Spike-based learning and sensory processing in Silicon

Giacomo Indiveri

Institute of Neuroinformatics (INI)
UZH | ETH Zurich

Telluride, July 2007

Outline

VLSI: why go to all this trouble?

- Applications (Tobi, Ralph, Tony, Andre, Paul, . . .)



- Grand challenges (Kwabena, Paul, . . .)
- Basic research (Elisabetta, Shih-Chii, . . .)

VLSI: why go to all this trouble?

- Applications (Tobi, Ralph, Tony, Andre, Paul, . . .)



- Grand challenges (Kwabena, Paul, . . .)

- Basic research (Elisabetta, Shih-Chii, . . .)

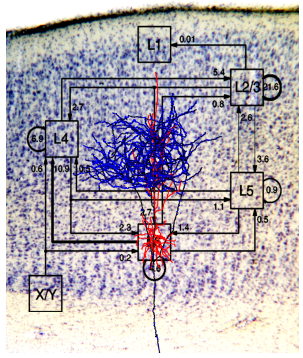
VLSI: why go to all this trouble?

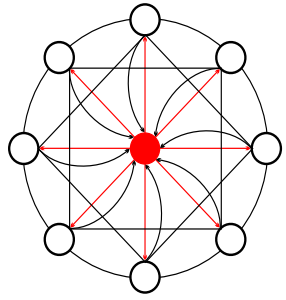
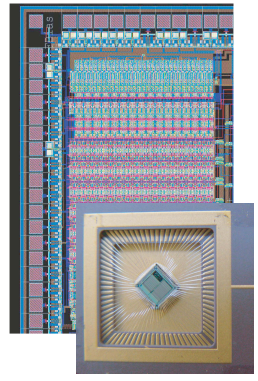
- Applications (Tobi, Ralph, Tony, Andre, Paul, . . .)



- Grand challenges (Kwabena, Paul, . . .)

- Basic research (Elisabetta, Shih-Chii, . . .)



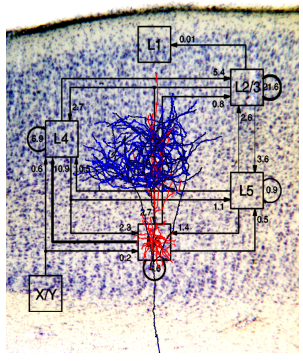
VLSI: why go to all this trouble?

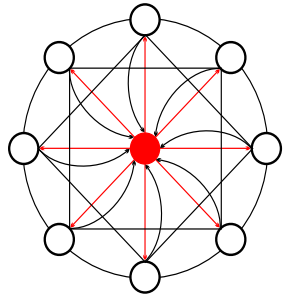
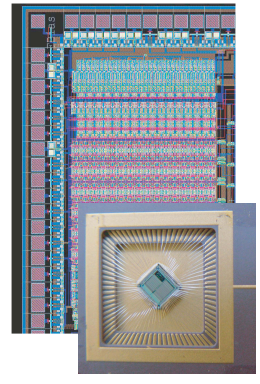
- Applications (Tobi, Ralph, Tony, Andre, Paul, . . .)



- Grand challenges (Kwabena, Paul, . . .)

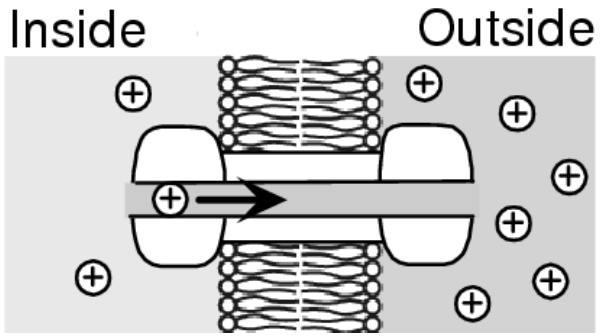
- Basic research (Elisabetta, Shih-Chii, . . .)

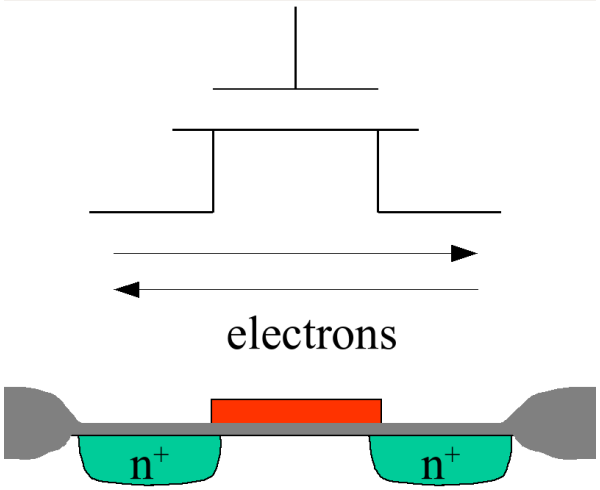


Neuromorphic VLSI systems

An attractive alternative computing paradigm

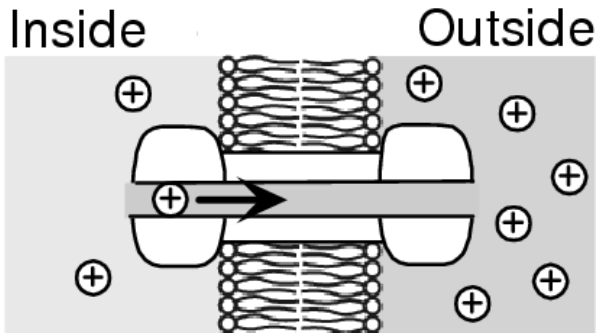


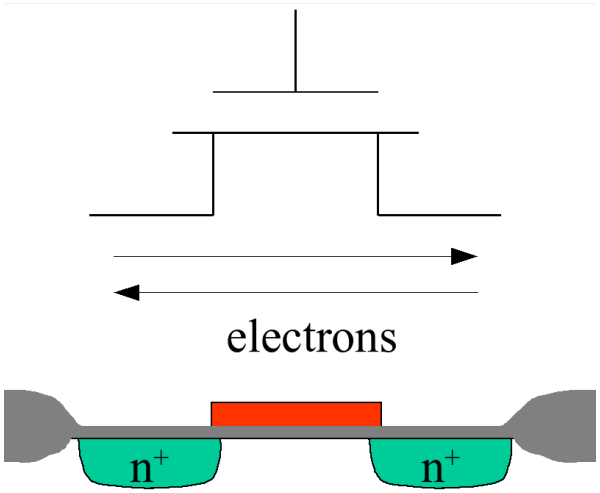


Exploit the physics of silicon to reproduce the *bio*-physics of neural systems.

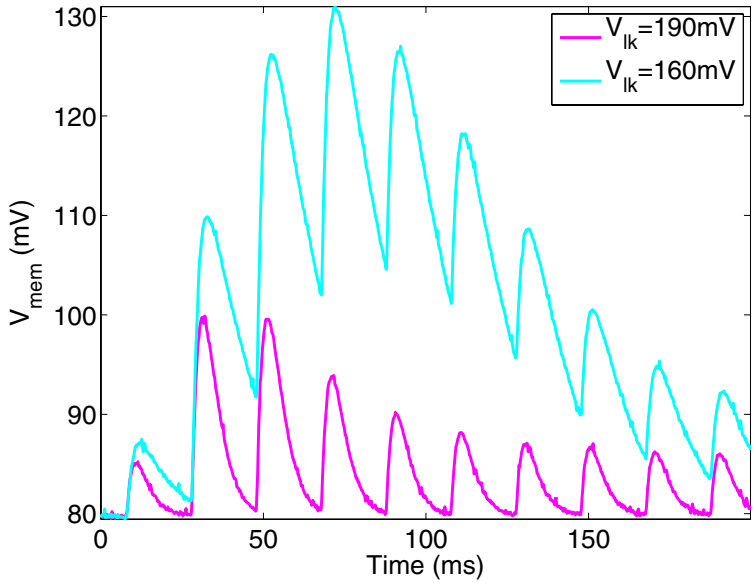
Neuromorphic VLSI systems

An attractive alternative computing paradigm



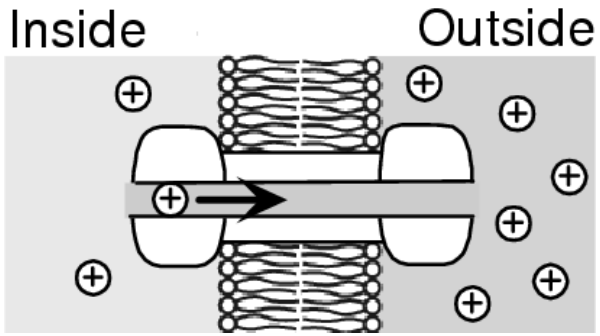


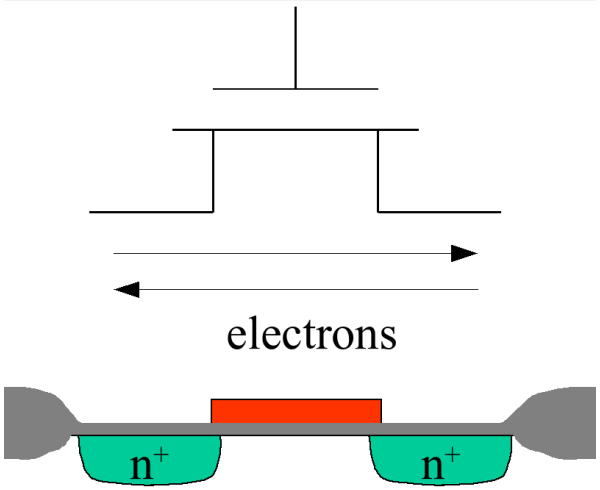
Exploit the physics of silicon to reproduce the *bio*-physics of neural systems.



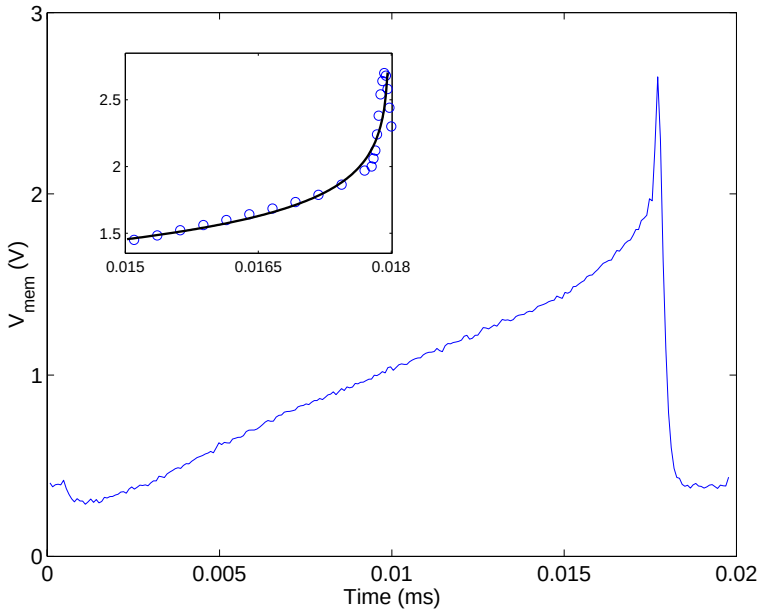
Neuromorphic VLSI systems

An attractive alternative computing paradigm



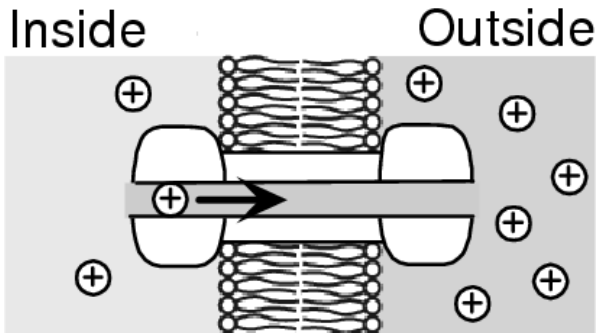


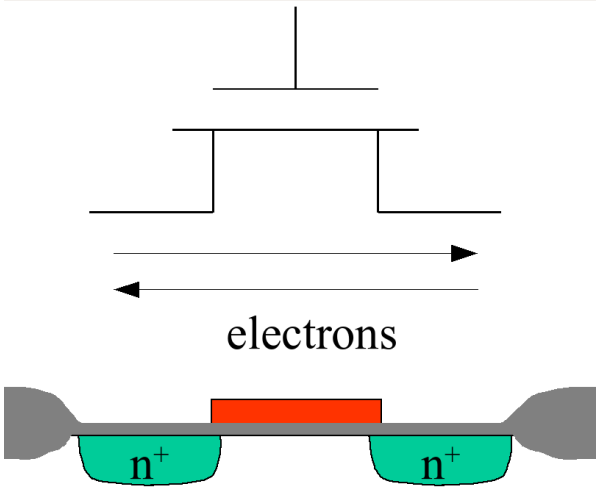
Exploit the physics of silicon to reproduce the *bio*-physics of neural systems.



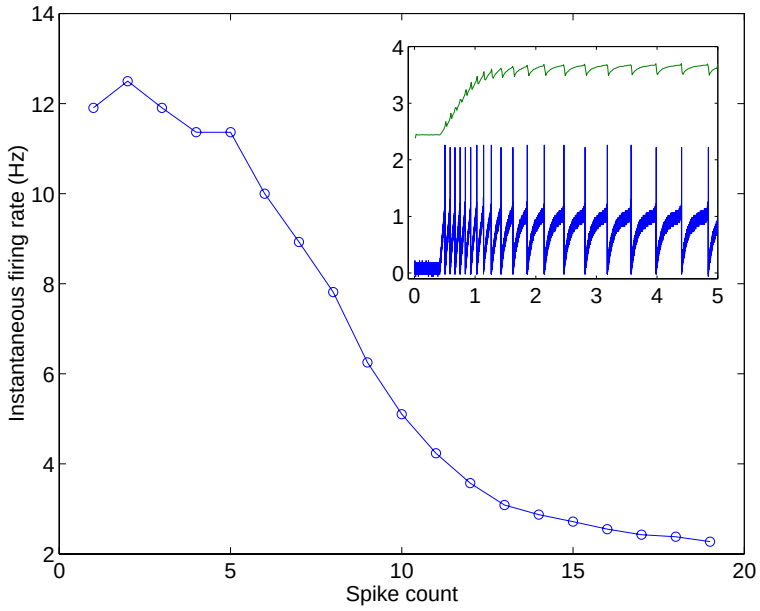
Neuromorphic VLSI systems

An attractive alternative computing paradigm



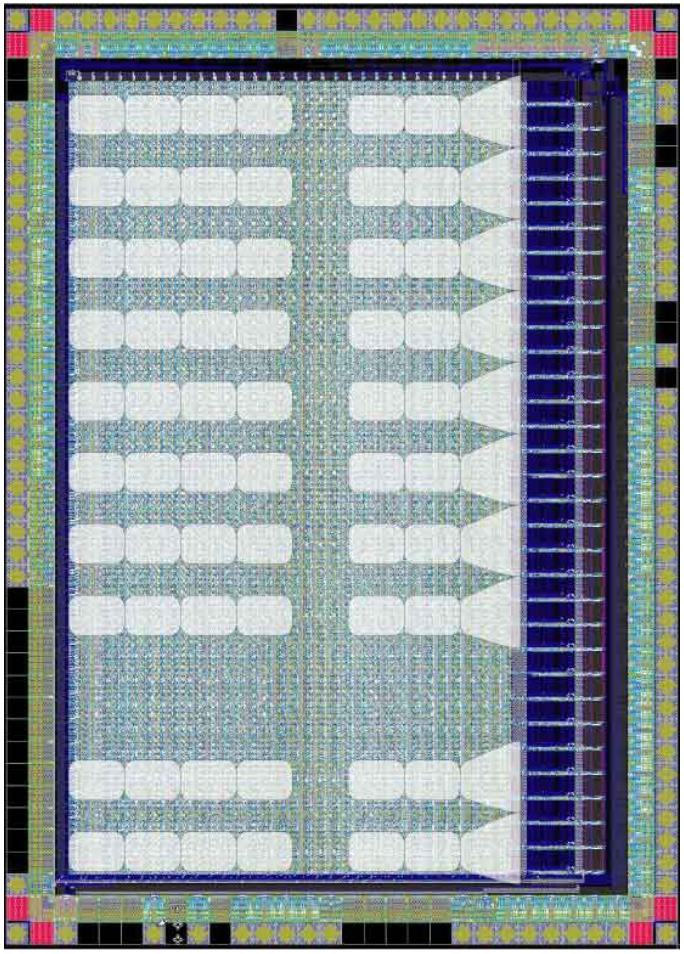


Exploit the physics of silicon to reproduce the *bio*-physics of neural systems.



Neuromorphic VLSI chips

Arrays of silicon synapses and integrate and fire neurons

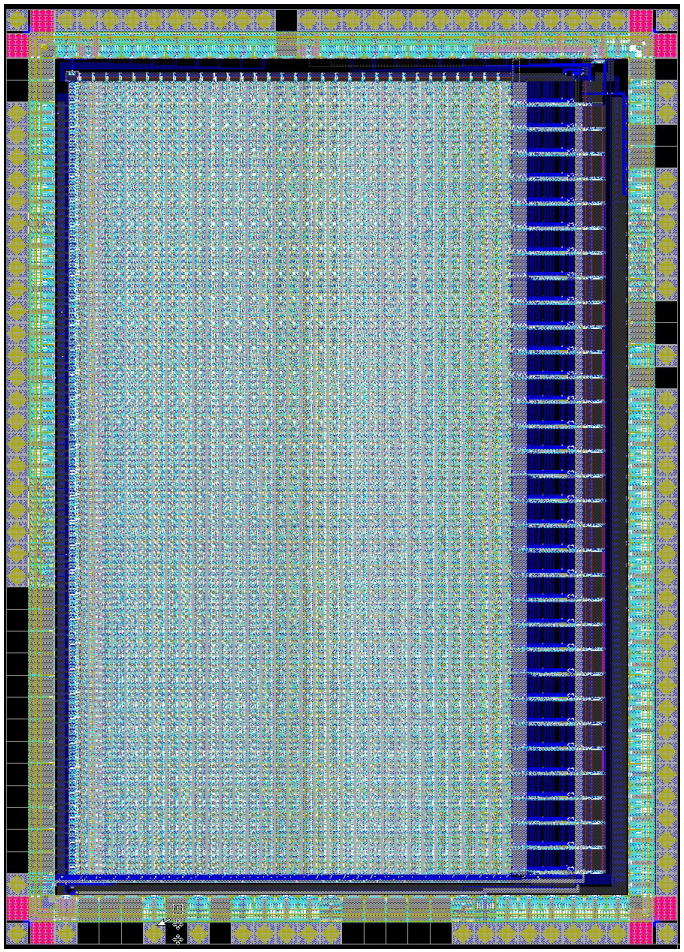


Neuromorphic chips comprising silicon neurons and synapses.

- Time constants are biologically plausible
- Synaptic currents are integrated in parallel
- Synapses are the site of memory *and* computation
- Neurons generate and transmit “spikes” in an asynchronous (non-clocked) fashion.

Neuromorphic VLSI chips

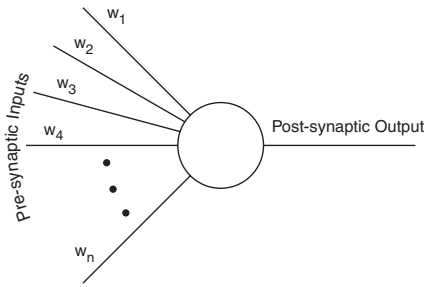
Arrays of silicon synapses and integrate and fire neurons

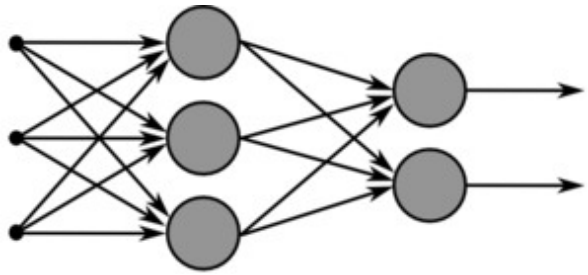


- Standard CMOS Technology
- Process independent
- Massively parallel
- Mismatch insensitive
- Fault tolerant
- Compact
- Low-power

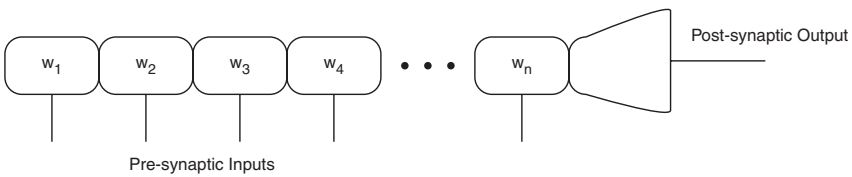
Neurons and synapses

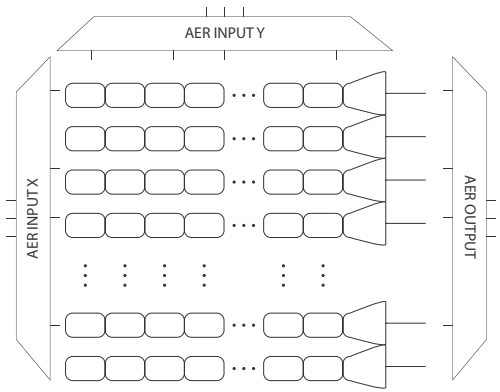
Classical neural networks





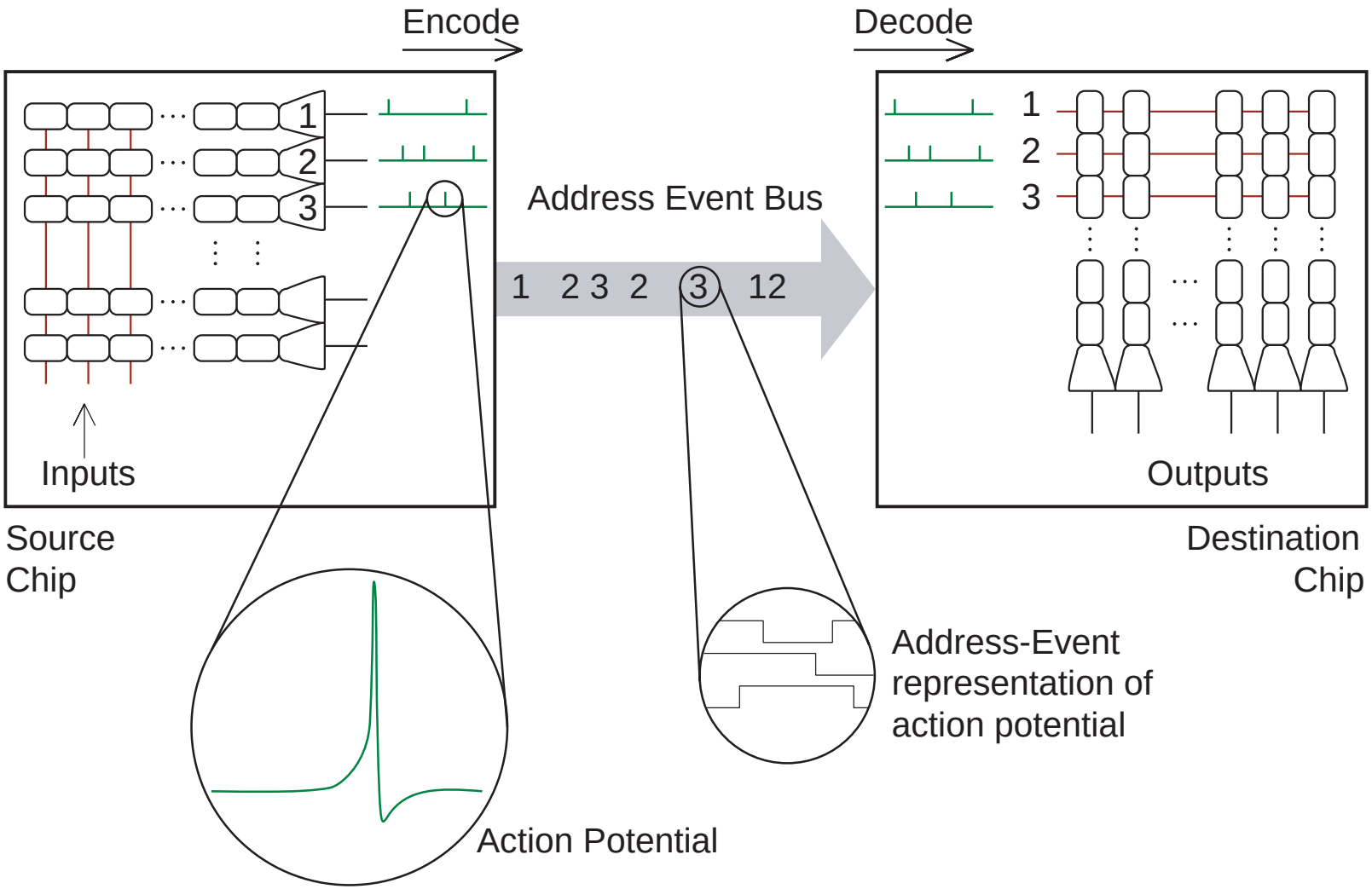
AER neural networks



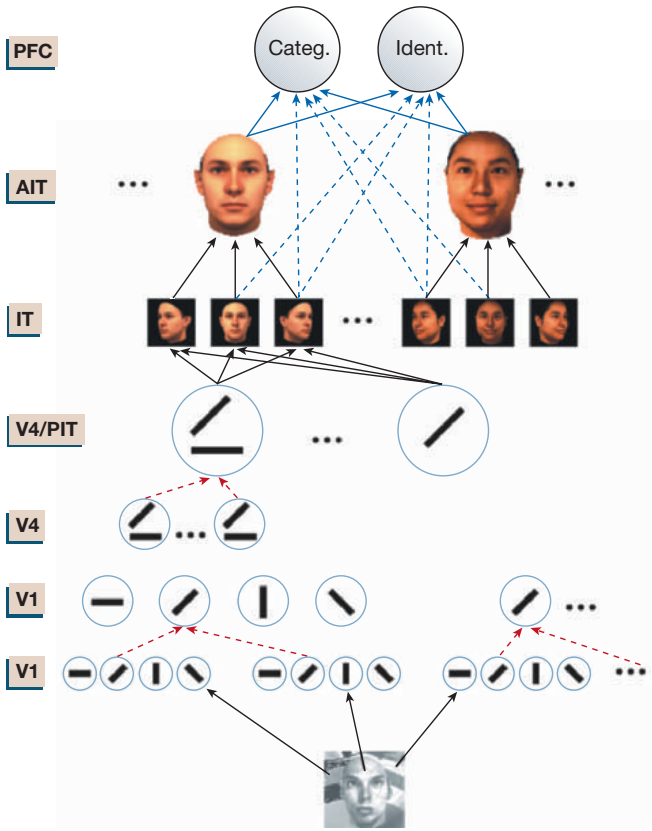


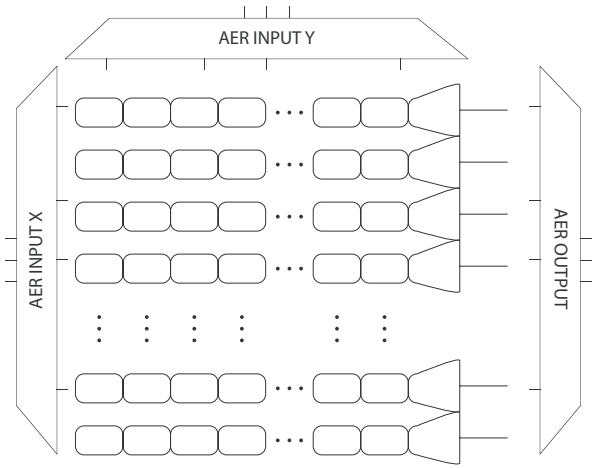
Address Event Representation

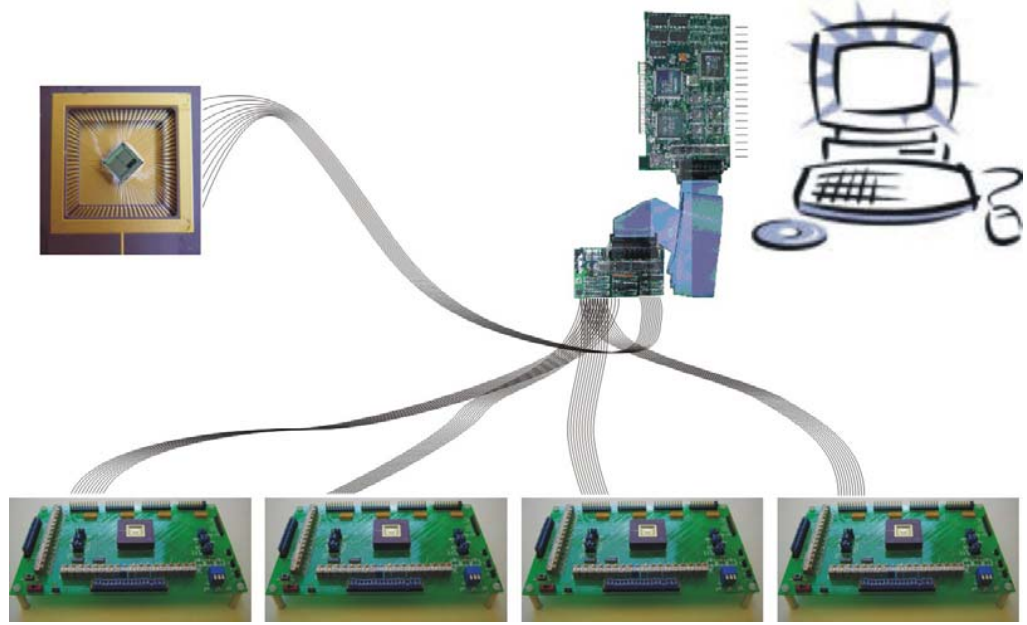
Best of both (digital & analog) worlds



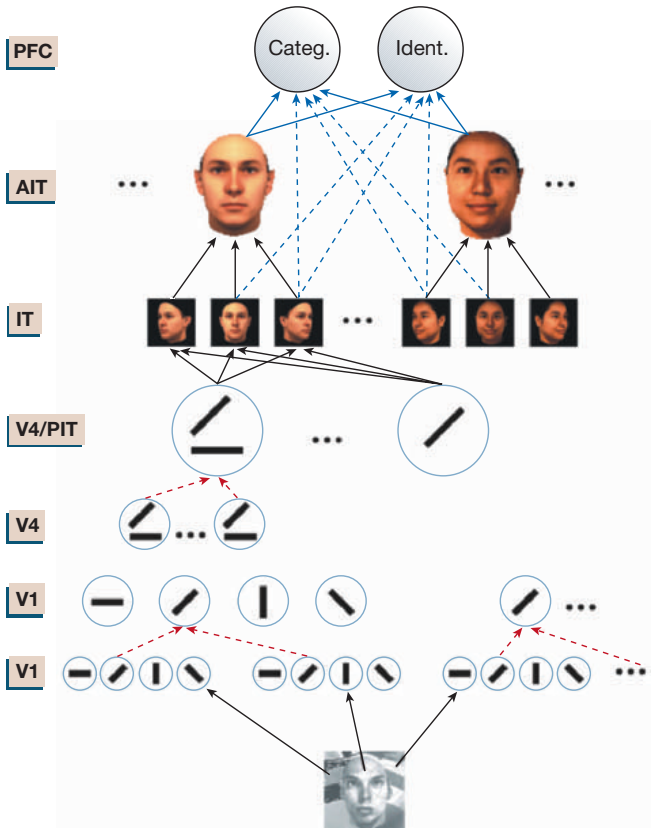
Hierarchical or multi-layer networks

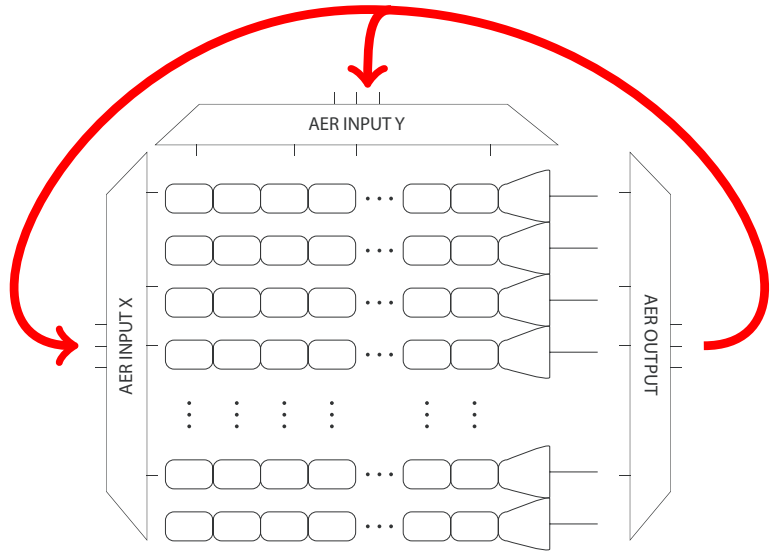


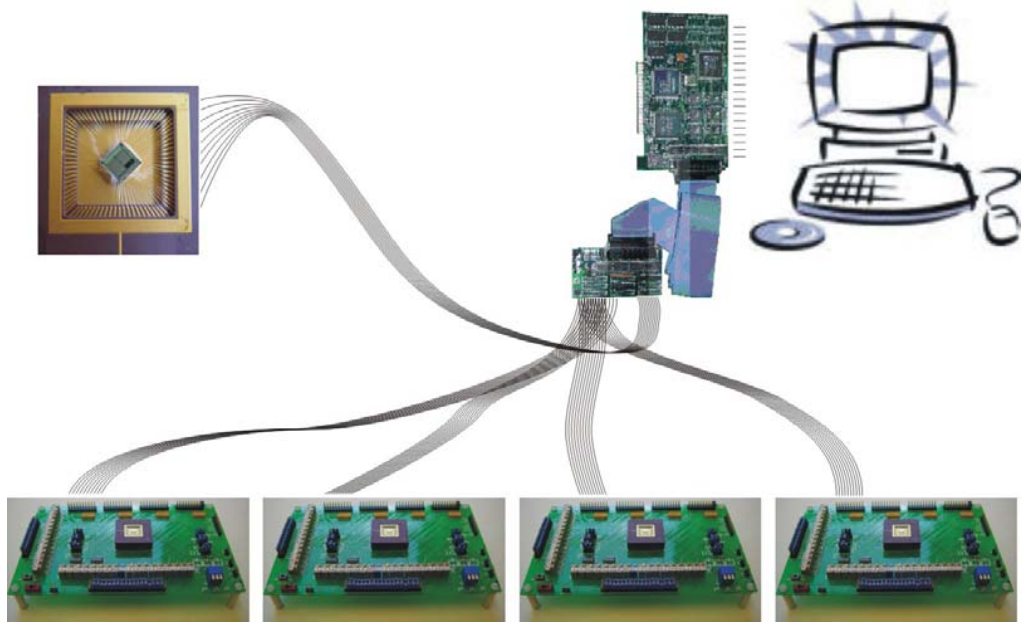




Hierarchical or multi-layer networks







Outline

Hardware implementations of spiking neurons

Hardware implementations of this model date back to the period when McCulloch and Pitts proposed the first artificial neuron model (1943)

Hardware implementations of spiking neurons

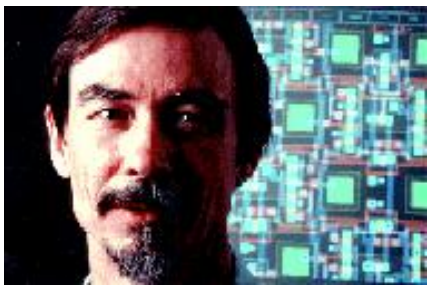
Hardware implementations of this model date back to the period when McCulloch and Pitts proposed the first artificial neuron model (1943)

VLSI implementations of *spiking* neurons are relatively new.

Hardware implementations of spiking neurons

Hardware implementations of this model date back to the period when McCulloch and Pitts proposed the first artificial neuron model (1943)

VLSI implementations of *spiking* neurons are relatively new.

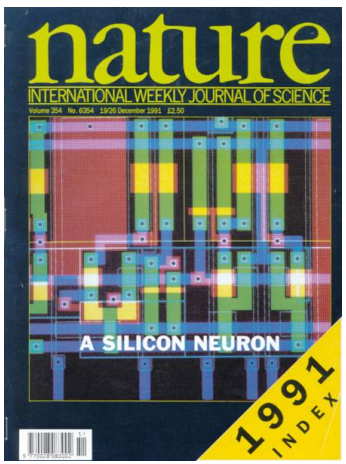


One of the most influential circuits that implements an *integrate and fire* model of a neuron was proposed by Carver Mead in the late 1980s.

Hardware implementations of spiking neurons

Hardware implementations of this model date back to the period when McCulloch and Pitts proposed the first artificial neuron model (1943)

VLSI implementations of *spiking* neurons are relatively new.



In 1991 Misha Mahowald and Rodney Douglas proposed a conductance-based silicon neuron and showed that it had properties remarkably similar to those of real cortical neurons.

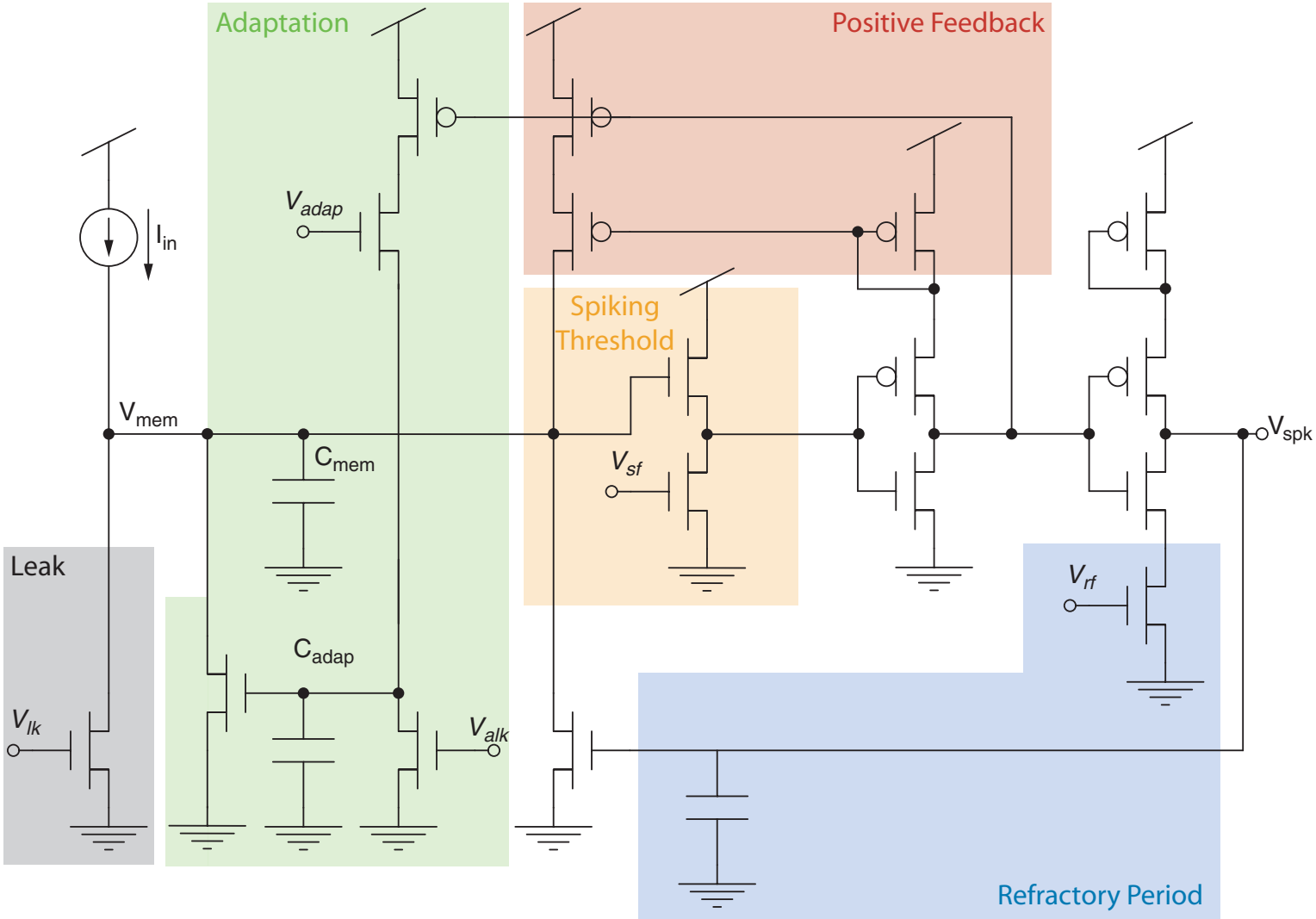
Hardware implementations of spiking neurons

Hardware implementations of this model date back to the period when McCulloch and Pitts proposed the first artificial neuron model (1943)

VLSI implementations of *spiking* neurons are relatively new.

The whole community kept on developing various flavors of spiking VLSI neurons.

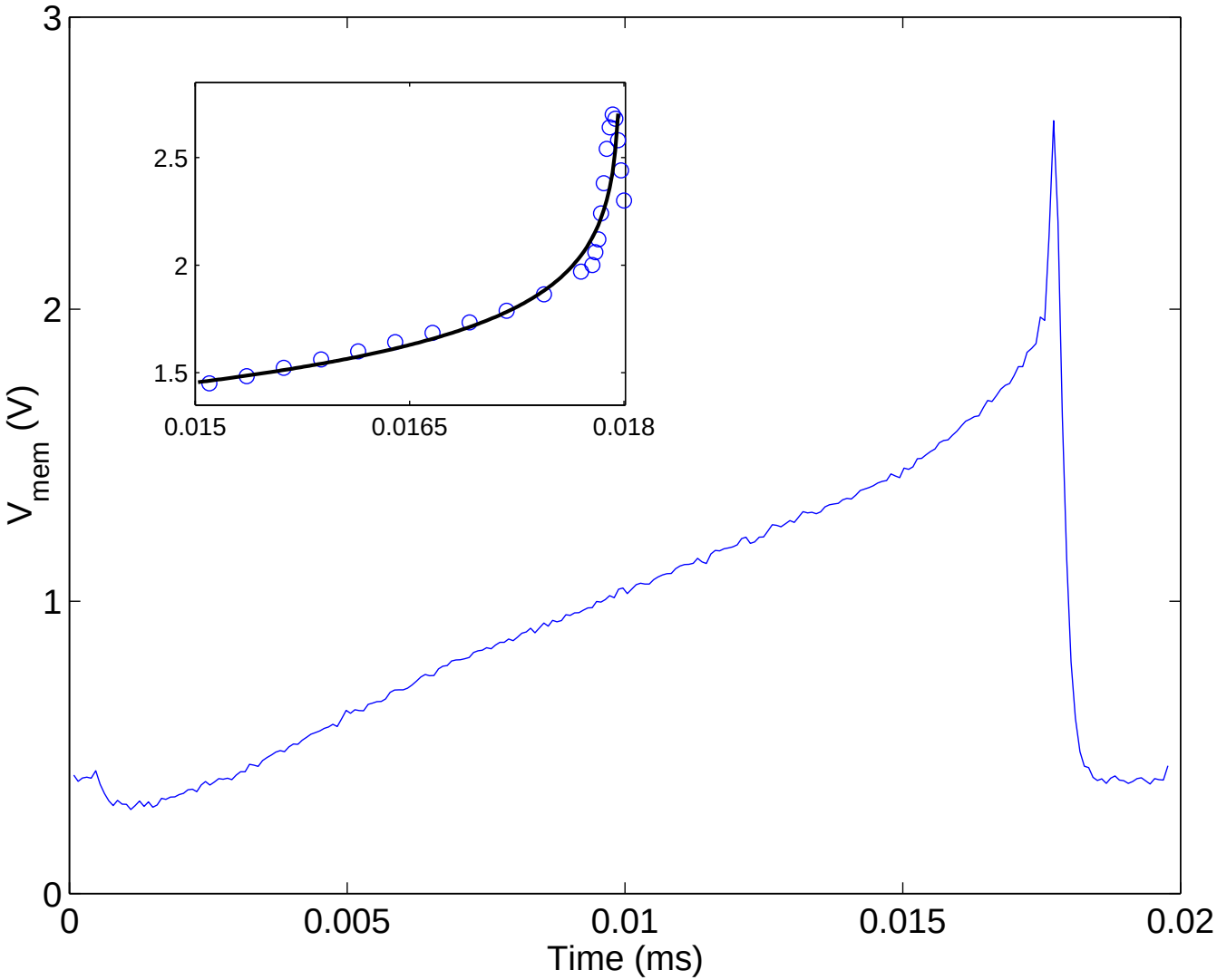
The low-power I&F neuron



Indiveri et al. IEEE Transactions on Neural Networks, Jan 2006

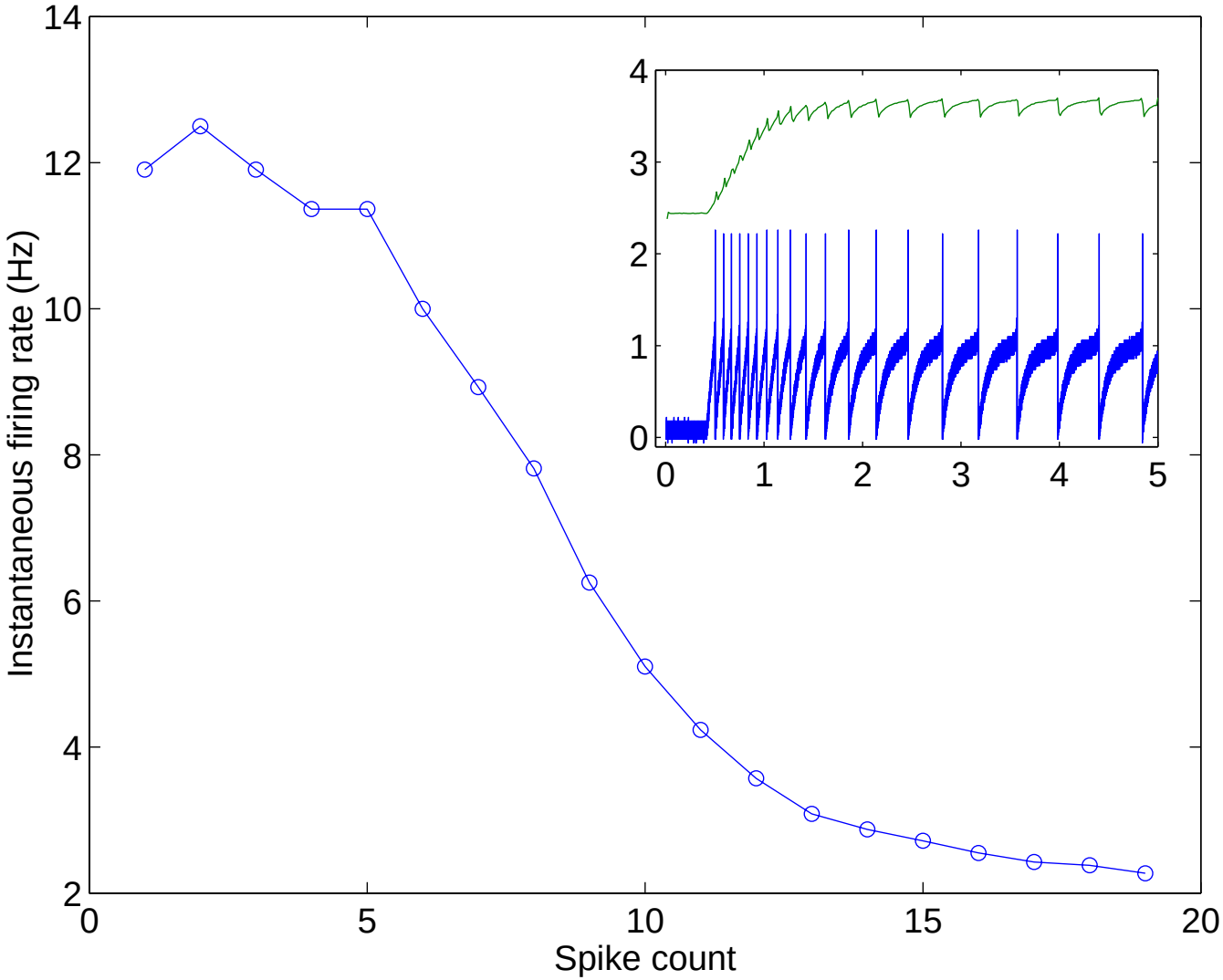
The low-power I&F neuron

Positive Feedback (*low power*)



The low-power I&F neuron

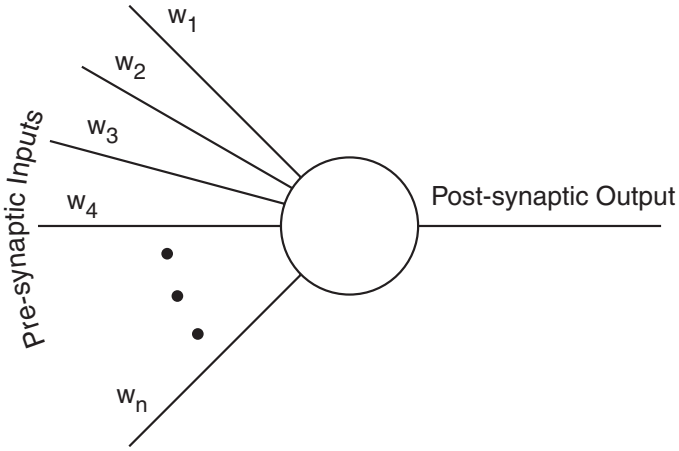
Adaptation (*intrinsic plasticity*)



Synapses

Real synapses

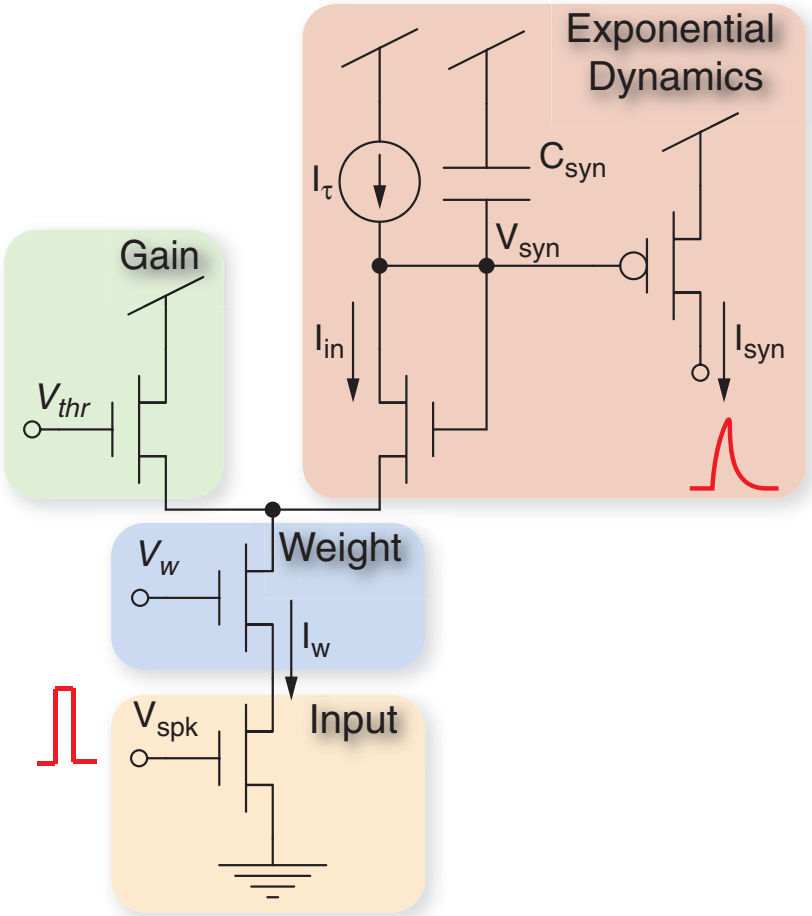
Artificial synapses



Synapses are often modeled as instantaneous multipliers.

Science and Engineering Visualization Challenge
`http://www.sciencemag.org/sciext/vis2005/show/slide1.dtl`
2005 winner, Graham Johnson, *Medical Media*, Boulder, Colorado.

The diff-pair integrator (DPI) circuit



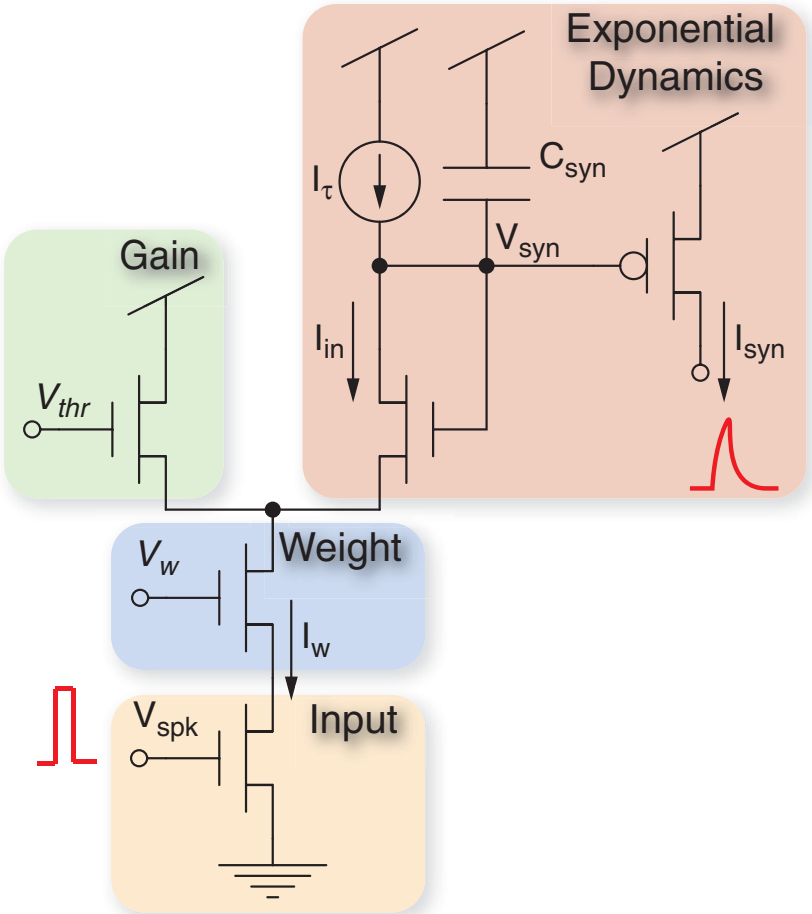
$$I_{syn}(t) = I_0 e^{-\frac{\kappa}{U_T}(V_{syn}(t) - V_{dd})}$$

$$I_{thr} = I_0 e^{-\frac{\kappa(V_{thr} - V_{dd})}{U_T}}$$

$$C_{syn}\frac{d}{dt}V_{syn} = -(I_{in} - I_\tau)$$

Bartolozzi, Indiveri *Neural Computation*, 2007

The diff-pair integrator (DPI) circuit



$$I_{syn}(t) = I_0 e^{-\frac{\kappa}{U_T}(V_{syn}(t) - V_{dd})}$$

$$I_{thr} = I_0 e^{-\frac{\kappa(V_{thr} - V_{dd})}{U_T}}$$

$$C_{syn}\frac{d}{dt}V_{syn} = -(I_{in} - I_\tau)$$

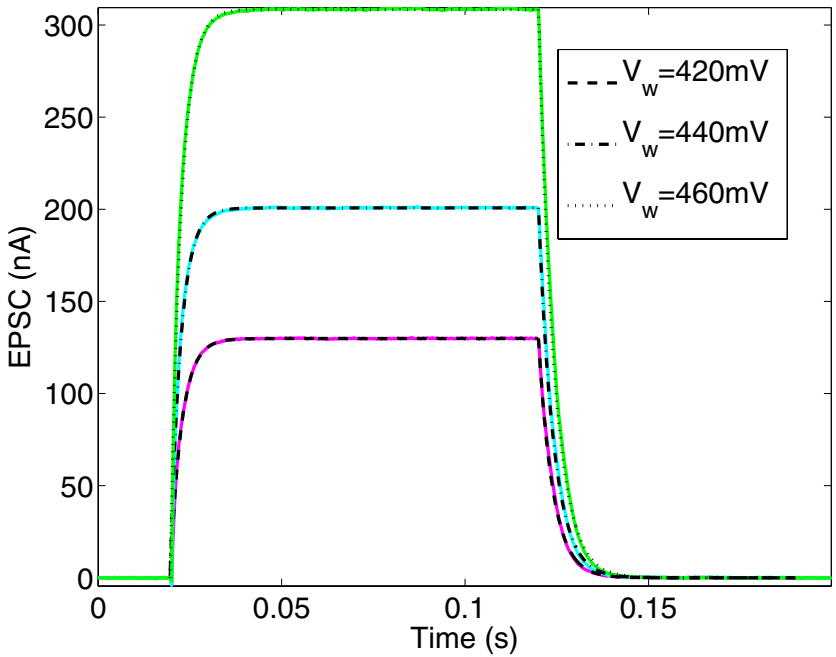
$$\tau\frac{d}{dt}I_{syn} + I_{syn} = \frac{I_{thr}I_w}{I_\tau}$$

The EPSC amplitude can be controlled with *three* parameters.

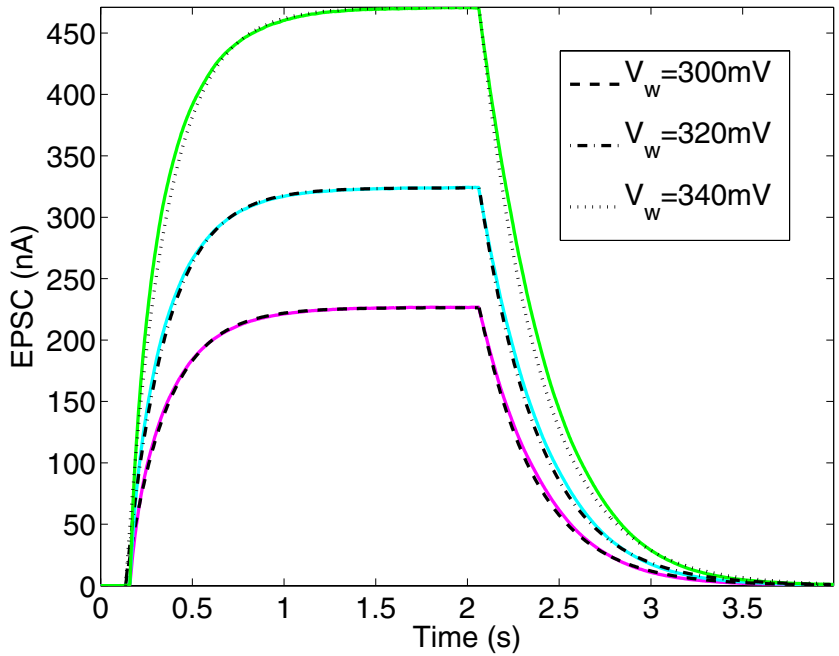
Bartolozzi, Indiveri *Neural Computation*, 2007

Diff-pair integrator step-response

AMPA time constants (large I_τ)

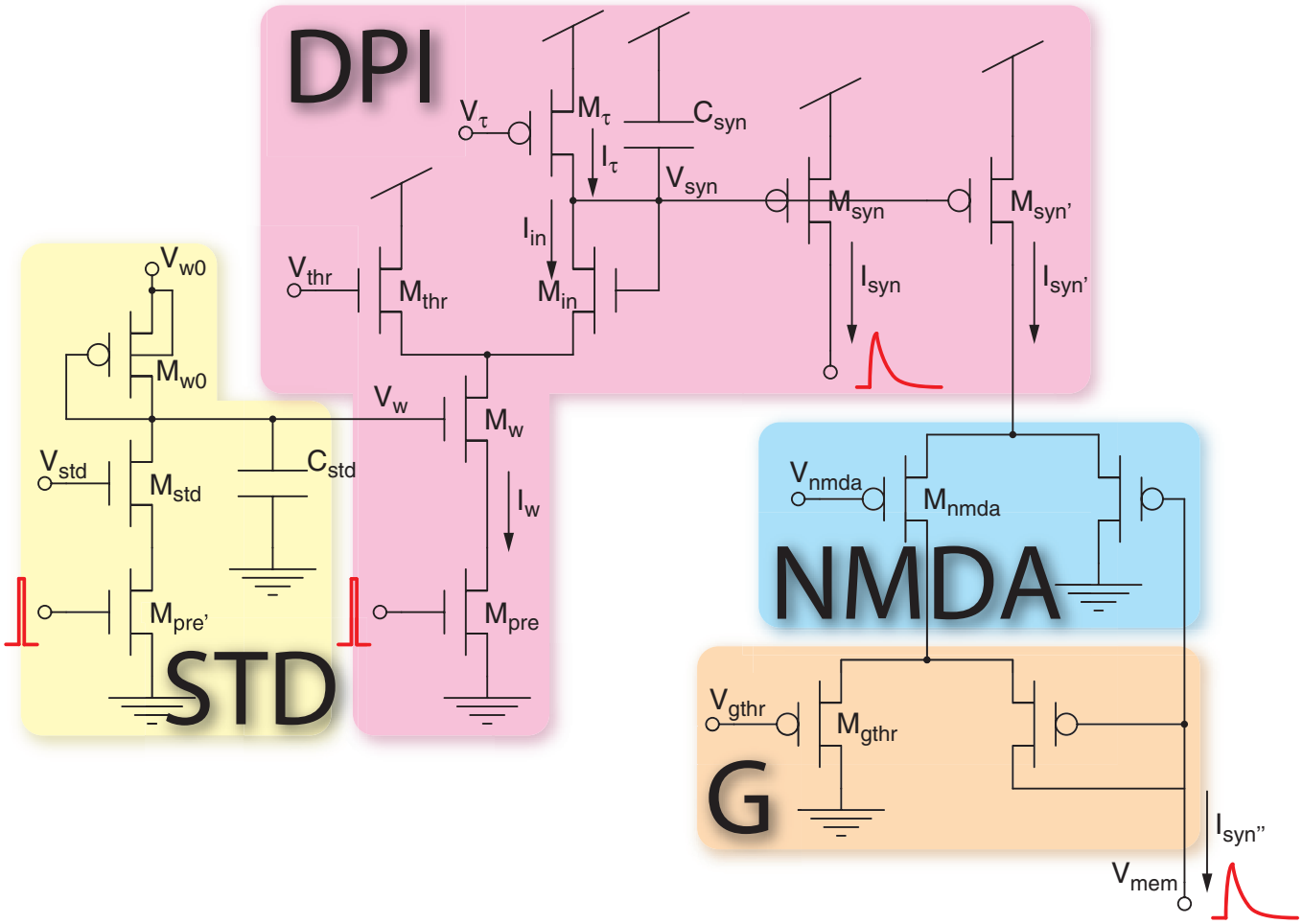


NMDA time constants (small I_τ)

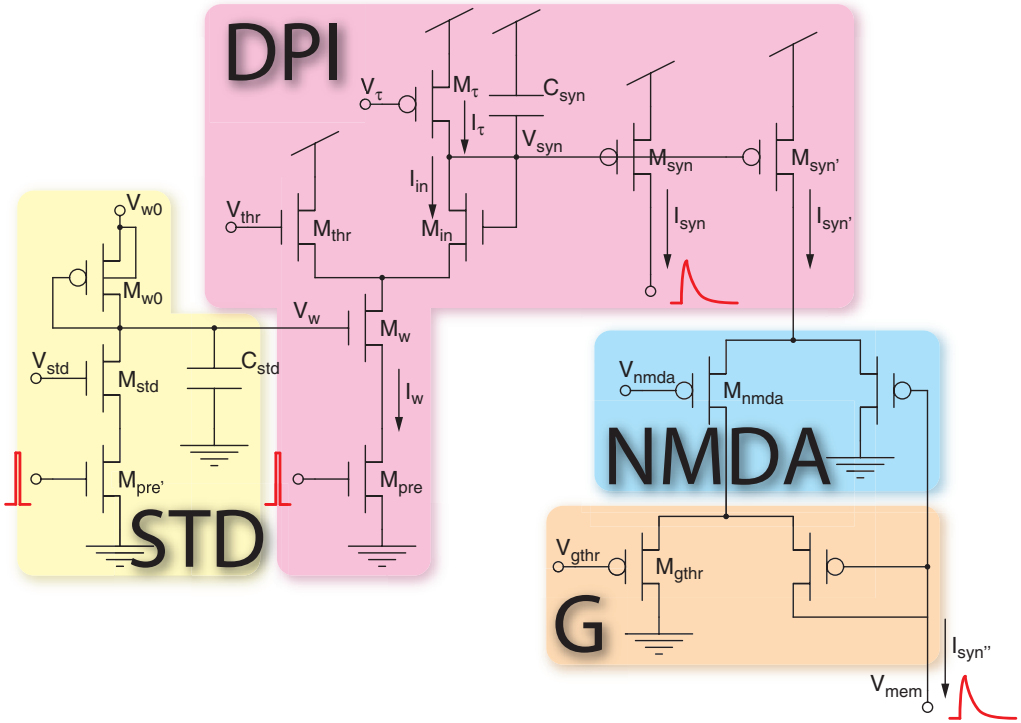


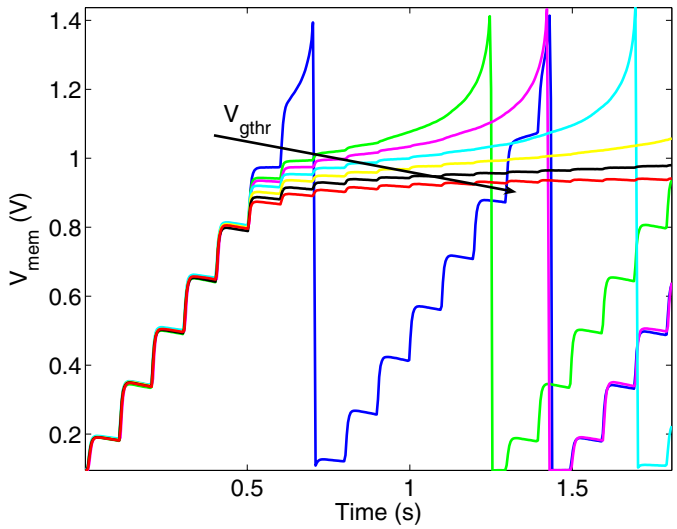
“Elaborate” dynamic behavior

NMDA, conductance-based and short-term depression



Conductance-based synapse



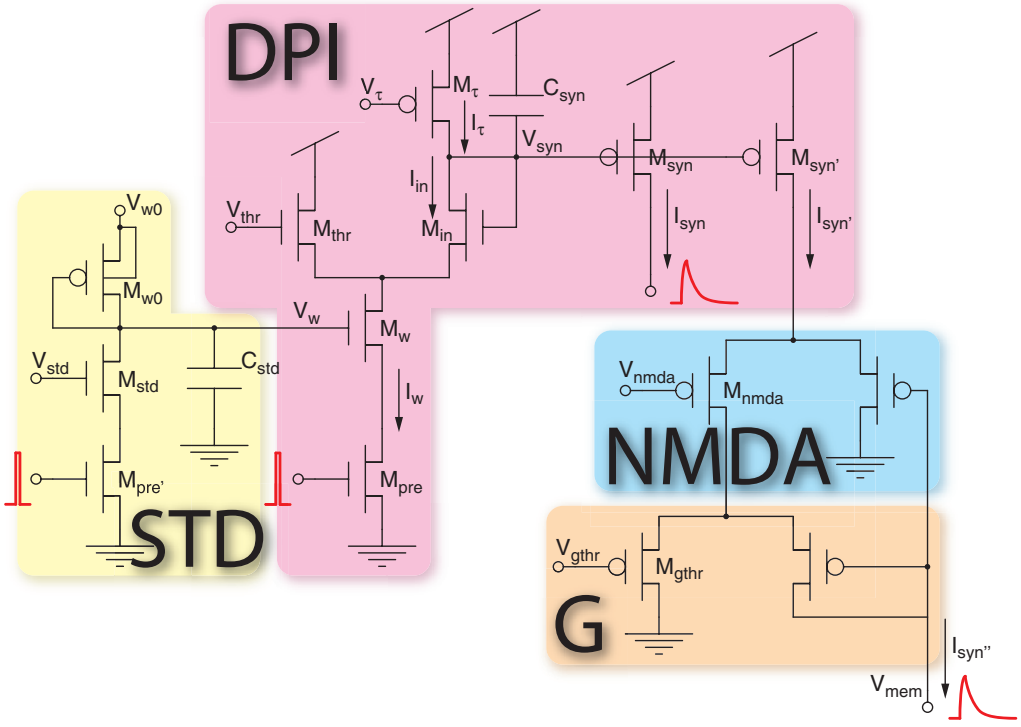


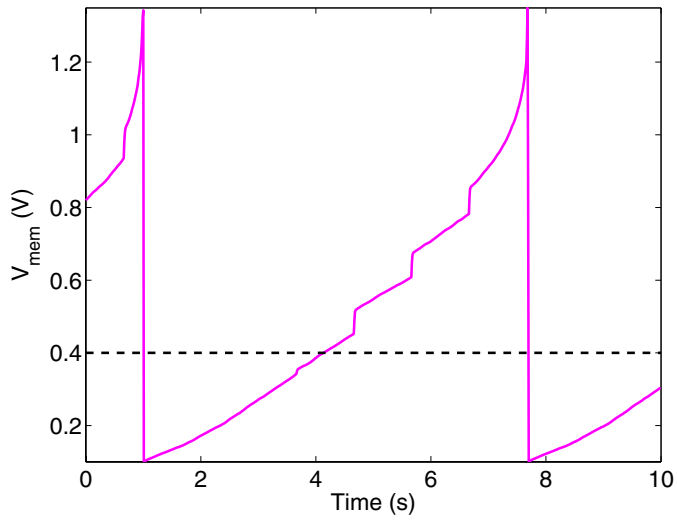
$$I_{syn} = g_{syn}(V_{mem} - E_{ion})$$

$$I_{syn''} = I_{syn'} \frac{1}{1 + e^{\frac{\kappa}{U_T}(V_{mem} - V_{gthr})}} \approx \frac{I_{syn'}}{2} + g_{syn}(V_{mem} - V_{gthr})$$

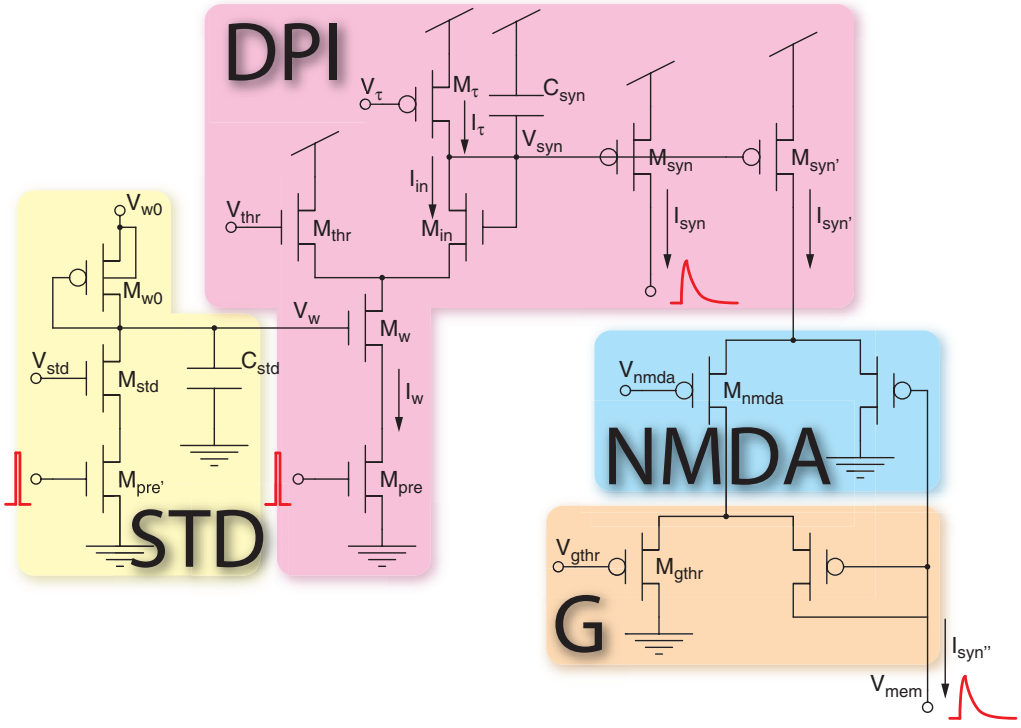
where $g_{syn} = I_{syn'} \frac{\kappa}{4U_T}$.

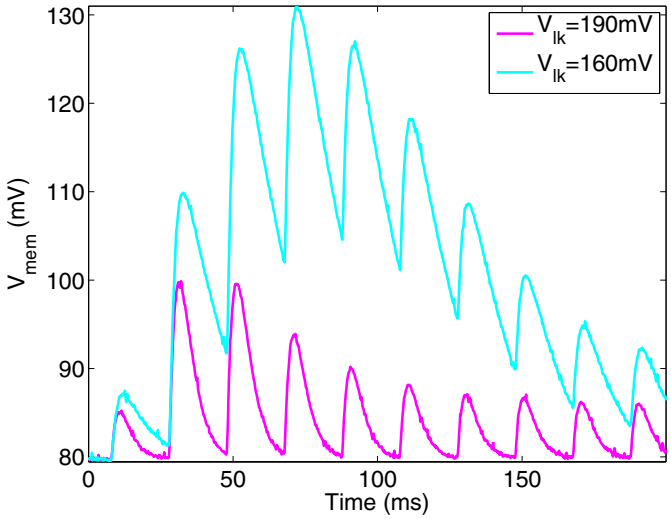
Voltage-gated (NMDA) synapse





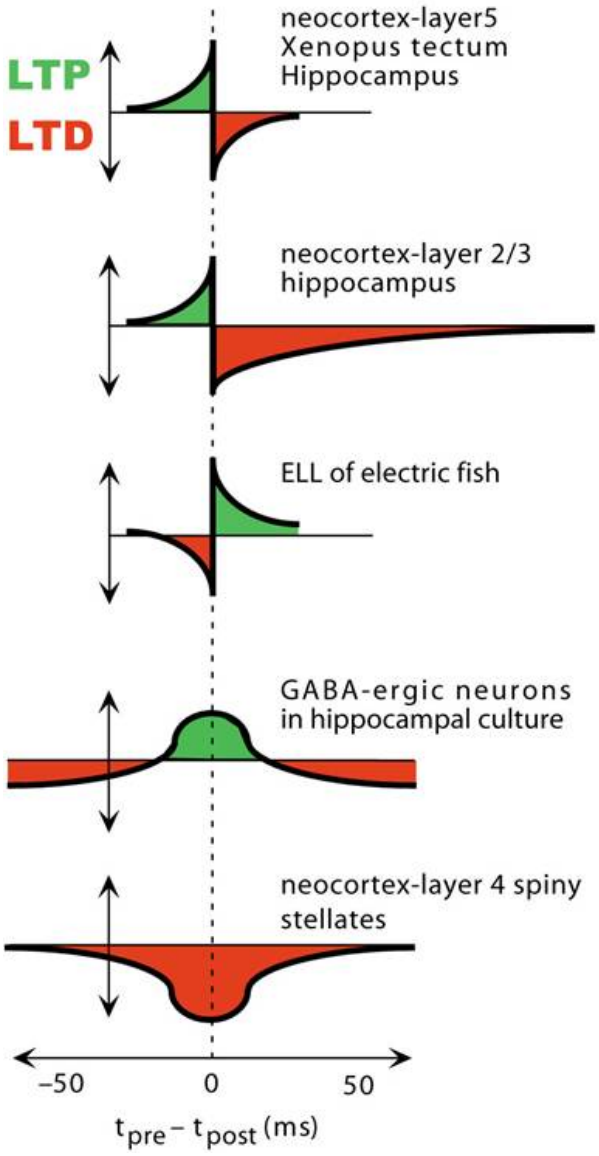
Short-term depressing synapse





Outline

Spike-timing dependent plasticity



STDP mechanism

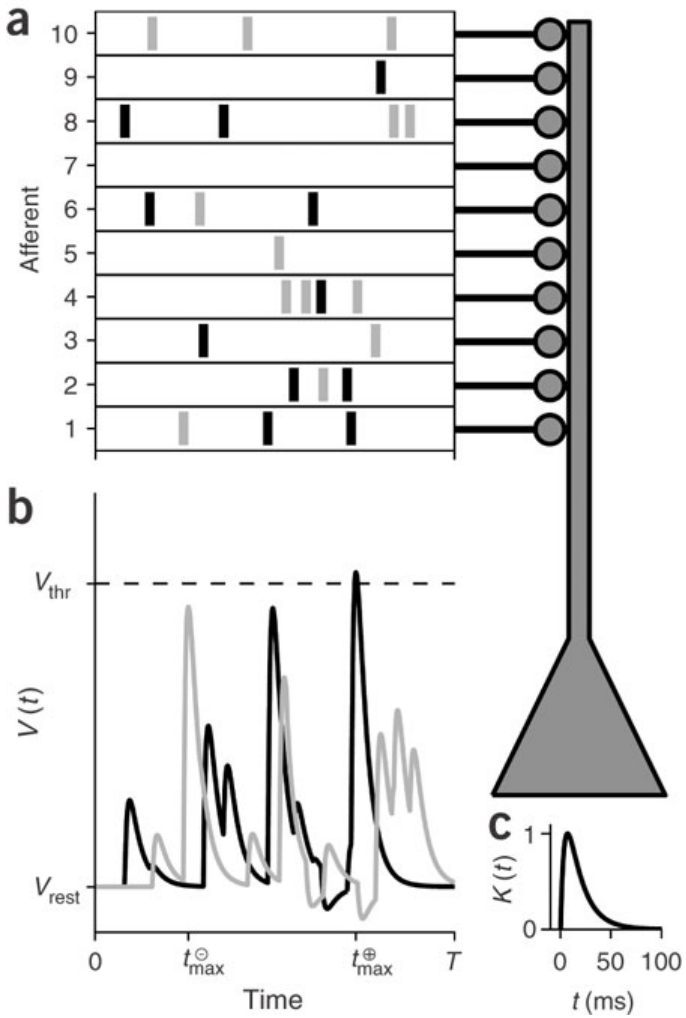
1. If a pre-synaptic spike arrives at the synaptic terminal before a post-synaptic spike is emitted, within a critical time window, the synaptic efficacy is increased.
2. If the post-synaptic spike is emitted soon before the pre-synaptic one arrives, the synaptic efficacy is decreased.

Abbot, Nelson, 2000

Hafliger et al. 1997, Bofil, Murray, 2001, Indiveri 2002, Arthur, Boahen 2005

STDP is not enough

for learning complex spatio-temporal patterns

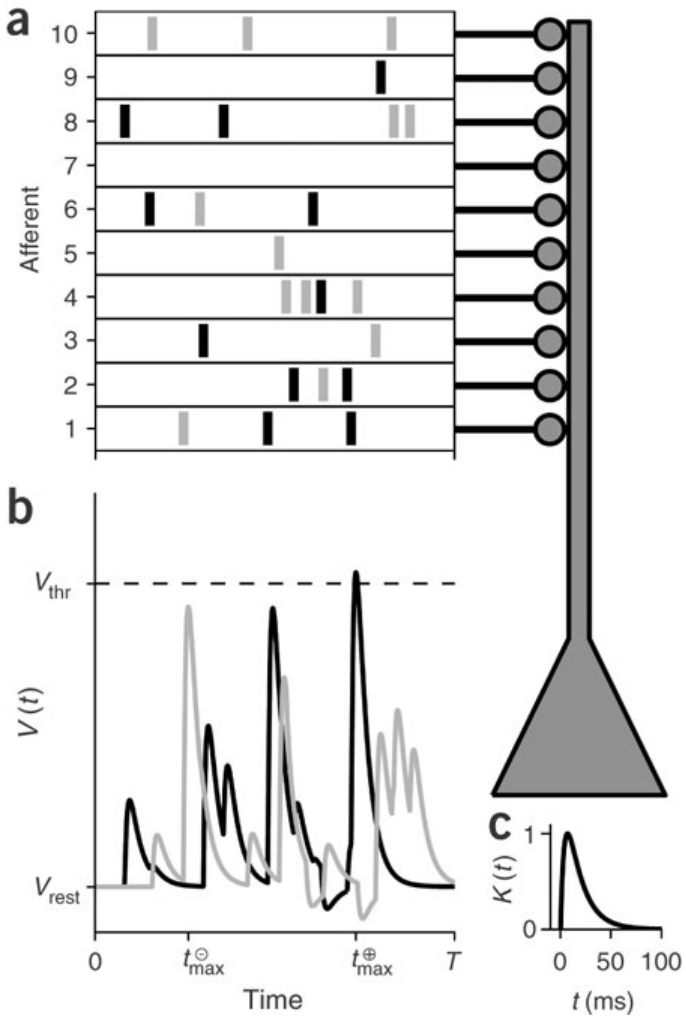


Senn, Biological Cybernetics, 2002

[...] additional nonlinearities are required if STDP should be relevant for both encoding information represented in a spike correlation code and a mean rate code without spike correlations

STDP is not enough

for learning complex spatio-temporal patterns

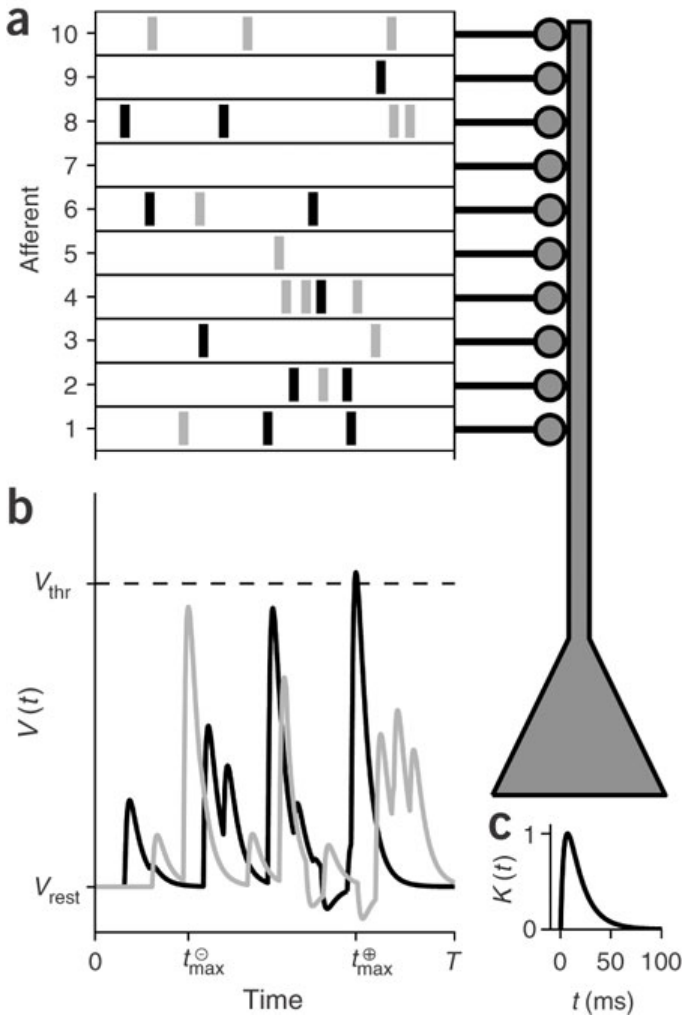


Lisman and Spruston, Nature Neuroscience, 2005

Postsynaptic depolarization requirements for LTP and LTD: a critique of spike timing-dependent plasticity

STDP is not enough

for learning complex spatio-temporal patterns



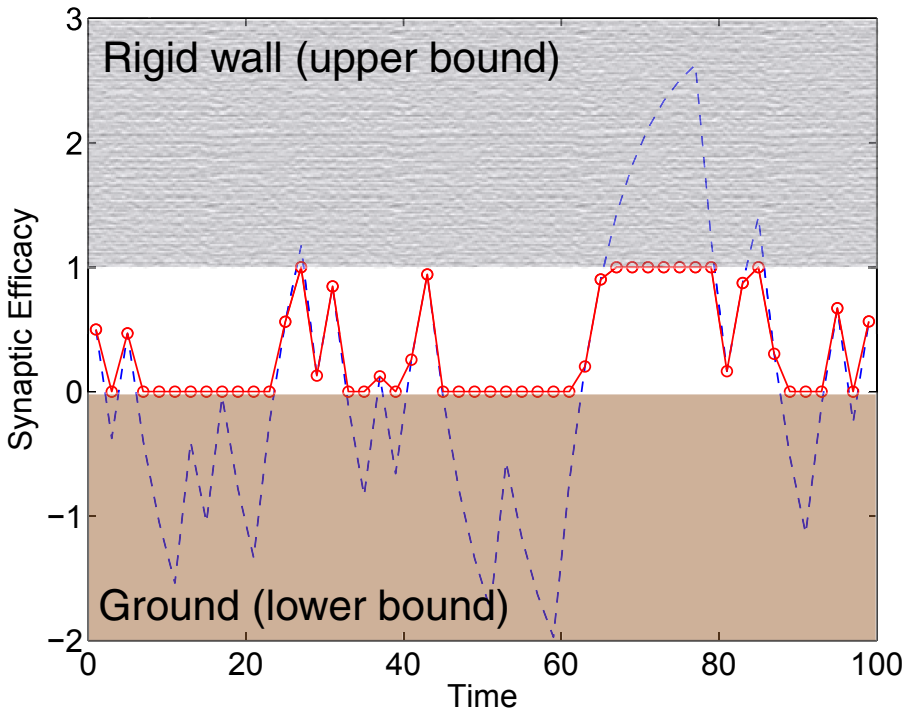
Gütig and Sompolinsky, Nature Neuroscience, 2006

We propose a new, biologically plausible supervised synaptic learning rule that enables neurons to efficiently learn a broad range of decision rules, even when information is embedded in the spatio-temporal structure of spike patterns rather than in mean firing rates. [...]
Our neuron model consists of a leaky integrate-and-fire neuron driven by exponentially decaying synaptic currents [...]

Physical implementations of learning mechanisms

Hard constraints

When constructing physical implementations of learning mechanisms one is immediately confronted with two hard constraints on the synaptic efficacies:



1. Limited dynamic range
2. Limited resolution

Physical implementations of plastic synapses

Limited resolution

Maximum number of patterns p that can be stored in an associative memory:

$$p < -\frac{\log(N_{syn})}{\left(\sqrt{Q_+} - \sqrt{Q_-}\right)^2 + \left(\frac{\sqrt{Q_+ Q_-}}{n^2}\right)}$$

- N_{syn} =Total number of synapses.
- n = number of synaptic states.
- Q_+ =LTP Probability
- Q_- =LTD Probability

Balanced case: $p \approx n^2$.

Inbalanced case: p does not depend on n.

Fusi, Abbot, *Nature Neuroscience* 2007

Recipe for efficient VLSI implementation

Basic specifications:

1. Bistability: use just two synaptic states ($n = 2$).
2. Redundancy: implement many (imprecise) synapses that see the same pre- and post-synaptic activity.
3. Stochasticity & inhomogeneity: induce LTP or LTD in a randomly selected subset of synapses.

Pros and Cons

- \- Requires a large number of synapses.
- \- Learning is slow: only a fraction of the synapses memorize the training pattern.
- \+ We are not fighting against technology (large-scale integration, mismatch and fault tolerance).

Recipe for efficient VLSI implementation

Basic specifications:

1. Bistability: use just two synaptic states ($n = 2$).
2. Redundancy: implement many (imprecise) synapses that see the same pre- and post-synaptic activity.
3. Stochasticity & inhomogeneity: induce LTP or LTD in a randomly selected subset of synapses.

Pros and Cons

- \- Requires a large number of synapses.
- \- Learning is slow: only a fraction of the synapses memorize the training pattern.
- \+ We are not fighting against technology (large-scale integration, mismatch and fault tolerance).

A new spike-based synaptic plasticity mechanism

Stop Learning

Update weight only when necessary. Stop learning if

1. the desired activity is high and the neuron responds with a high activity to the input pattern
2. the desired activity is low and the neuron responds with a low activity to the input pattern

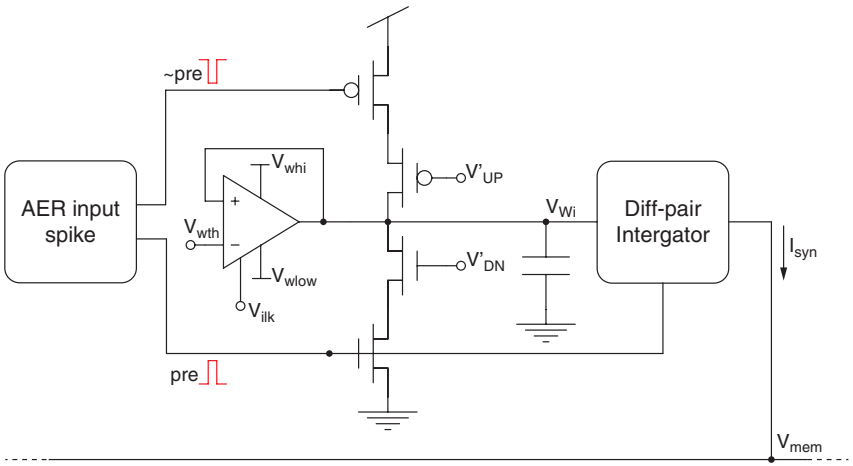
Powerful

Can learn complex correlated patterns. SW simulations comparable with state of the art classifiers (*e.g.* MNIST character classification benchmark).

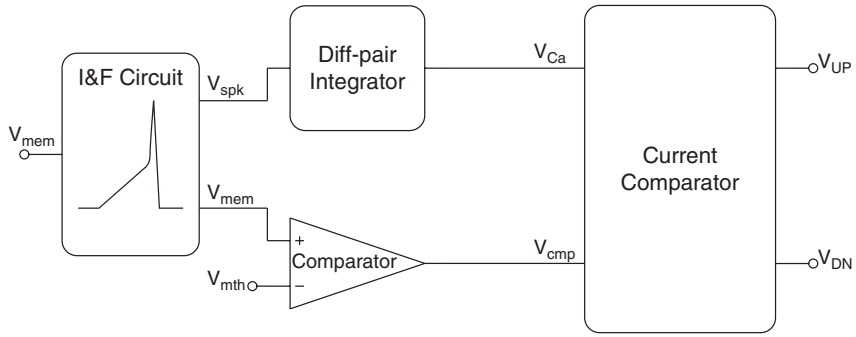
Brader et al., Neural Computation, 2007

Spike-based plasticity circuits

Weight update

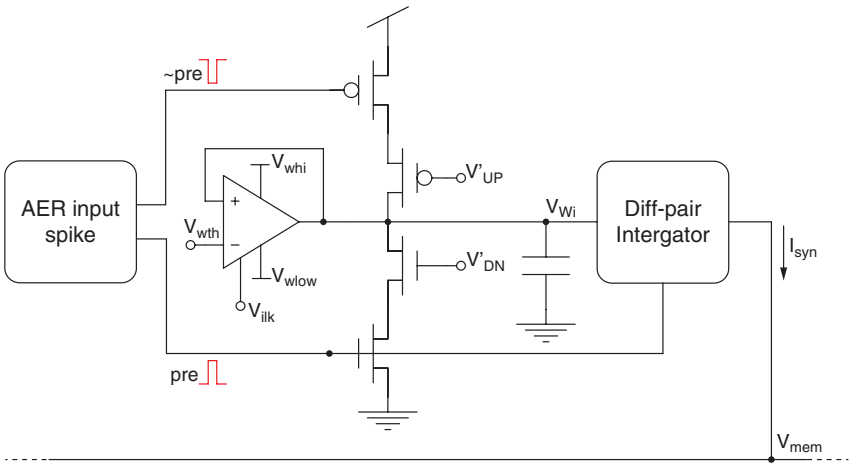


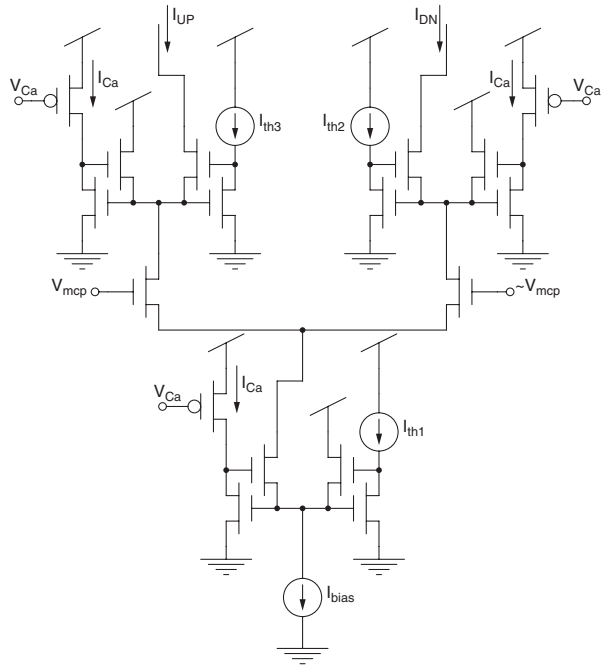
Stop learning



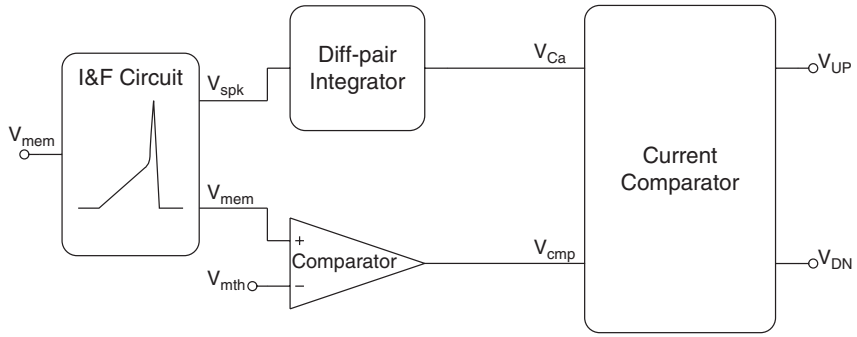
Spike-based plasticity circuits

Weight update



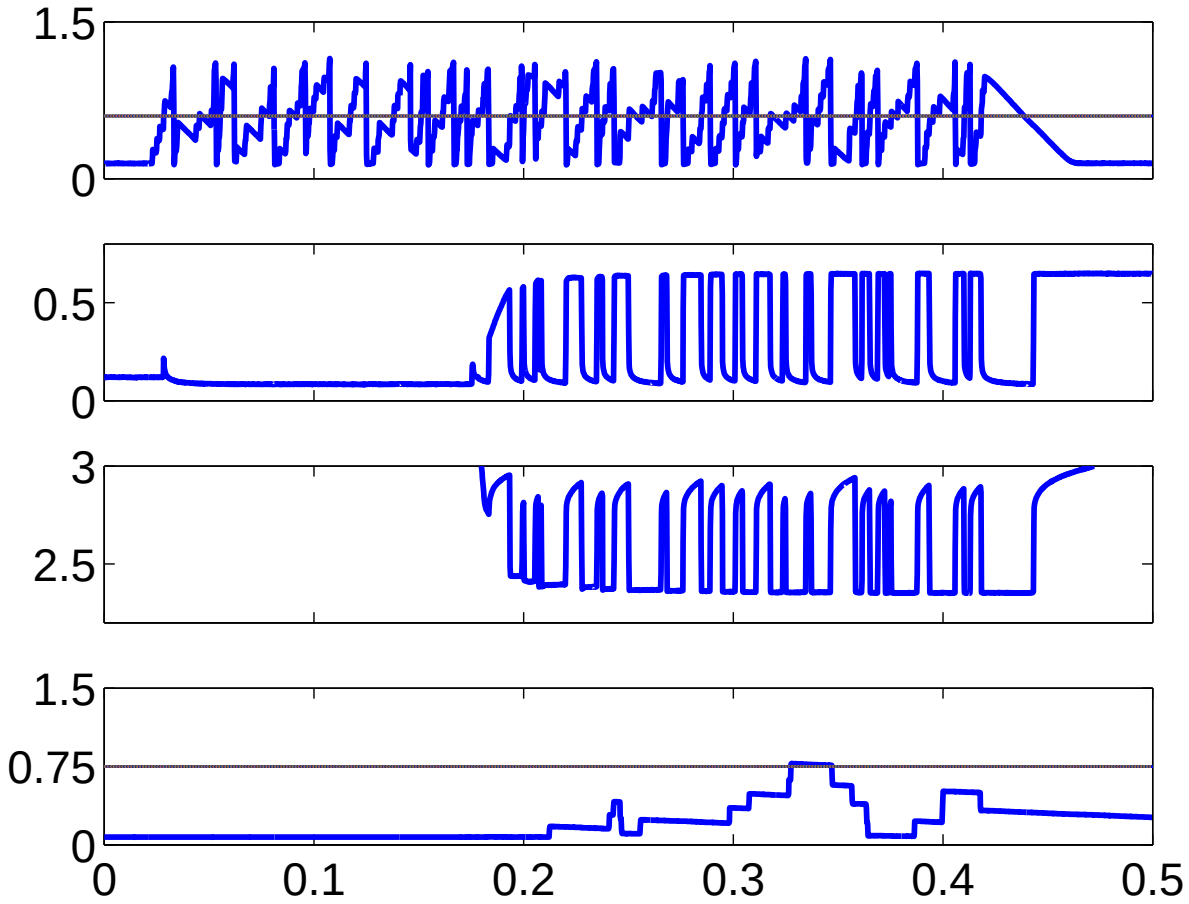


Stop learning



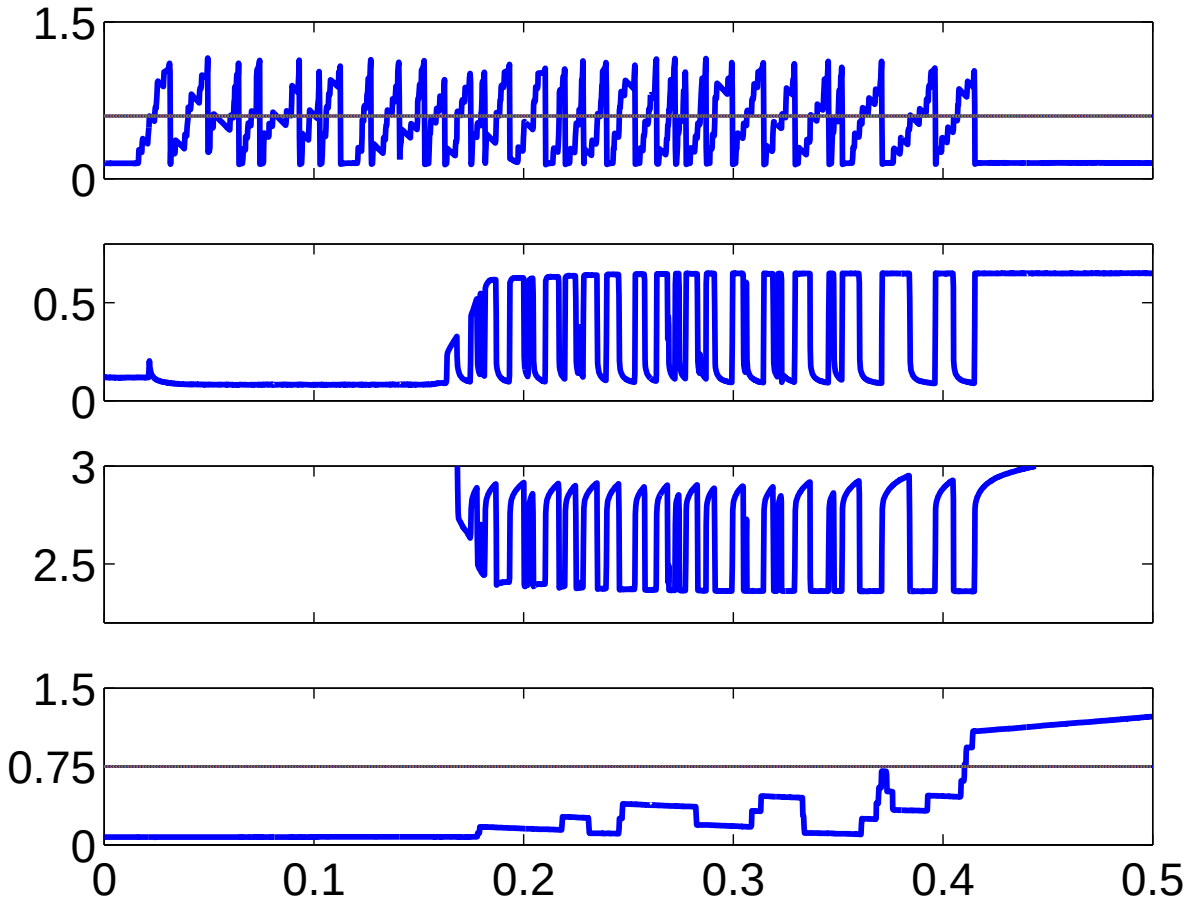
Stochastic LTP

Single synapse



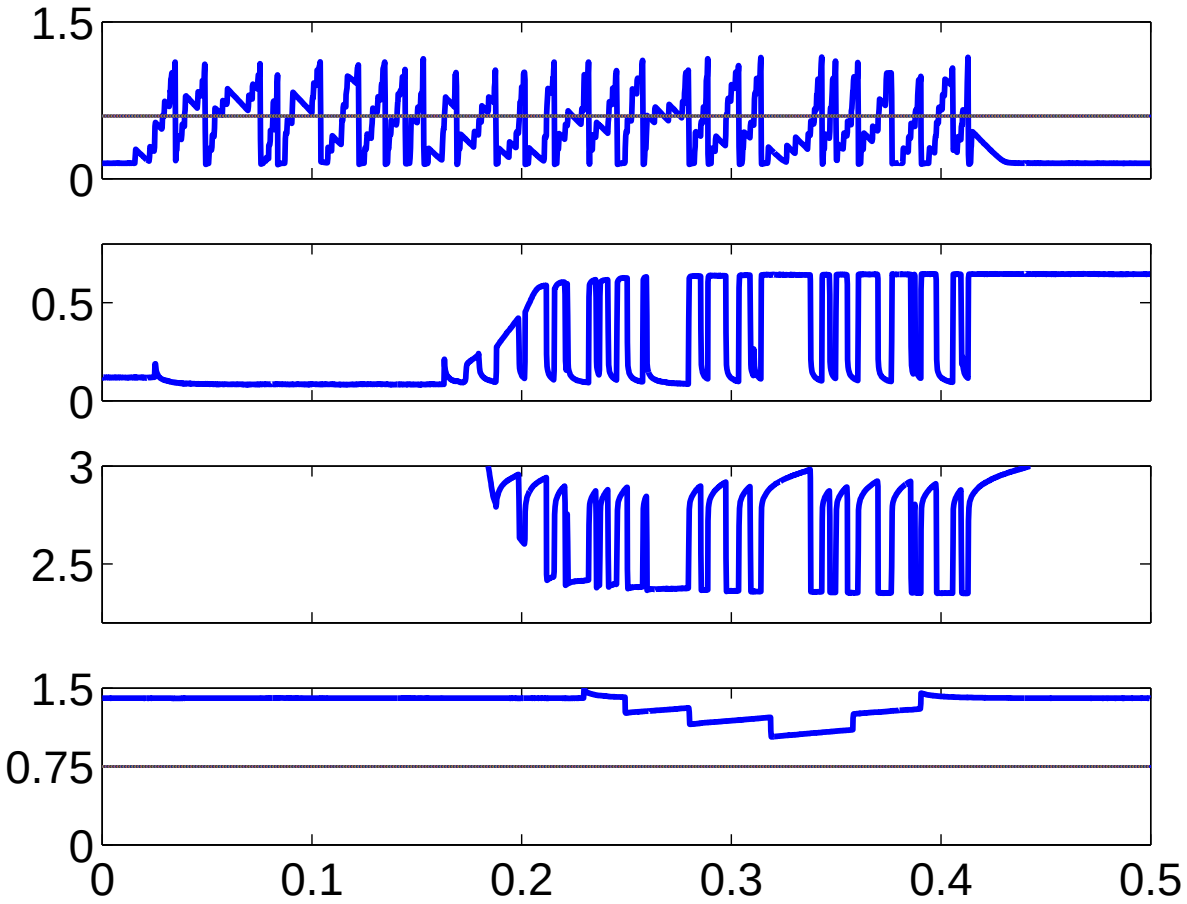
Stochastic LTP

Single synapse



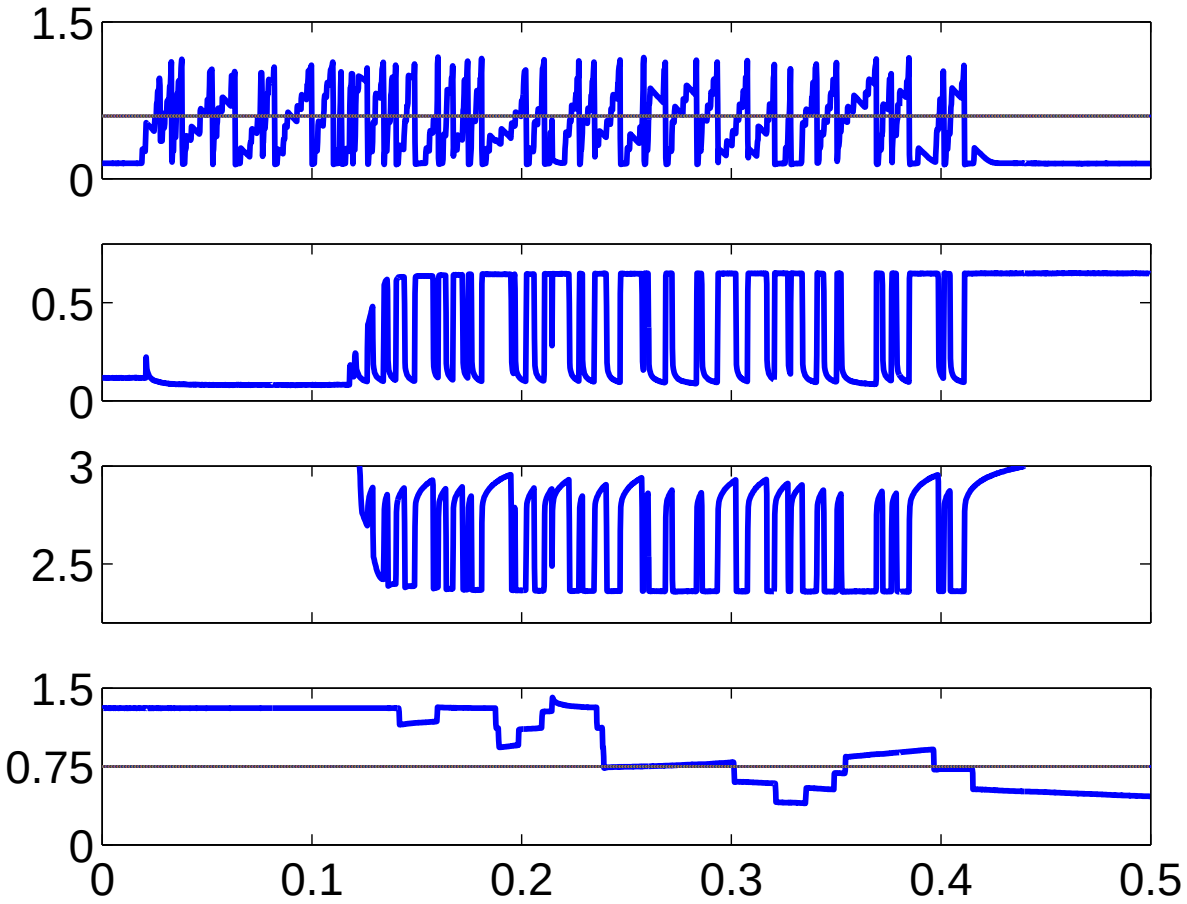
Stochastic LTP

Single synapse



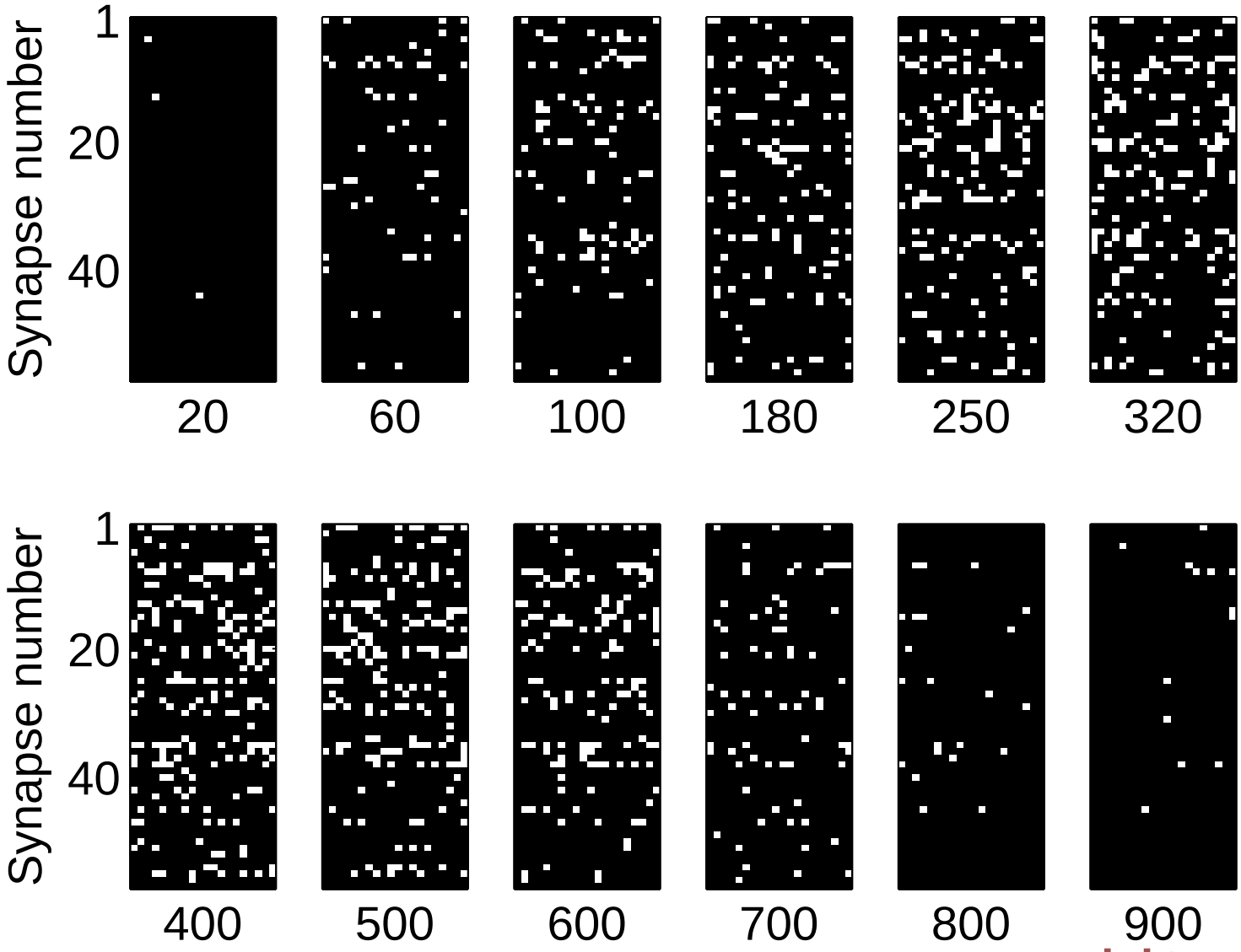
Stochastic LTP

Single synapse



Stochastic LTP

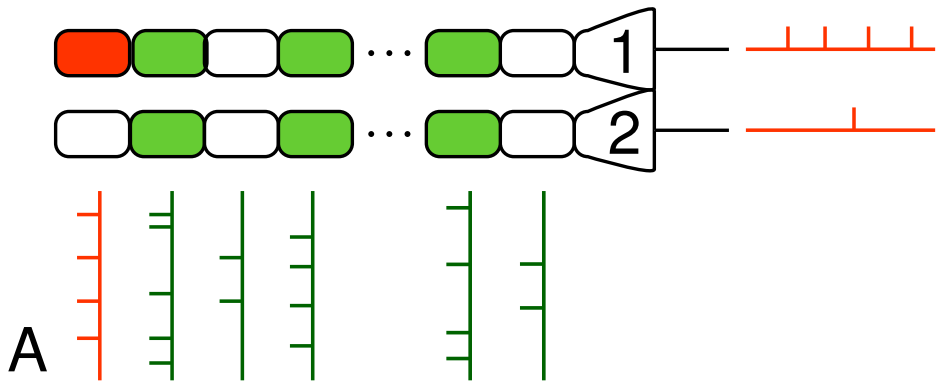
Controlling the LTP transition probability



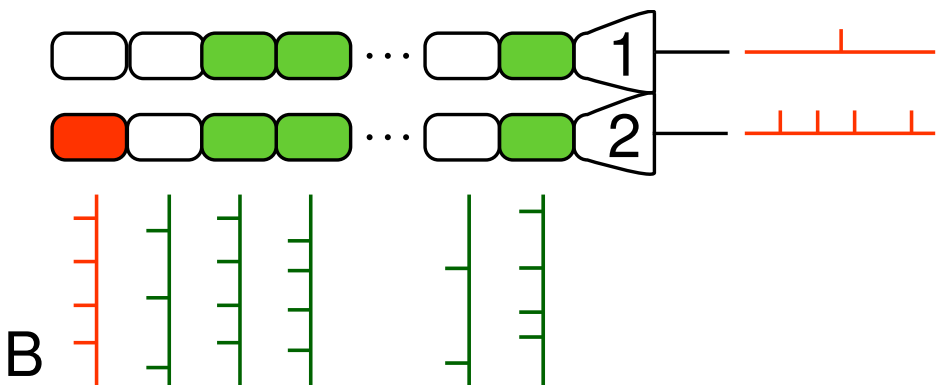
Classifying Random Patterns

Training phase

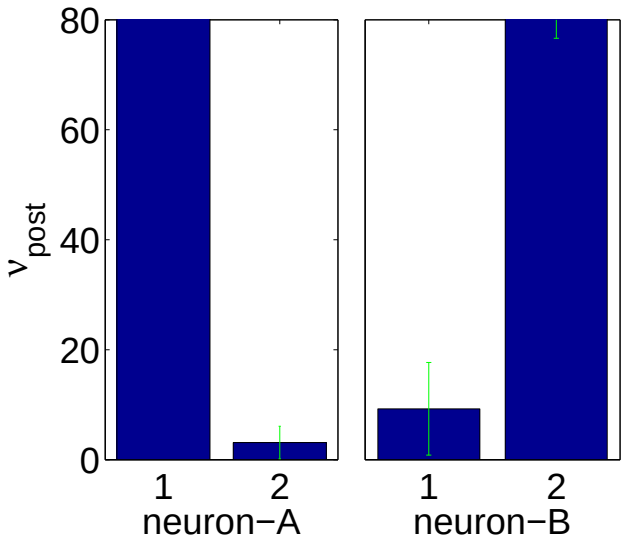
Train Neuron 1 - Class A



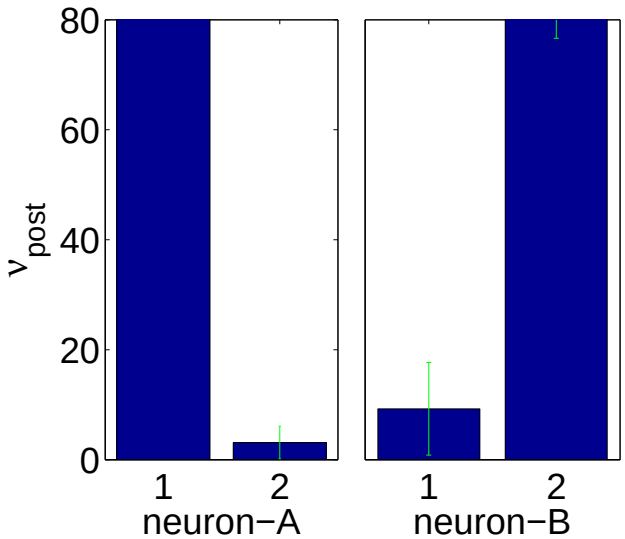
Train Neuron 2 - Class B

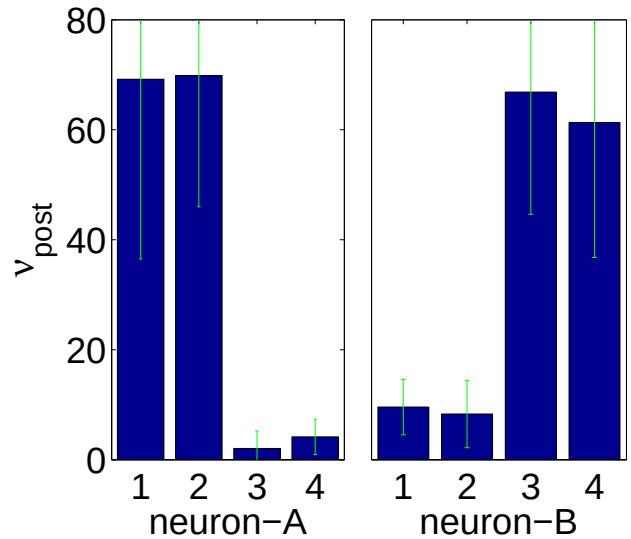


Random binary patterns

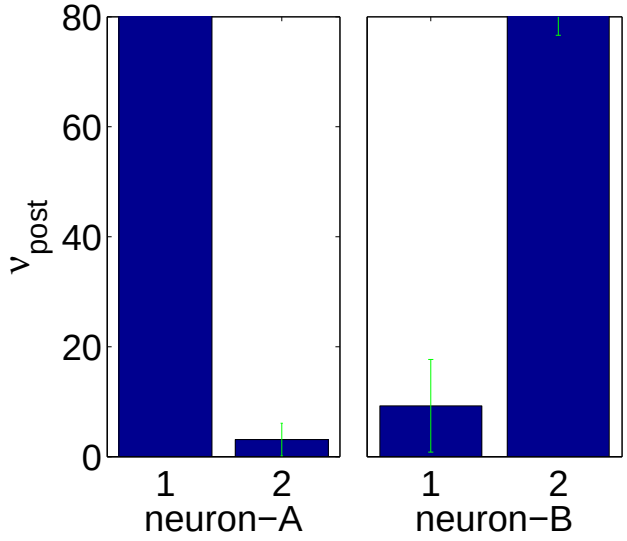


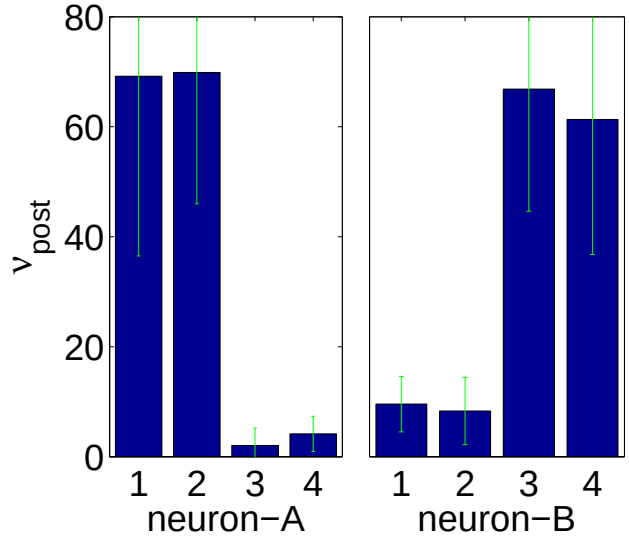
Random binary patterns

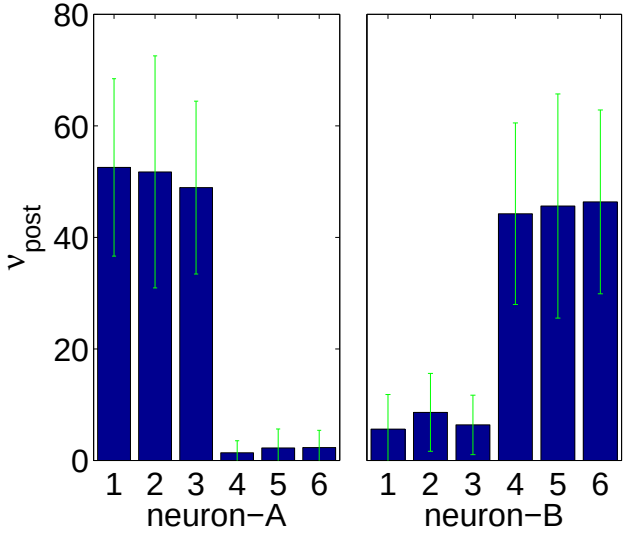




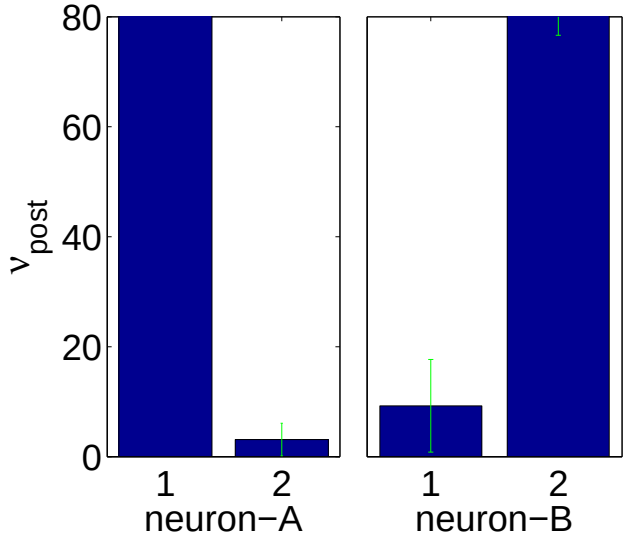
Random binary patterns

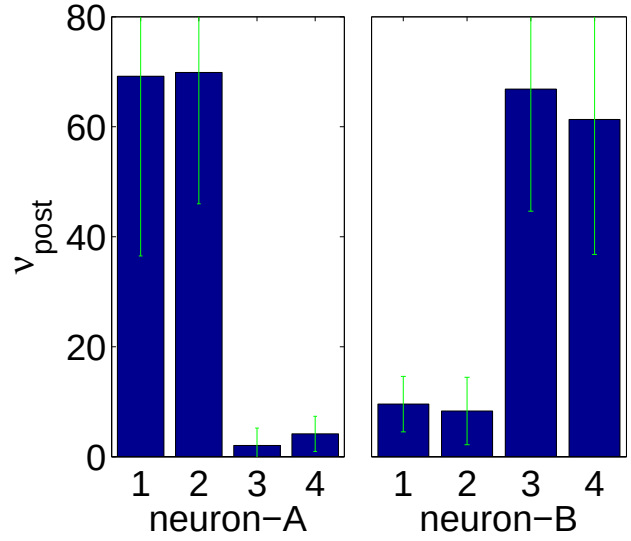


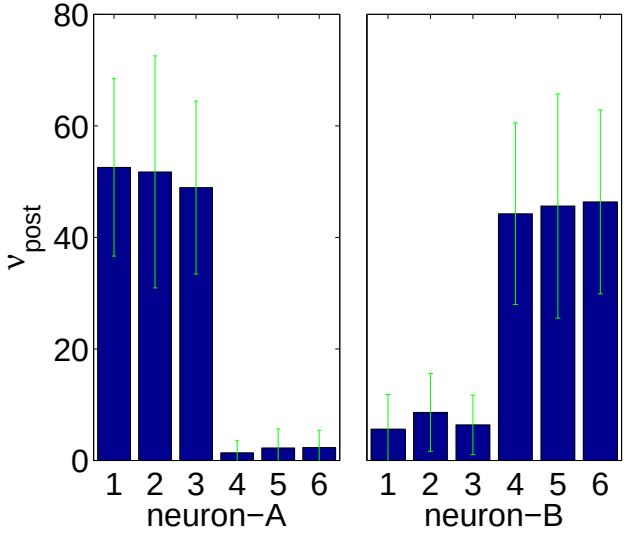


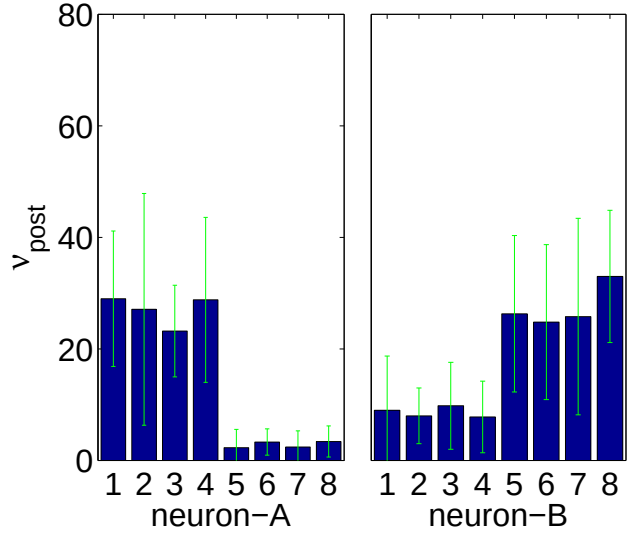


Random binary patterns



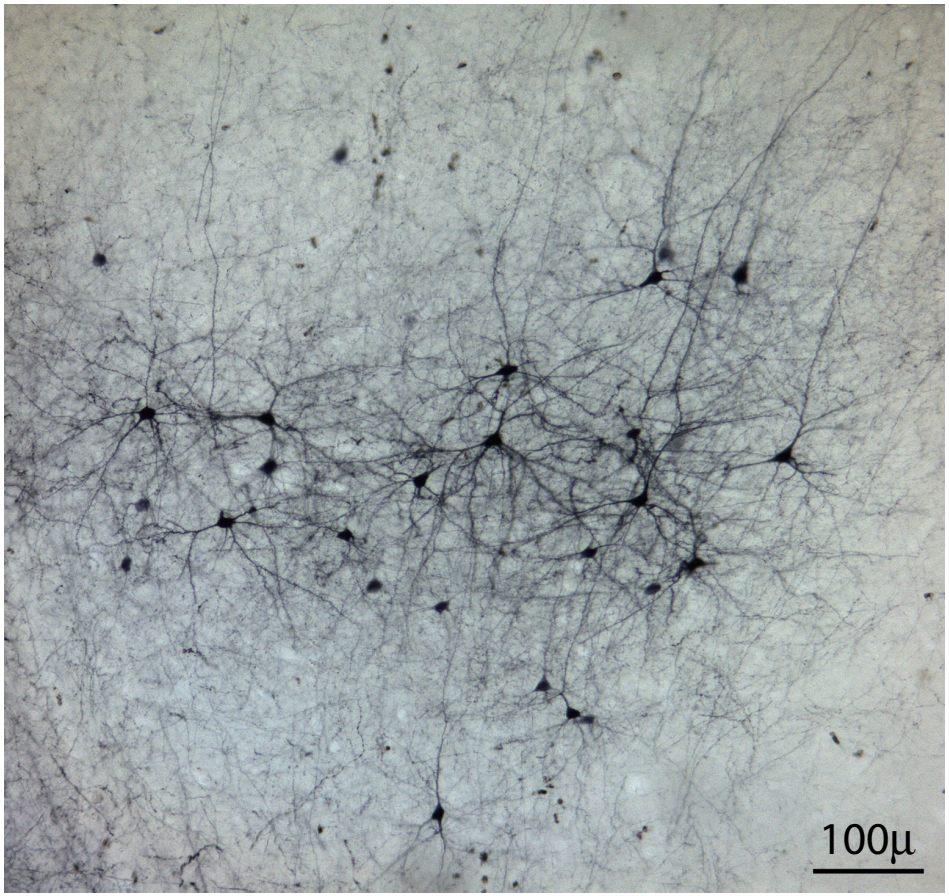


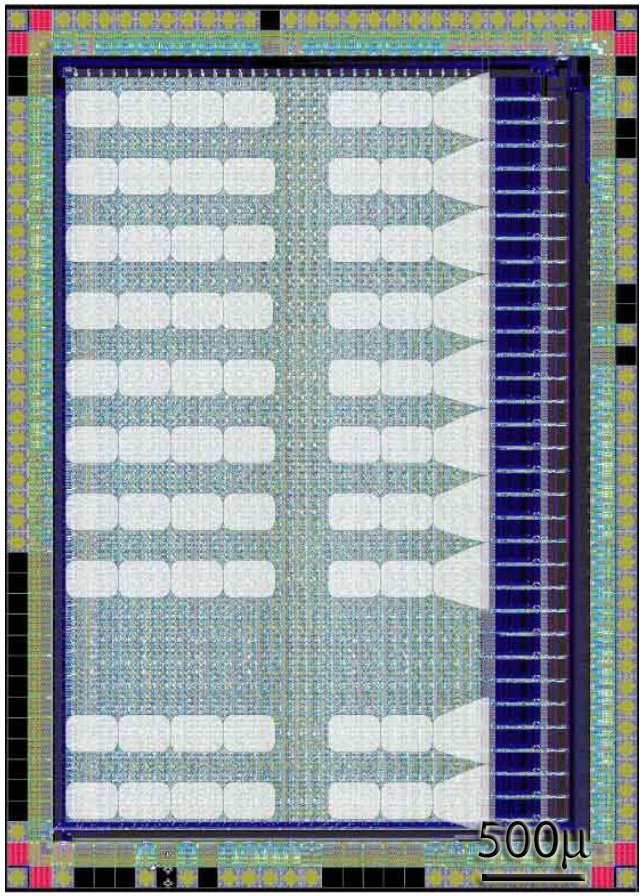




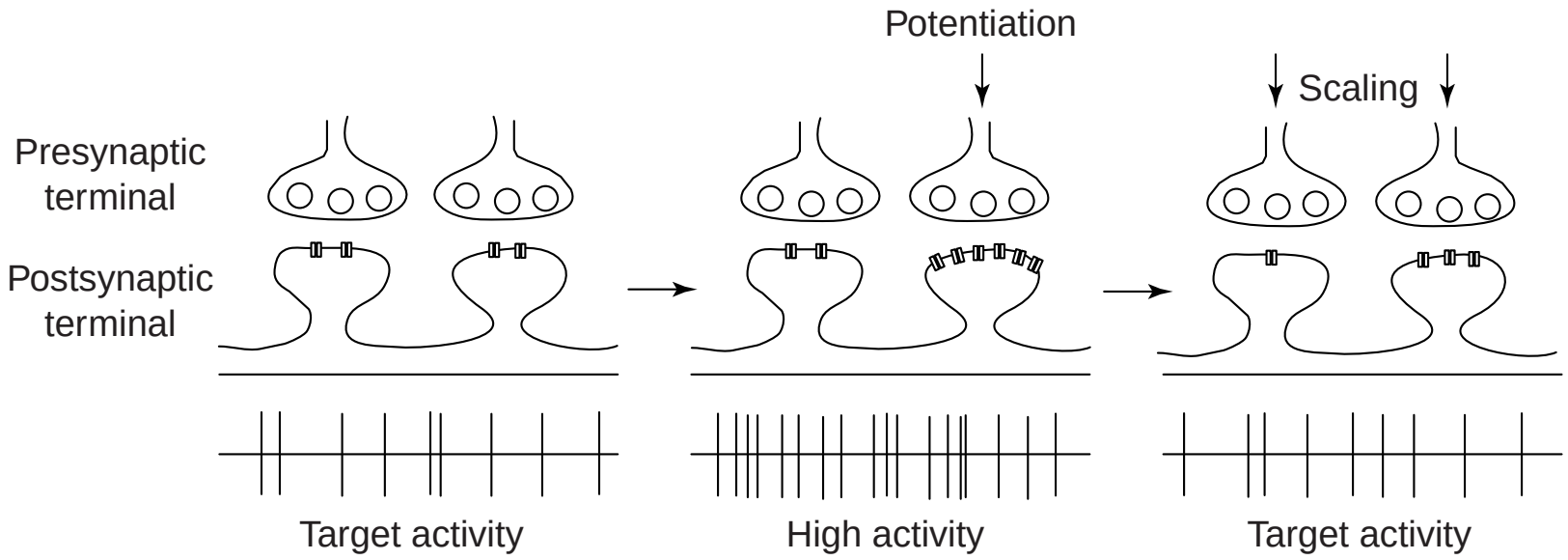
Outline

Homeostasis

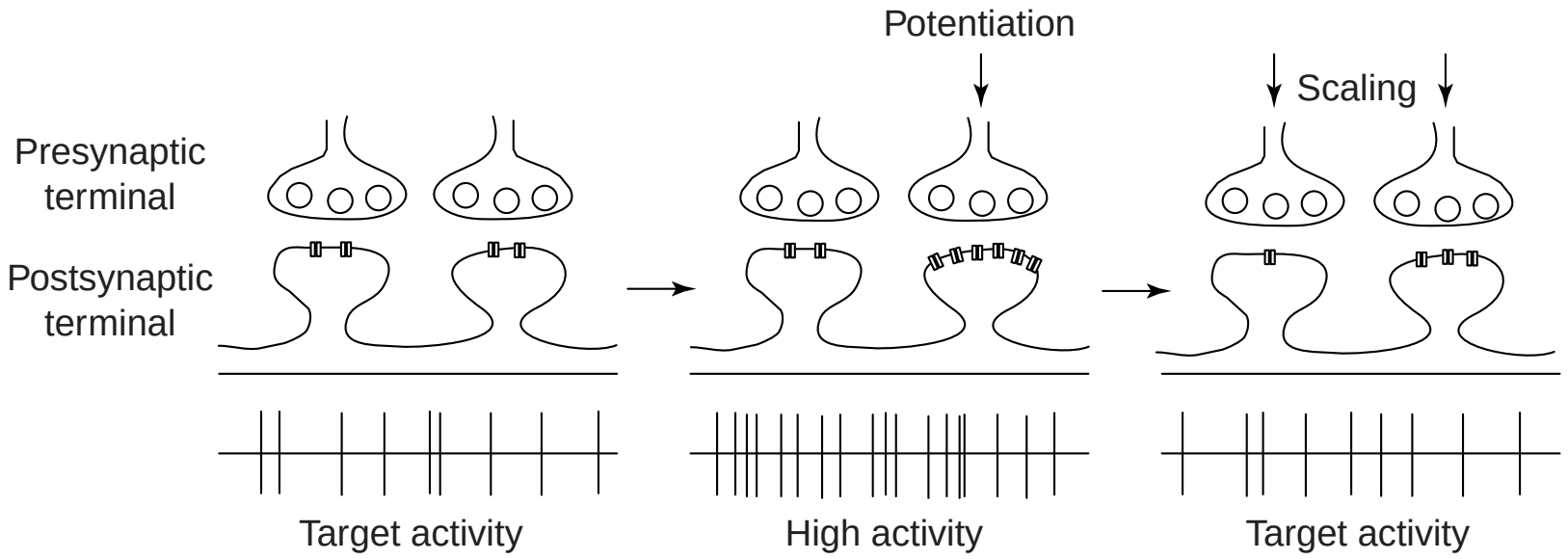


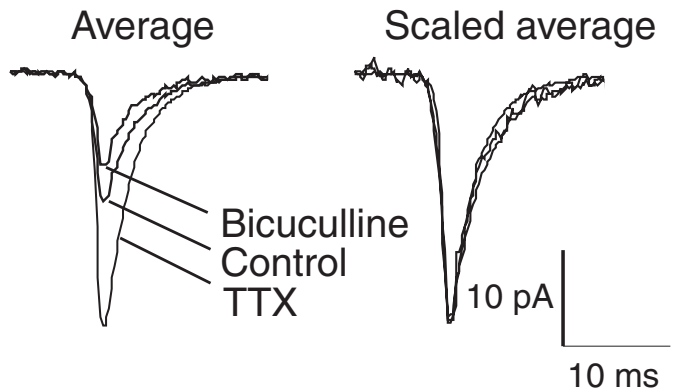


Synaptic scaling



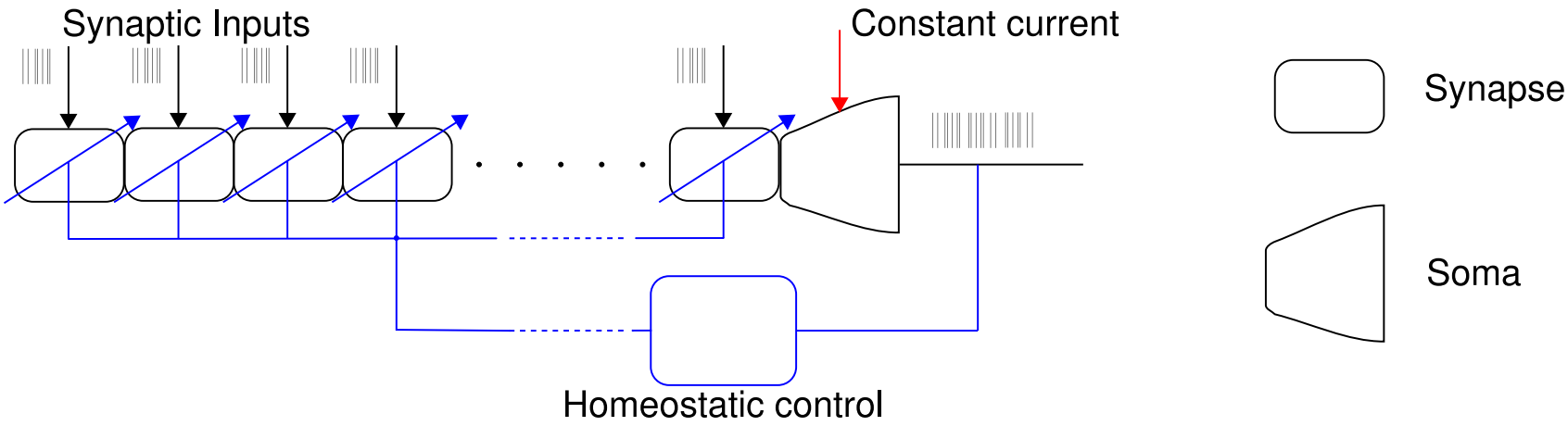
Synaptic scaling



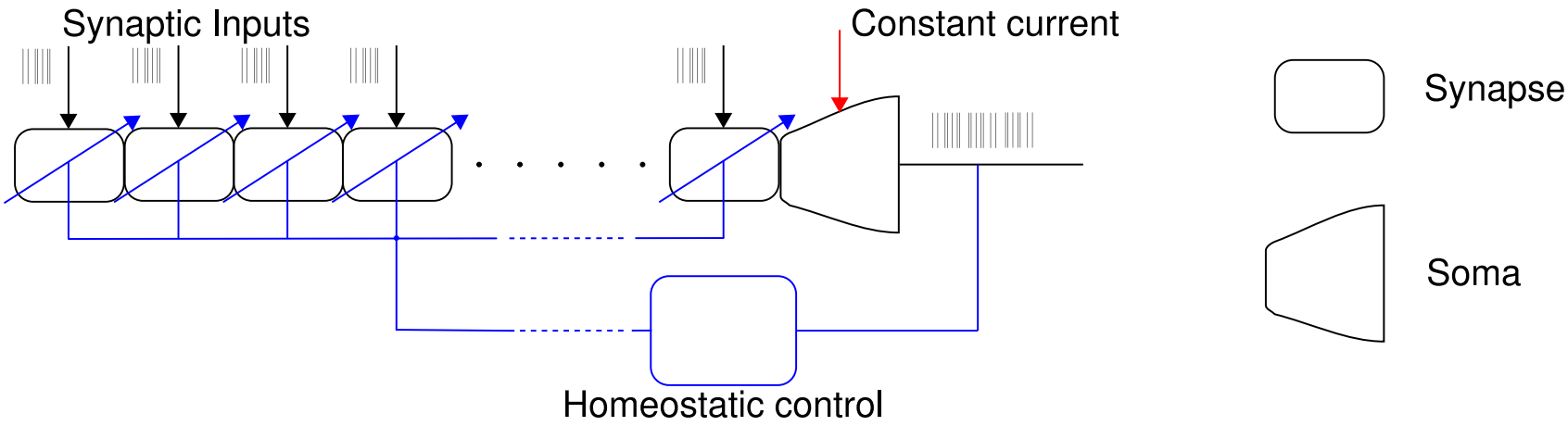


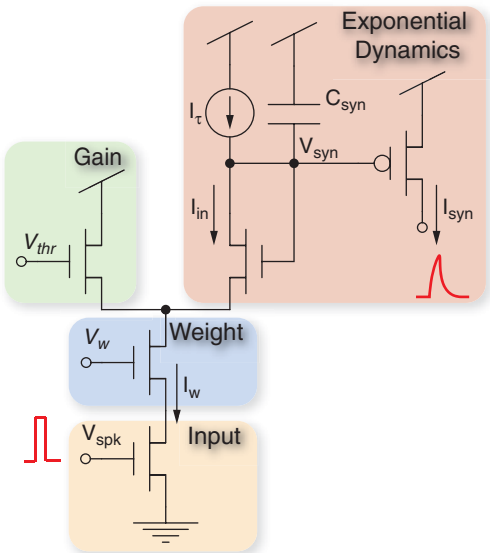
Turrigiano et al., *Nature*, 1998

Synaptic scaling in silicon

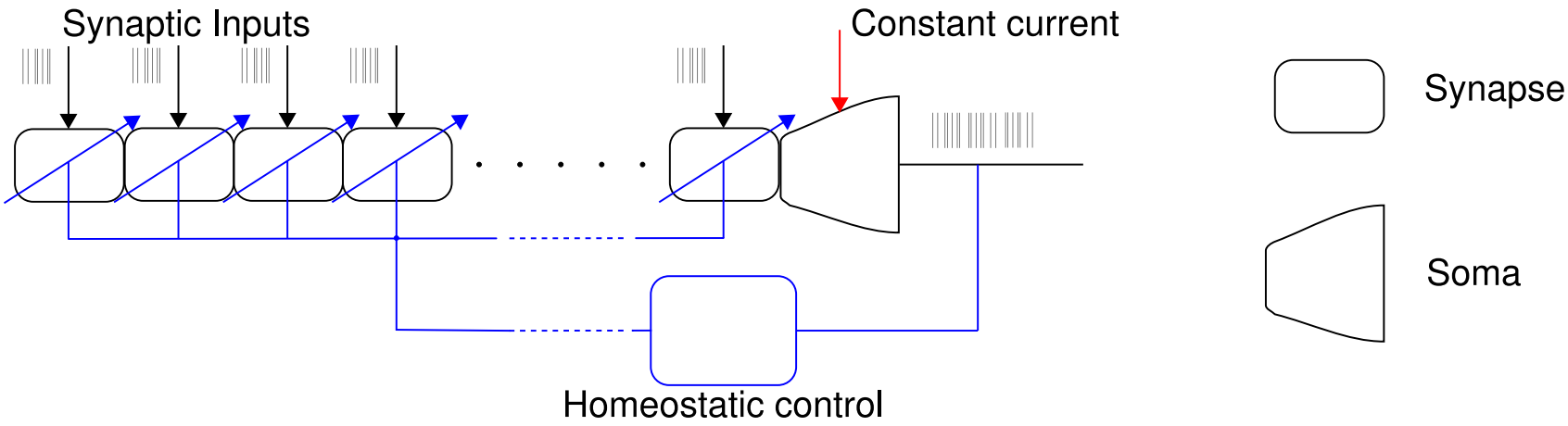


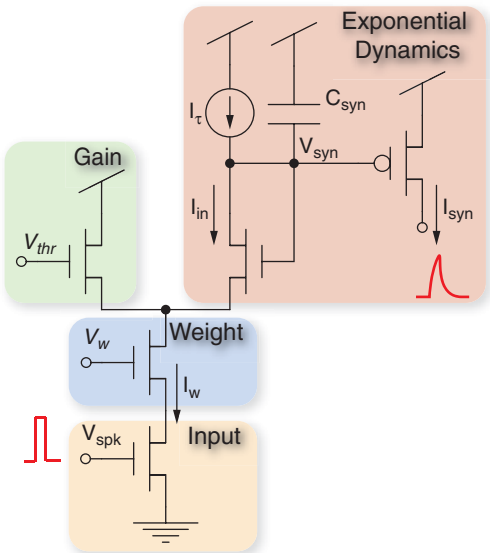
Synaptic scaling in silicon

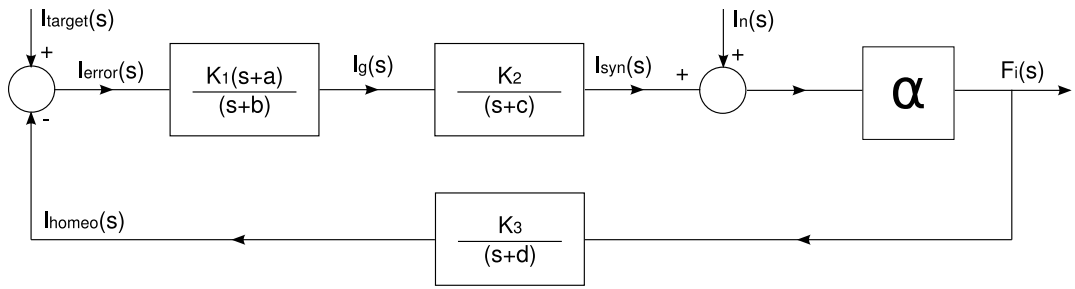




Synaptic scaling in silicon

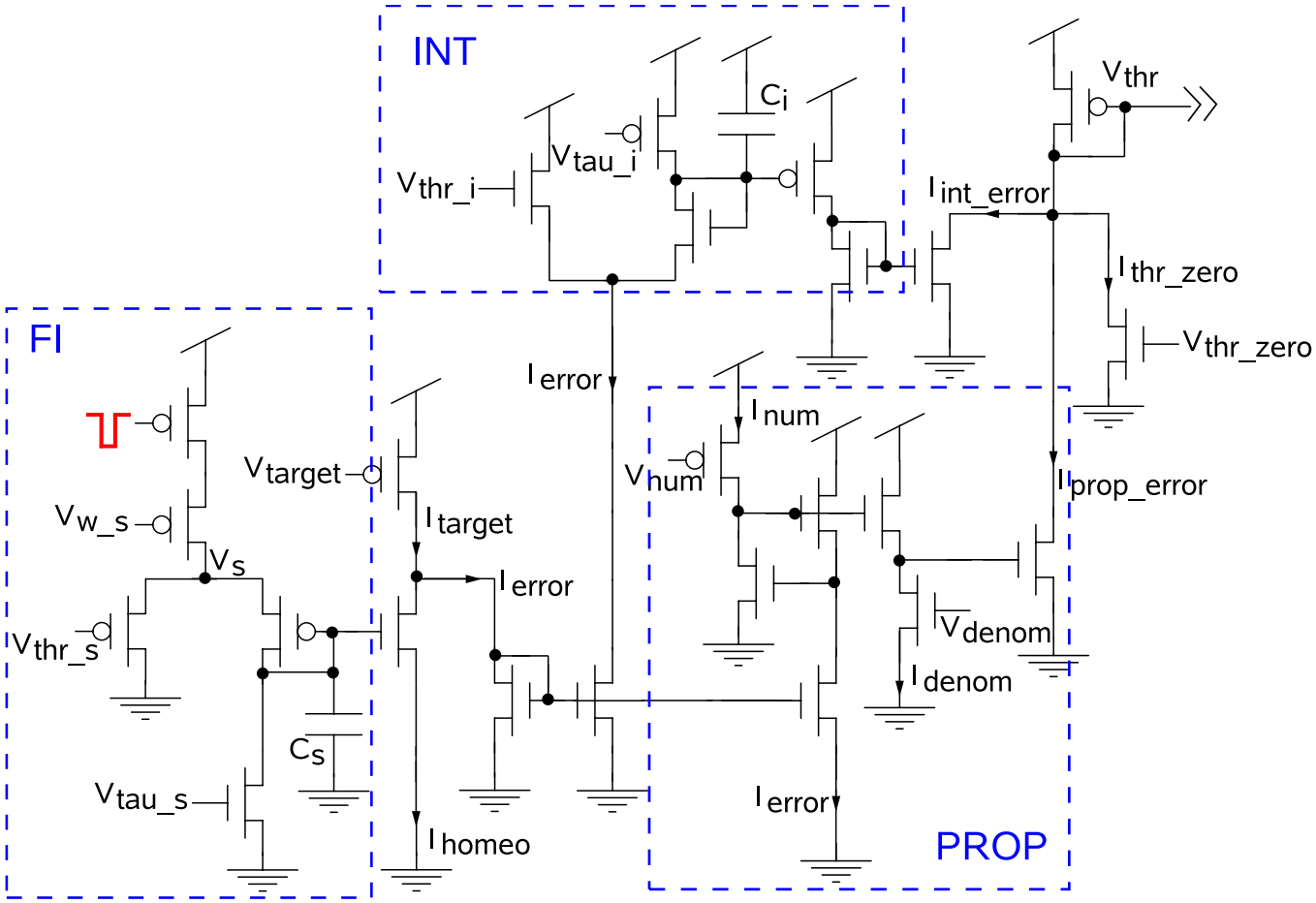






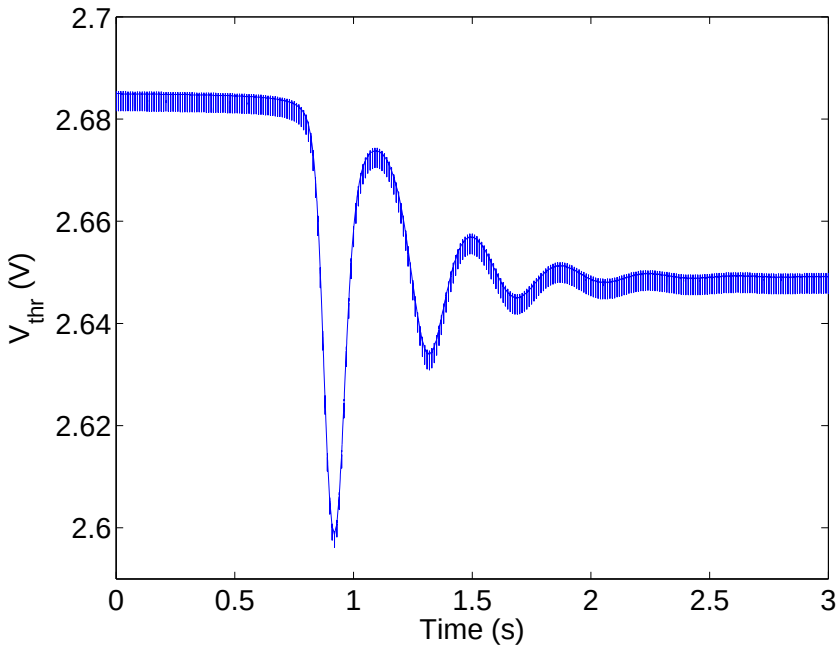
Nikolayeva et al. *Neurocomputing*, 2007

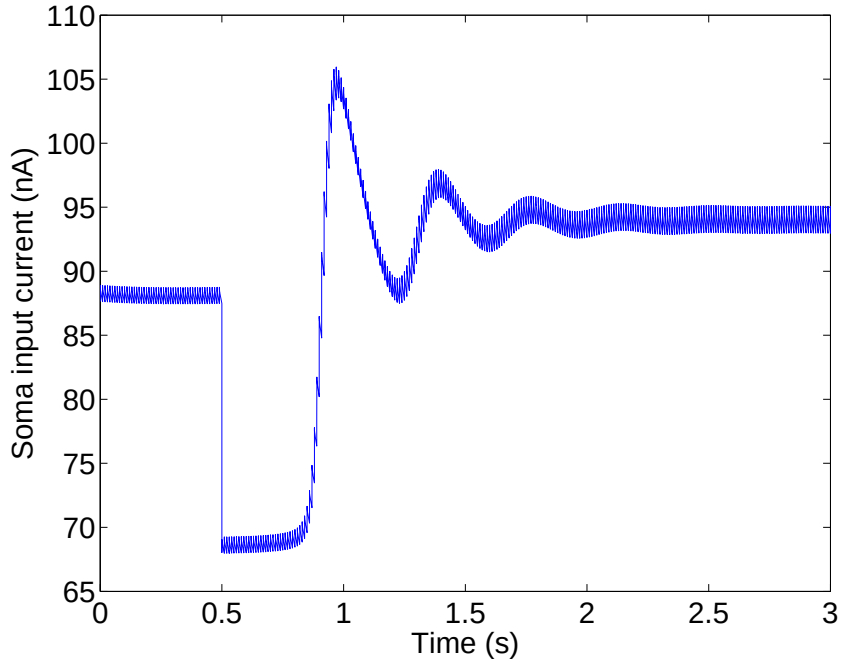
Homeostatic control circuit



Homeostatic control circuit

Response to a decrease of soma input current (at $t = 0.5$s).

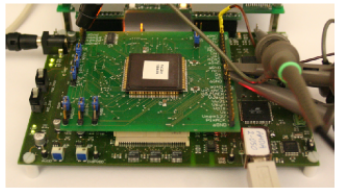


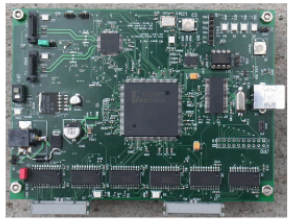


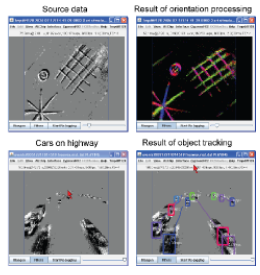
Outline

Putting all the pieces together...

...and getting them to work



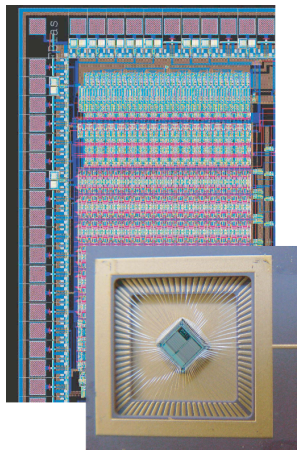


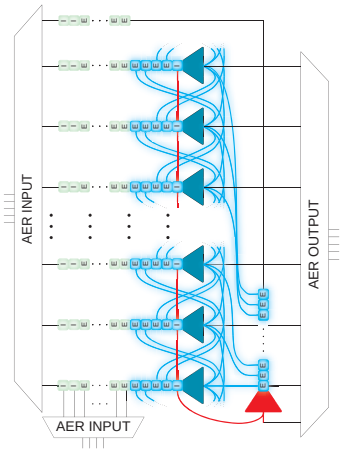


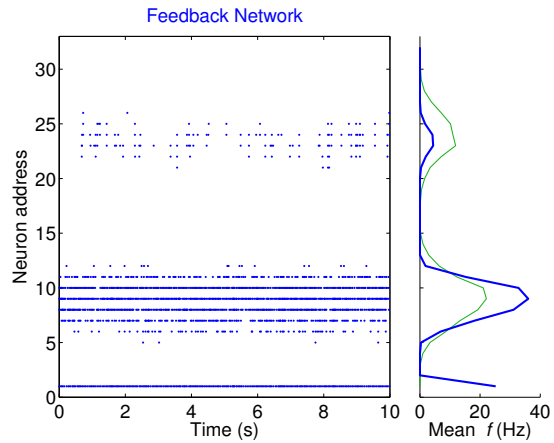
1. Neuromorphic VLSI chips (sensors, learning chips, analog filters, . . .).
2. Industry-strength commodity circuits (asynchronous logic, temperature compensated bias generators, . . .).
3. Printed Circuit Boards (PCBs) for hosting the neuromorphic chips.
4. On-board communication logic.
5. Off-board communication interfaces.
6. Board-PC interfaces.
7. Low-level PC drivers.
8. Software user-interfaces, Matlab toolboxes, *etc.*
9. . . .

Reliable computation

With plastic cooperative-competitive networks



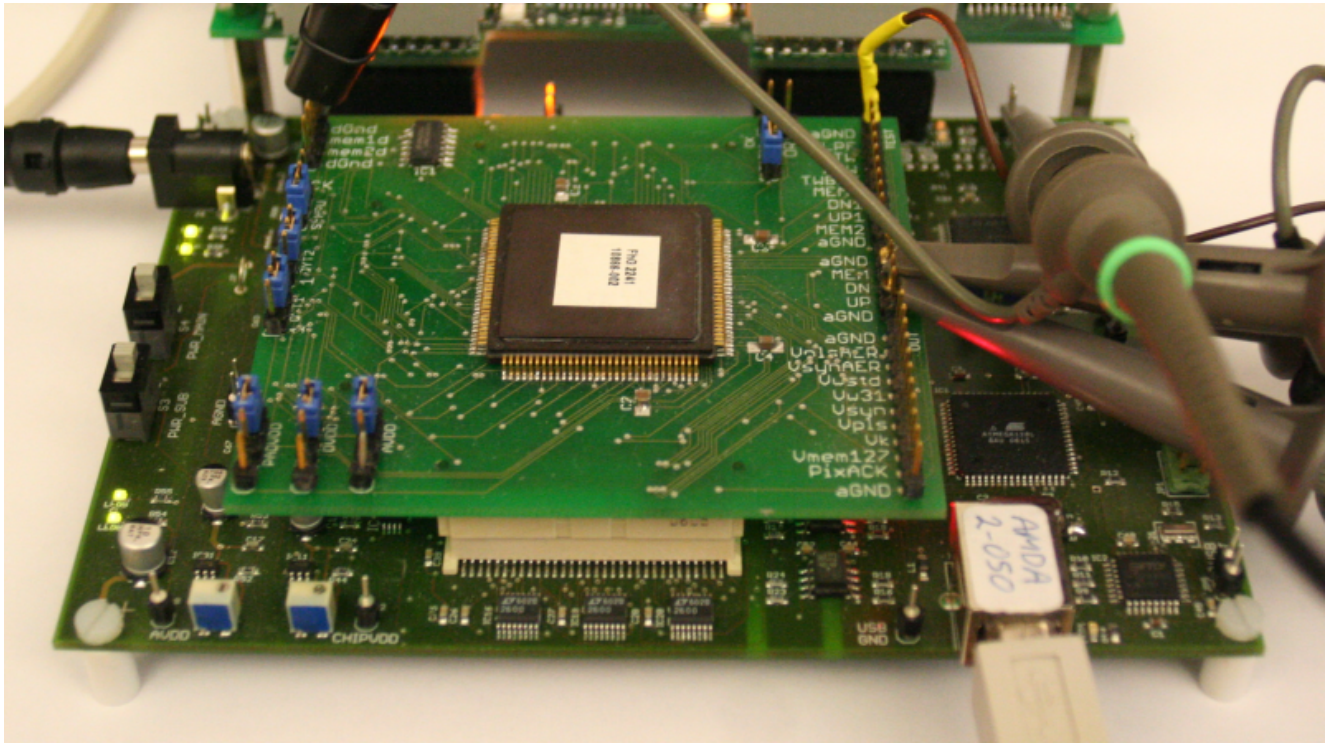




- Classify spatio-temporal patterns.
- Combine intrinsic & homeostatic plasticity with spike-based plasticity.
- Combine many parallel “weak” classifiers (boosting).
- Implement selective amplification and context dependent gain modulation.
- Implement “relational networks”.

Reliable chip configuration

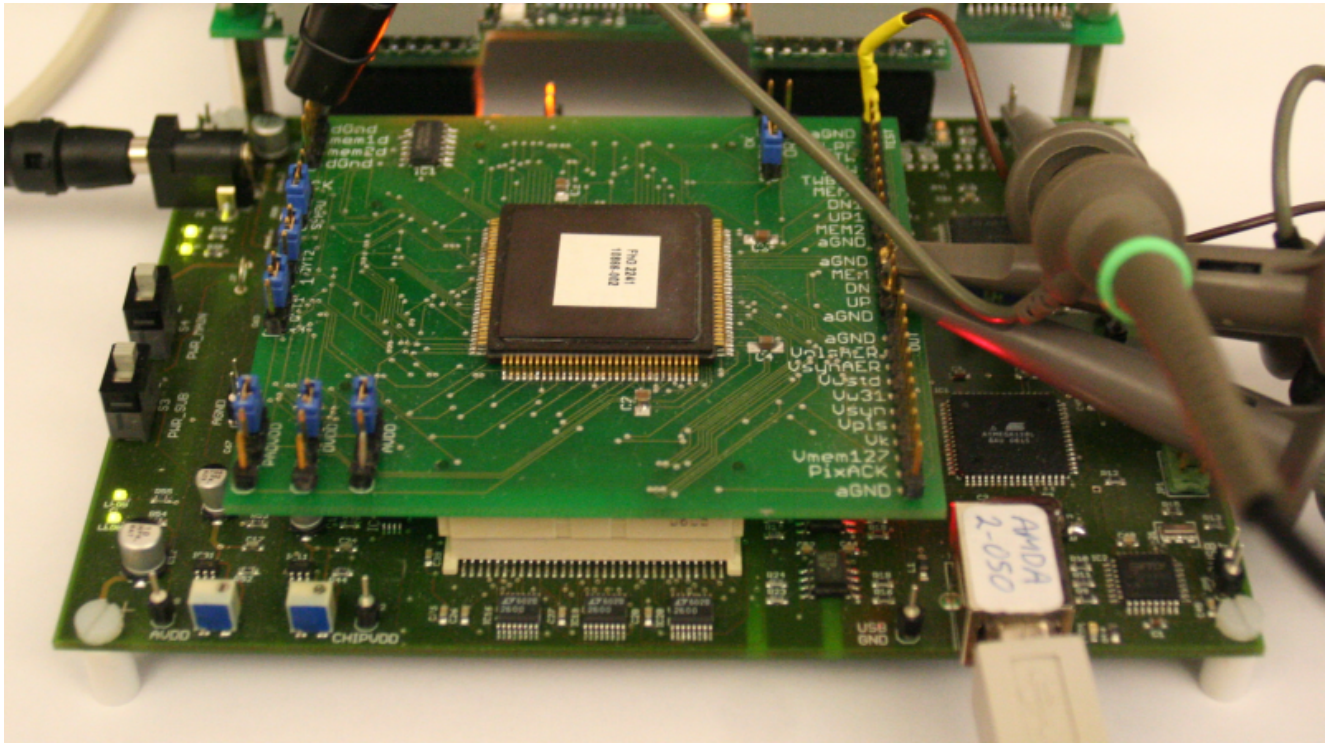
with an AMDA board



- USB protocol to interface to PCs.
- DACs and ADCs to configure chip parameters.
- Parallel AER circuits to interface to AER systems.
- Easy interchange of daughterboards

Reliable chip configuration

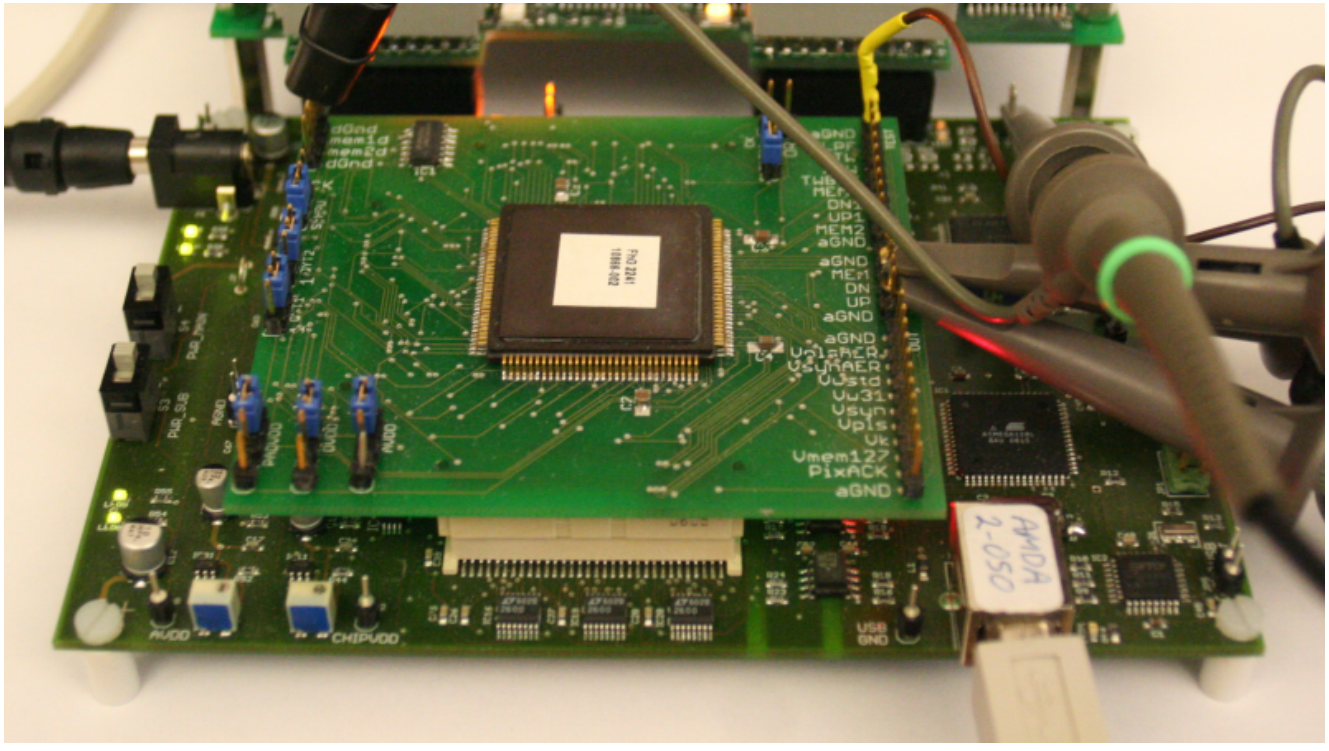
with an AMDA board



- USB protocol to interface to PCs.
- DACs and ADCs to configure chip parameters.
- Parallel AER circuits to interface to AER systems.
- Easy interchange of daughterboards

Reliable chip configuration

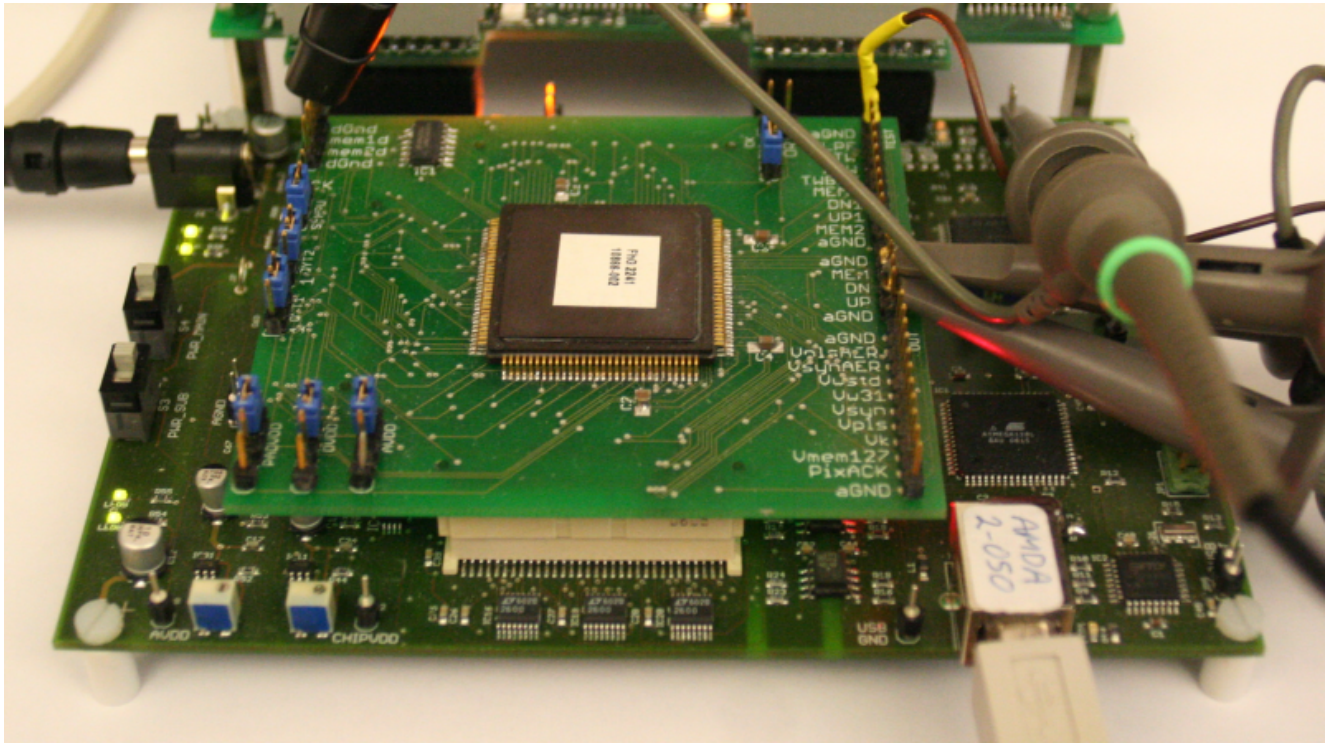
with an AMDA board



- USB protocol to interface to PCs.
- DACs and ADCs to configure chip parameters.
- Parallel AER circuits to interface to AER systems.
- Easy interchange of daughterboards

Reliable chip configuration

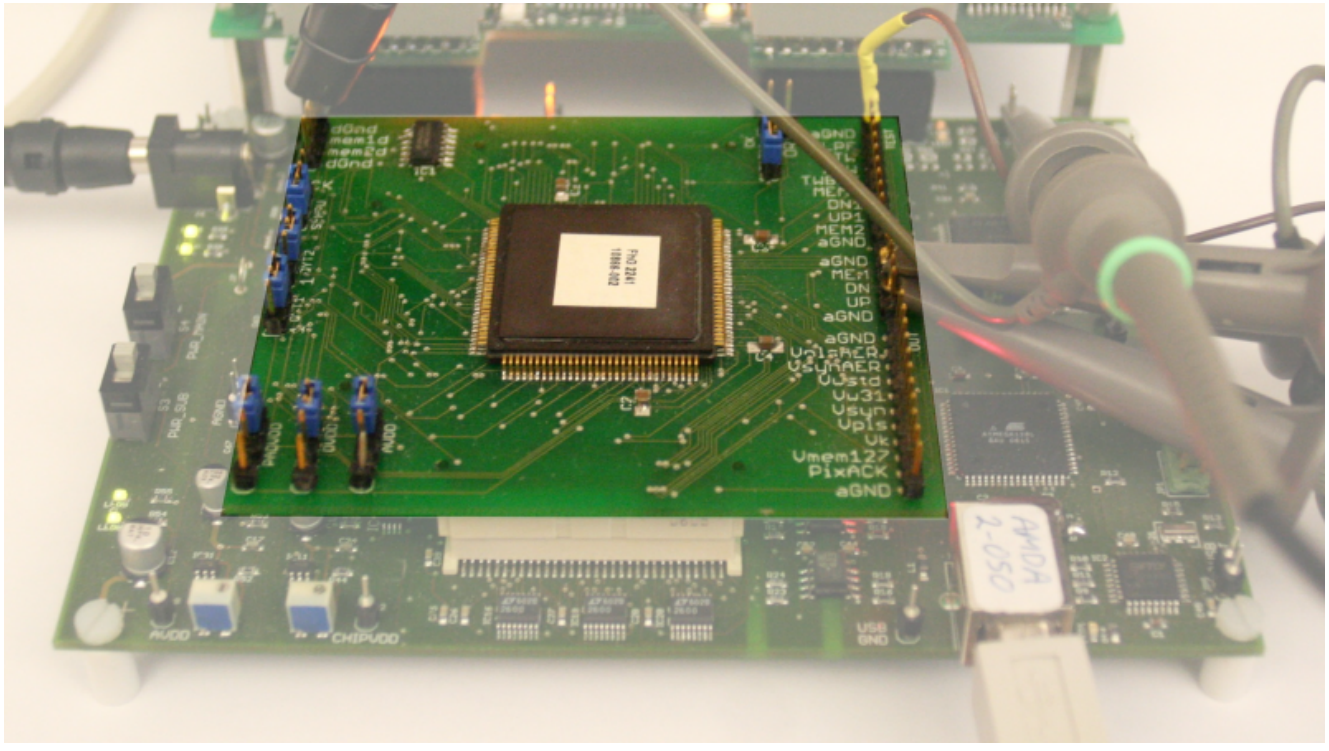
with an AMDA board



- USB protocol to interface to PCs.
- DACs and ADCs to configure chip parameters.
- Parallel AER circuits to interface to AER systems.
- Easy interchange of daughterboards

Reliable chip configuration

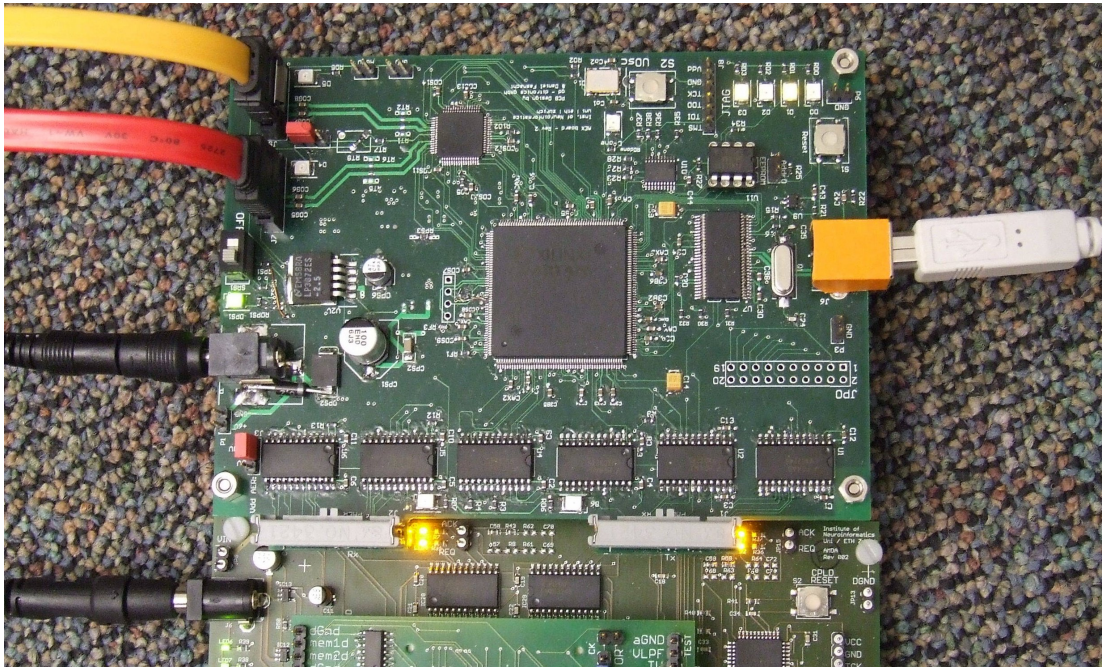
with an AMDA board



- USB protocol to interface to PCs.
- DACs and ADCs to configure chip parameters.
- Parallel AER circuits to interface to AER systems.
- Easy interchange of daughterboards

Reliable multi-chip communication

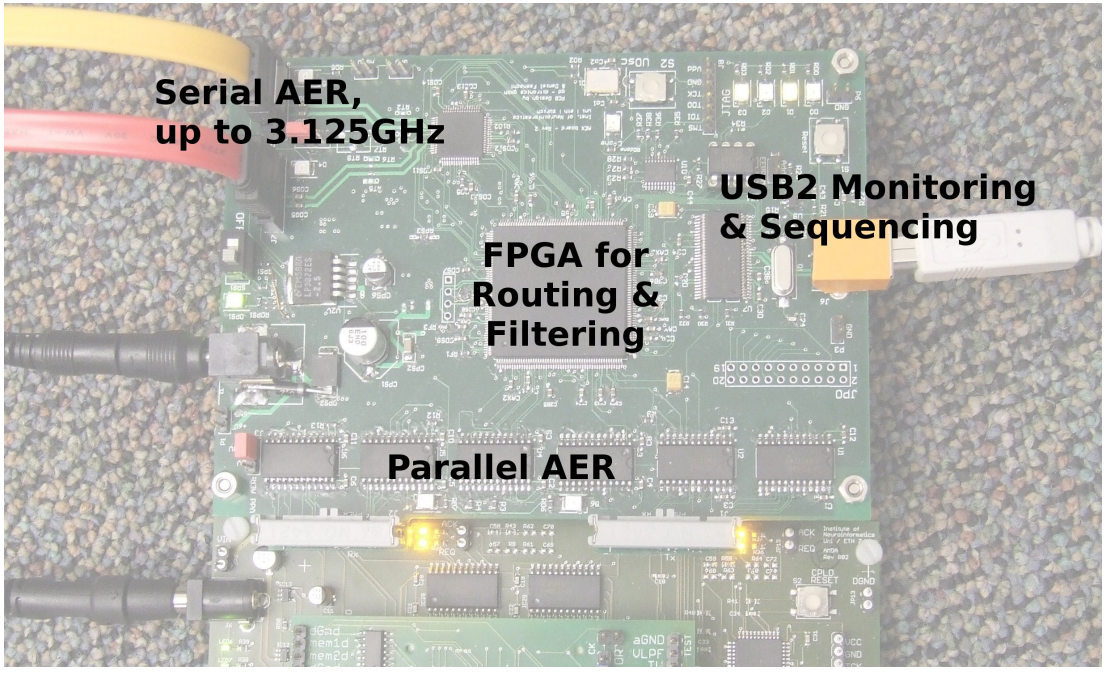
With a serial AER interface



- Use serial communication for board to board communication.
- Use industry standards (USB2, SATA).
- Fully configurable routing and filtering (via FPGA).

Reliable multi-chip communication

With a serial AER interface

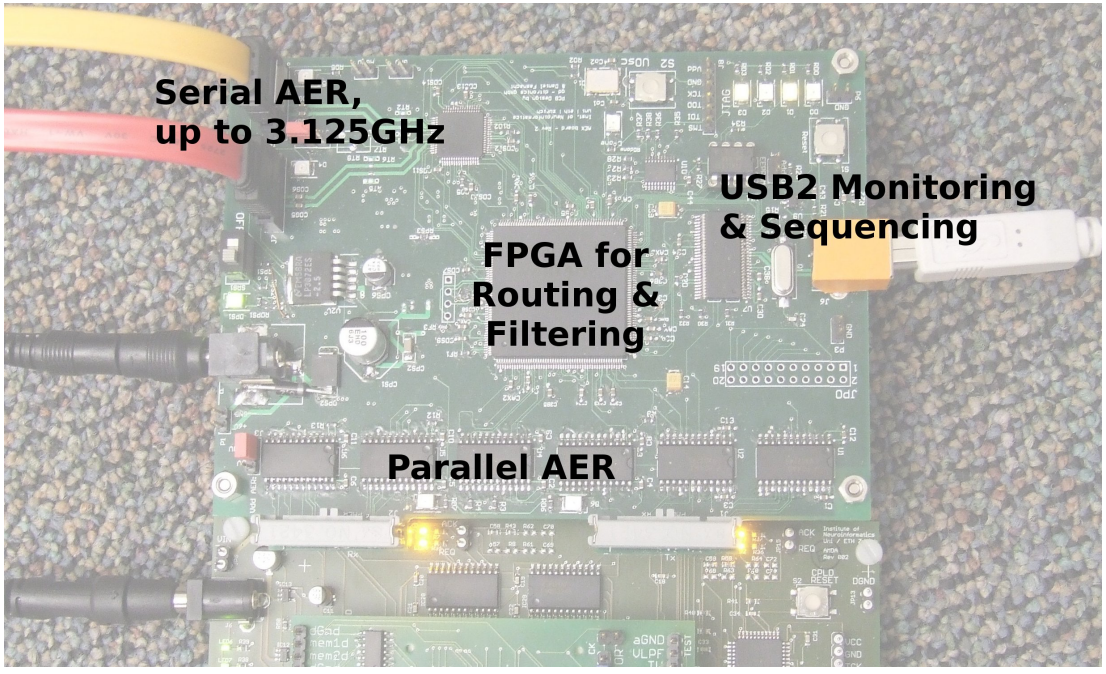


Parallel AER interface
3.3V/5V, P2P/SCX
USB2 Monitor/Sequencer
Up to 40MB/s (320Mb/s)
Serial AER
TI SerDes Chips (TLK2501/3101) Cheap SATA wiring Up to 3.125GHz / 2.5Gb/s Flow control Fully AC couppled, no GND ref
Spartan 3E FPGA
Configurable Routing AE Buffering & Filtering AE Bit-Mask Operations

- Use serial communication for board to board communication.
- Use industry standards (USB2, SATA).
- Fully configurable routing and filtering (via FPGA).

Reliable multi-chip communication

With a serial AER interface

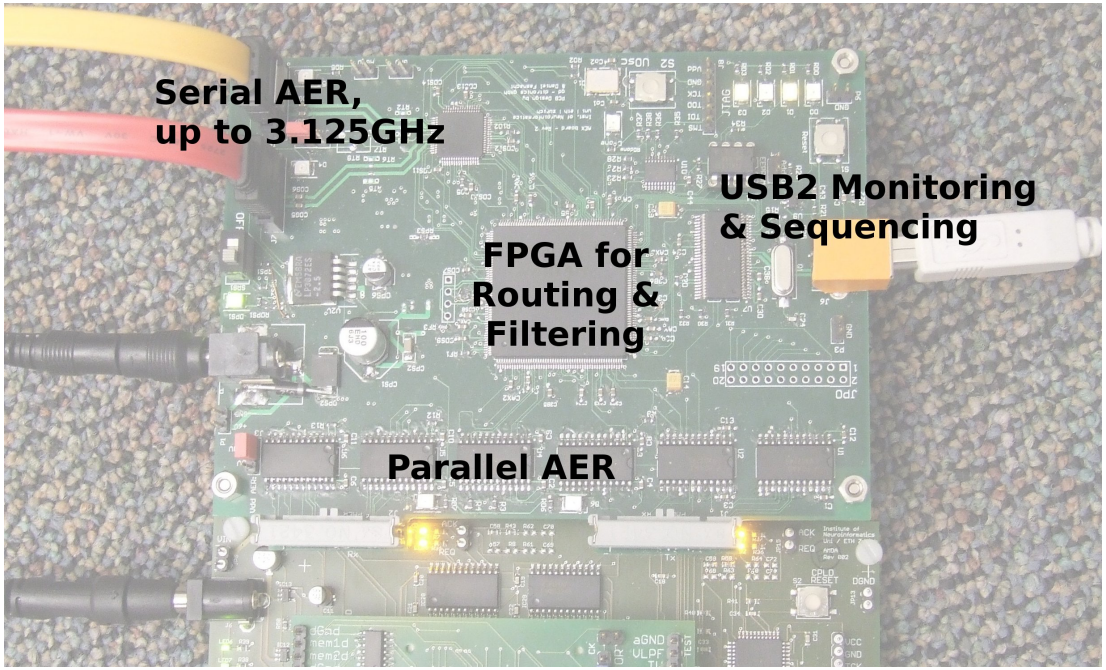


Parallel AER interface
3.3V/5V, P2P/SCX
USB2 Monitor/Sequencer
Up to 40MB/s (320Mb/s)
Serial AER
TI SerDes Chips (TLK2501/3101) Cheap SATA wiring Up to 3.125GHz / 2.5Gb/s Flow control Fully AC couppled, no GND ref
Spartan 3E FPGA
Configurable Routing AE Buffering & Filtering AE Bit-Mask Operations

- Use serial communication for board to board communication.
- Use industry standards (USB2, SATA).
- Fully configurable routing and filtering (via FPGA).

Reliable multi-chip communication

With a serial AER interface

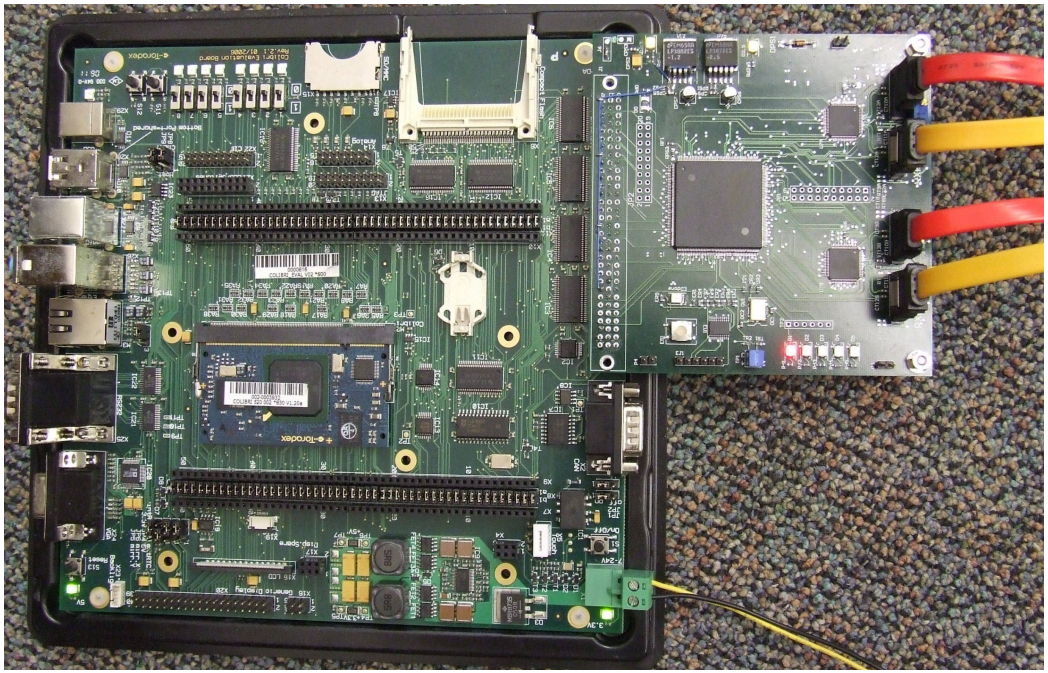


Parallel AER interface
3.3V/5V, P2P/SCX
USB2 Monitor/Sequencer
Up to 40MB/s (320Mb/s)
Serial AER
TI SerDes Chips (TLK2501/3101)
Cheap SATA wiring
Up to 3.125GHz / 2.5Gb/s
Flow control
Fully AC couppled, no GND ref
Spartan 3E FPGA
Configurable Routing
AE Buffering & Filtering
AE Bit-Mask Operations

- Use serial communication for board to board communication.
- Use industry standards (USB2, SATA).
- Fully configurable routing and filtering (via FPGA).

Reliable event processing and routing

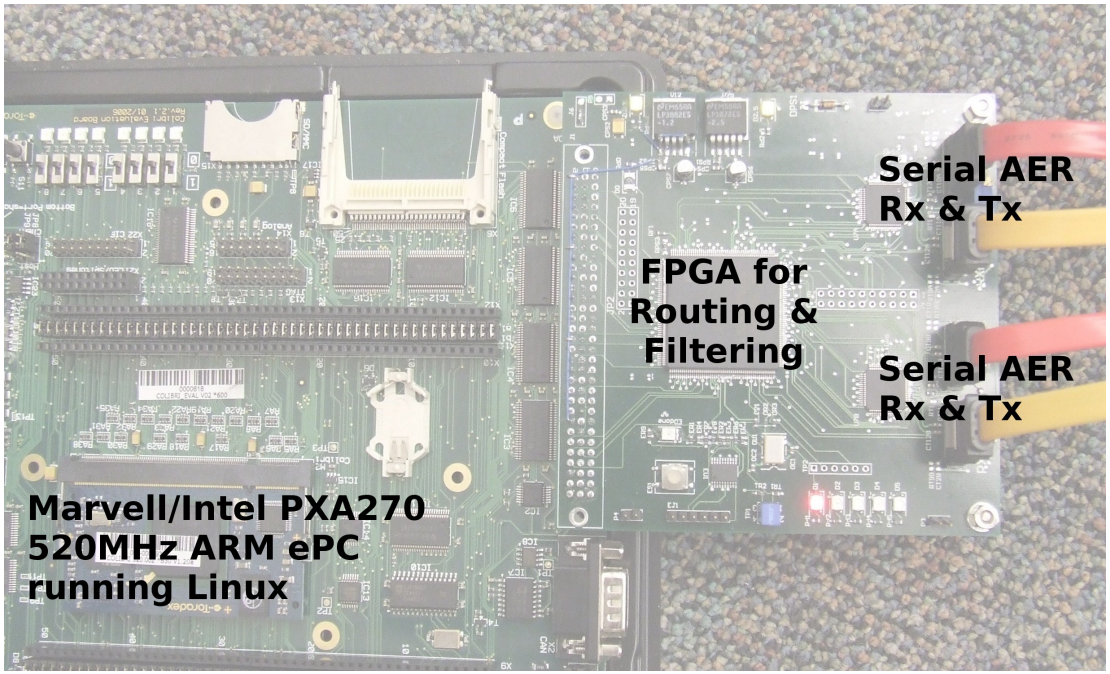
With an embedded PC platform



- Embedded PC for stand-alone operation.
- Event processing, complex (online) mapping, statistical analysis.
- Low latency ($< 500\mu s$ conservative estimate, $< 100\mu s$ typical).

Reliable event processing and routing

With an embedded PC platform

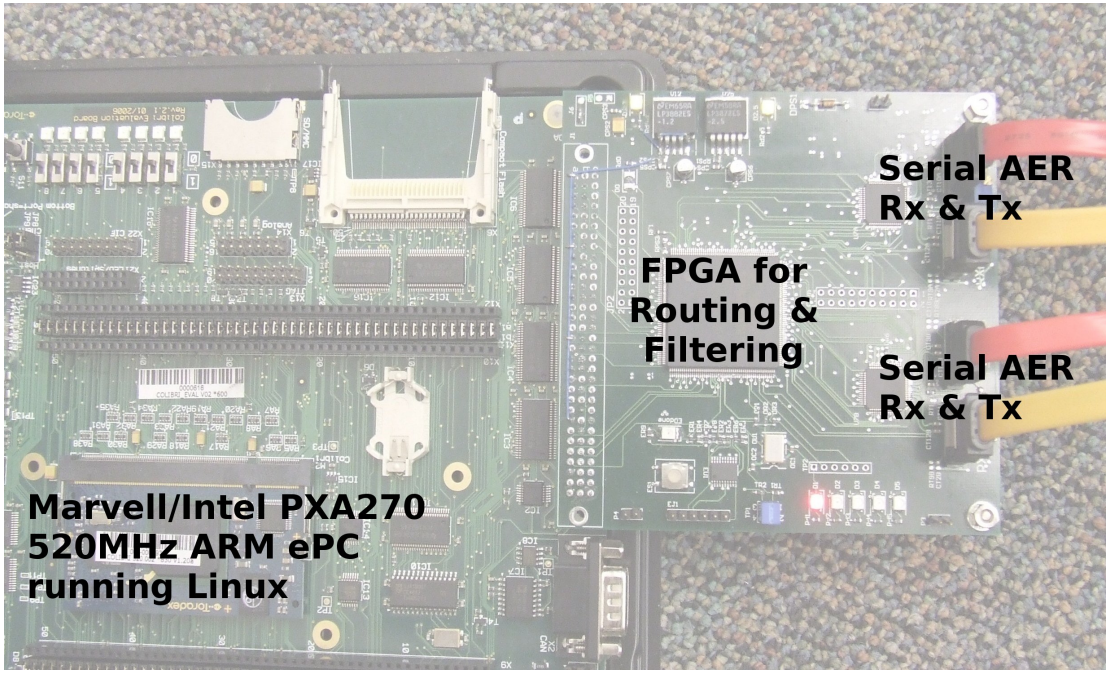


Serial AER: 2x RX, 2x TX
Spartan 3E FPGA
Interfacing AE Buffering & Filtering
Marvell/Intel PXA 270
520MHz ARM architecture 64MB RAM 32MB Flash + SD Card Running Linux

- Embedded PC for stand-alone operation.
- Event processing, complex (online) mapping, statistical analysis.
- Low latency ($< 500\mu s$ conservative estimate, $< 100\mu s$ typical).

Reliable event processing and routing

With an embedded PC platform

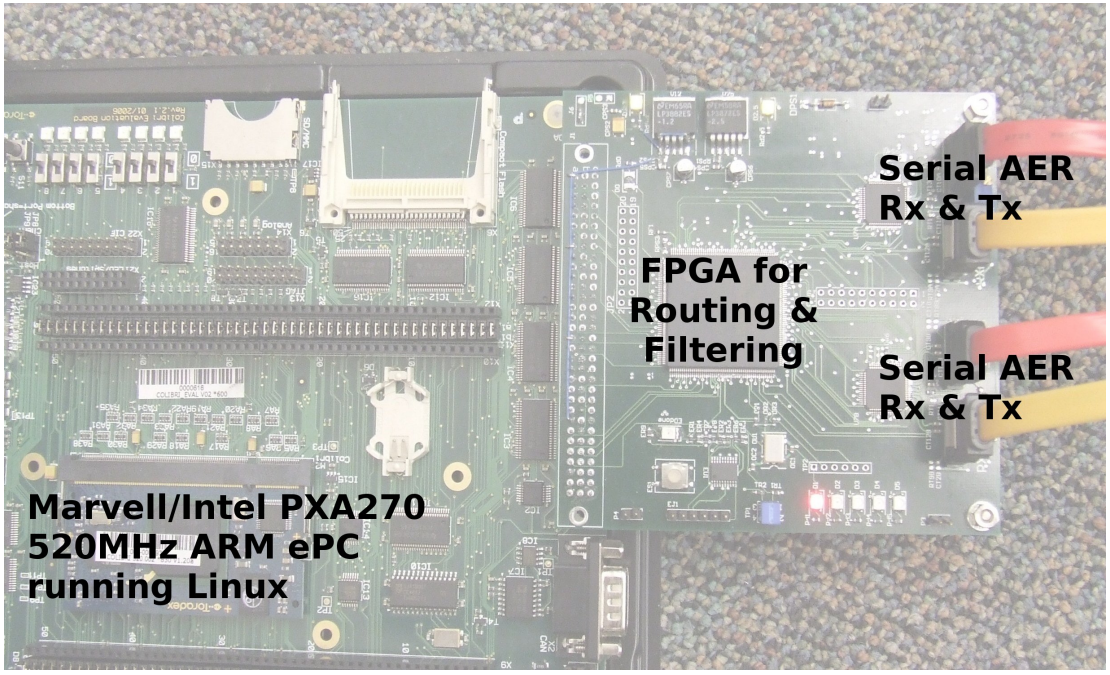


Serial AER: 2x RX, 2x TX
Spartan 3E FPGA
Interfacing AE Buffering & Filtering
Marvell/Intel PXA 270
520MHz ARM architecture 64MB RAM 32MB Flash + SD Card Running Linux

- Embedded PC for stand-alone operation.
- Event processing, complex (online) mapping, statistical analysis.
- Low latency ($< 500\mu s$ conservative estimate, $< 100\mu s$ typical).

Reliable event processing and routing

With an embedded PC platform



Serial AER: 2x RX, 2x TX
Spartan 3E FPGA
Interfacing AE Buffering & Filtering
Marvell/Intel PXA 270
520MHz ARM architecture 64MB RAM 32MB Flash + SD Card Running Linux

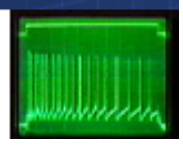
- Embedded PC for stand-alone operation.
- Event processing, complex (online) mapping, statistical analysis.
- Low latency ($< 500\mu s$ conservative estimate, $< 100\mu s$ typical).

Reliable interfaces to PCs

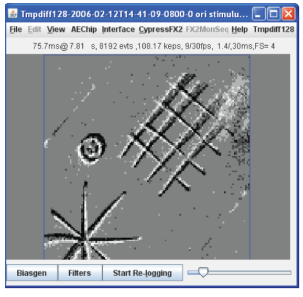
with custom firmware and software

The MathWorks
Accelerating the pace of engineering and science

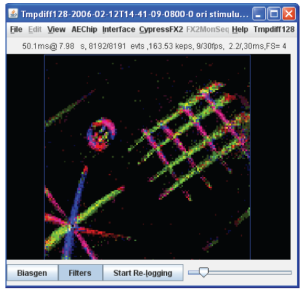
Spike ToolBox



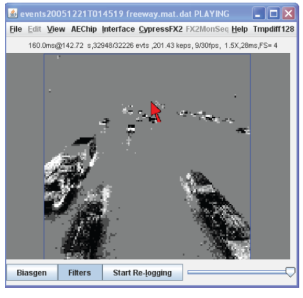
Source data



Result of orientation processing



Cars on highway



Result of object tracking

- High performance linux kernel driver.
- `libusb` platform independent library (user space) driver.
- C, Java, and Matlab user interfaces
- jAER open source project, Matlab Spike-Toolbox

Promises, promises, promises. . .

are we there yet?

- Preliminary results in all our labs
- Preliminary results in Telluride 2006
- Preliminary results in Telluride 2007?

Promises, promises, promises...

are we there yet?

- Preliminary results in all our labs
- Preliminary results in Telluride 2006
- Preliminary results in Telluride 2007?

Tough job

Going from *preliminary* results to practical, reliable and competitive applications is hard.

Promises, promises, promises. . .

are we there yet?

- Preliminary results in all our labs
- Preliminary results in Telluride 2006
- Preliminary results in Telluride 2007?

Tough job

Going from *preliminary* results to practical, reliable and competitive applications is hard.

Slow progress

The overall resources used by the community at large are less than 3 man/year on any of one of these tasks $\longrightarrow$

1. Neuromorphic VLSI chips (sensors, learning chips, analog filters, . . .).
2. Industry-strength commodity circuits (asynchronous logic, temperature compensated bias generators, . . .).
3. Printed Circuit Boards (PCBs) for hosting the neuromorphic chips.
4. On-board communication logic.
5. Off-board communication interfaces.
6. Board-PC interfaces.
7. Low-level PC drivers.
8. Software user-interfaces, Matlab toolboxes, *etc.*
9. . . .

Thank you for your attention

Acknowledgments

The Institute of Neuroinformatics, UNI-ETH Zurich

Chiara Bartolozzi
Srinjoy Mitra
Sung-do Choi
Daniel Fasnacht

Vittorio Dante
Adrian Whatley
Gerd Dietrich
Dylan Muir

Elisabetta Chicca
Stefano Fusi

Rodney Douglas
Kevan Martin

ALAVLSI (IST-2001-38099) EU grant
DAISY (FP6-2005-015803) EU grant
ETH TH0-20174-04 grant.

Serial AER

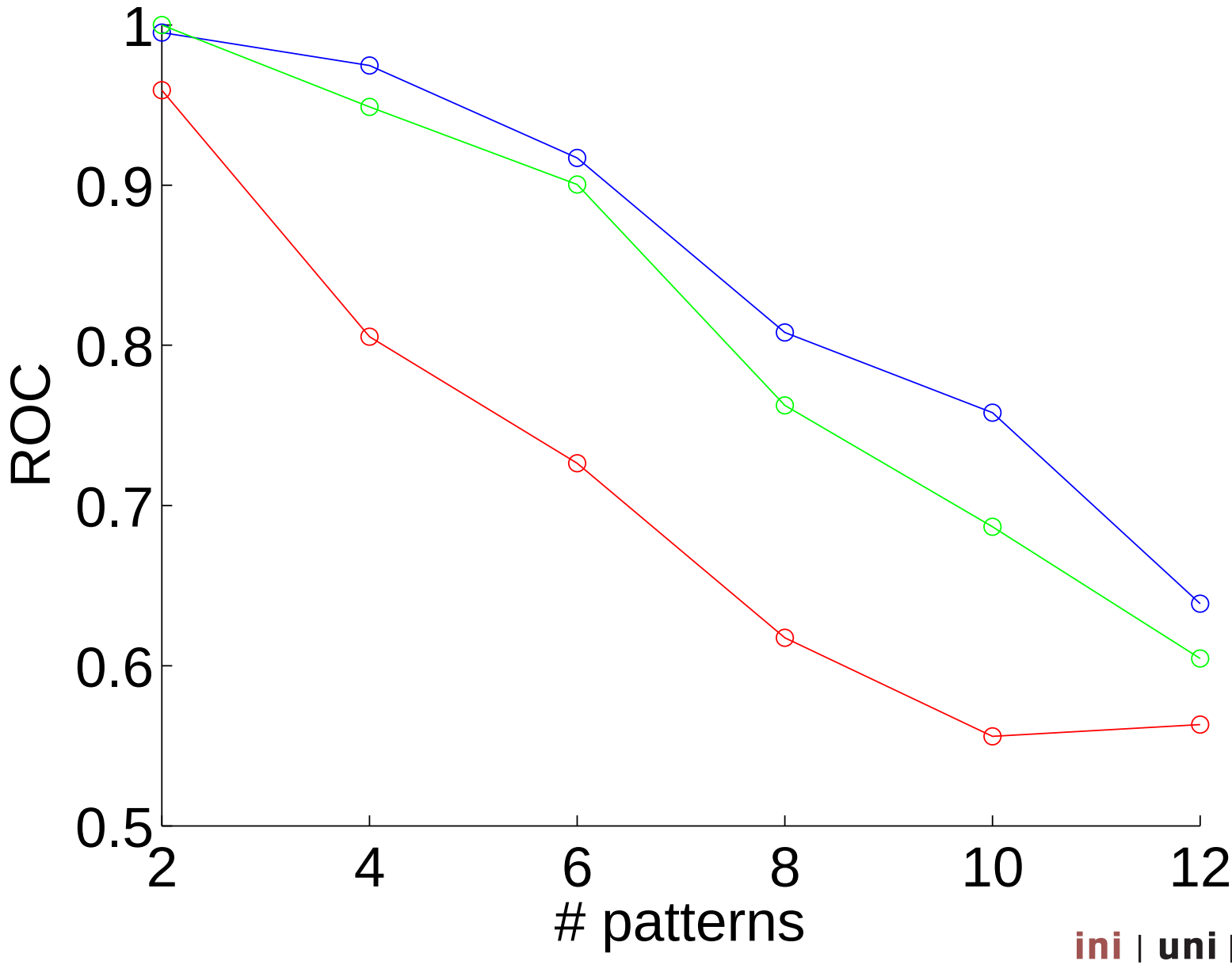
Disadvantages

- Extra chip (SerDes) or
- IP-core on expensive (<=130nm) processes
- Maybe higher communication power consumption
- 4+ layer PCB required for impedance control

Advantages

- Cheap wiring & connectors (i.e. SATA)
- Differential signaling, thus:
  - Common mode noise immunity (RF noise)
  - No net current flow
- Redundant encoding (8b/10b)
  - Clock recovery
  - Error detection
  - Extra tokens (i.e. IDLE, error)
  - DC-free
- Full AC coupling
  - No global GND reference
  - No ground bounce problems

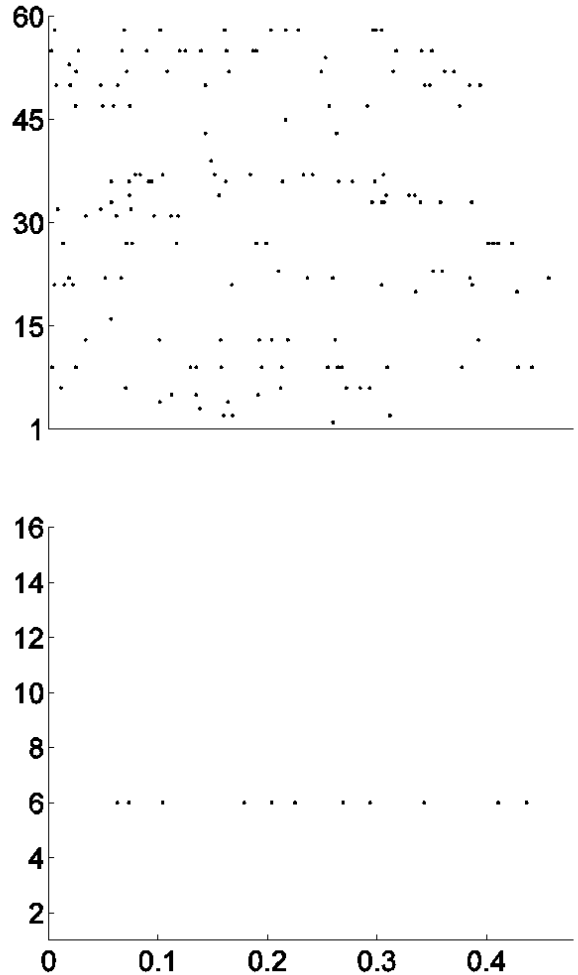
Receiver Operating Characteristic (ROC)



Random “graded” patterns

Class-B neuron

Class A



Class B