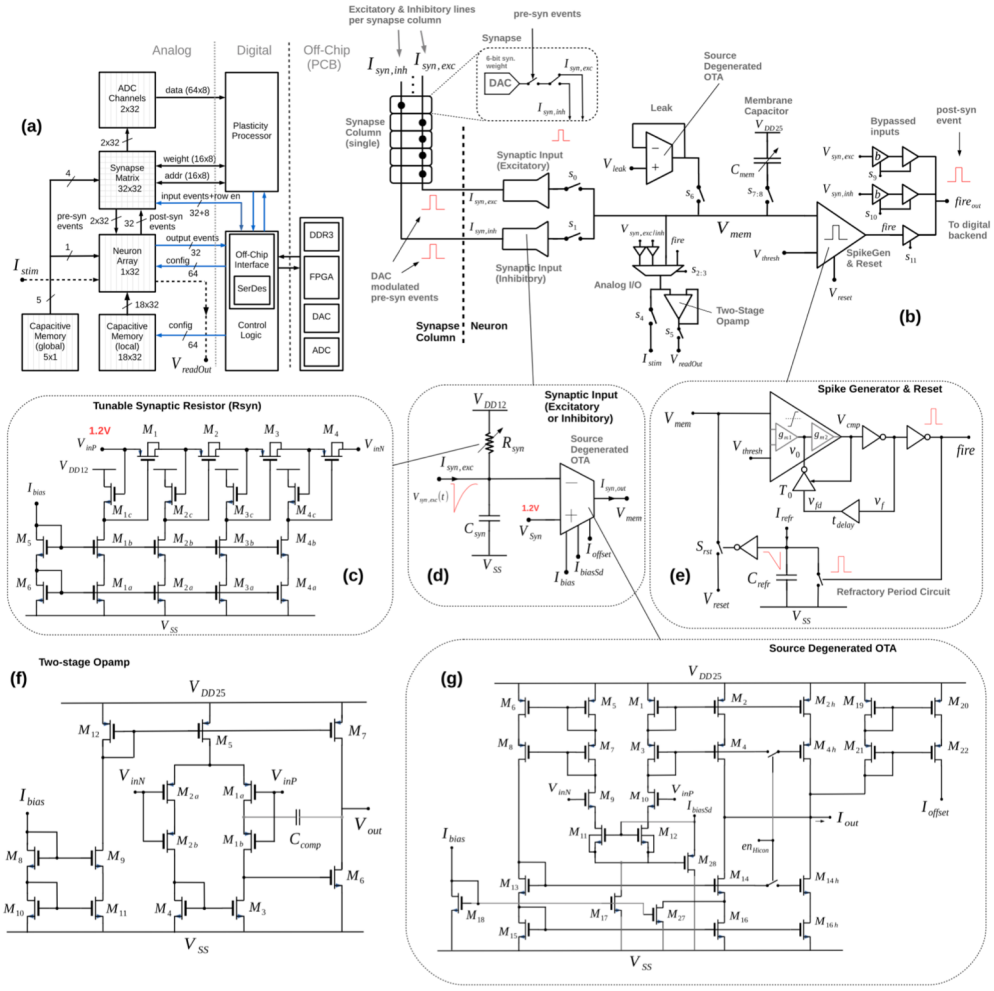


Fig. 1: (a) Architecture of the HICANN-DLS prototype chip and the measurement system. (b) The full circuit schematic of a single integrated neuron. (c) The schematic of the resistor used inside the synaptic input. (d) The architecture of the excitatory synaptic input (swapped terminals for inhibitory synaptic input). (e) The schematic of the spike pulse generator and reset circuit (implements refractory period duration). (f) The two-stage opamp schematic used inside the read-out buffer. (g) The schematic of the source-degenerated Operational Transconductance Amplifier (OTA) (used inside synaptic input and leak).

Neuromorphic Platform Specification — public version

18 September 2018 (git e02b5c7 — public)

Human Brain Project

Figure .1: Rendered View of the NM-PM1 system (for explanations see page 51)

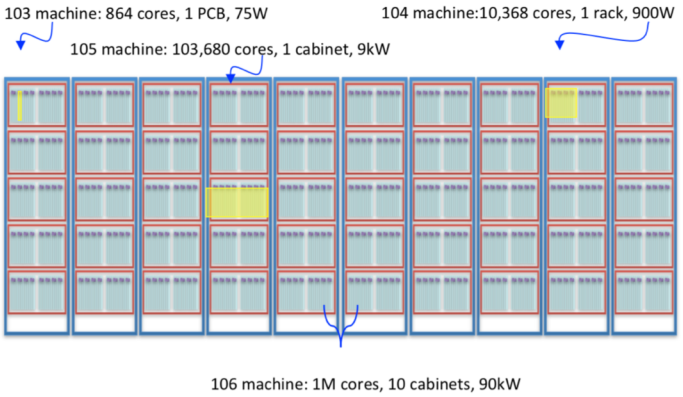


Figure .2: Concept view of the NM-MC1 system