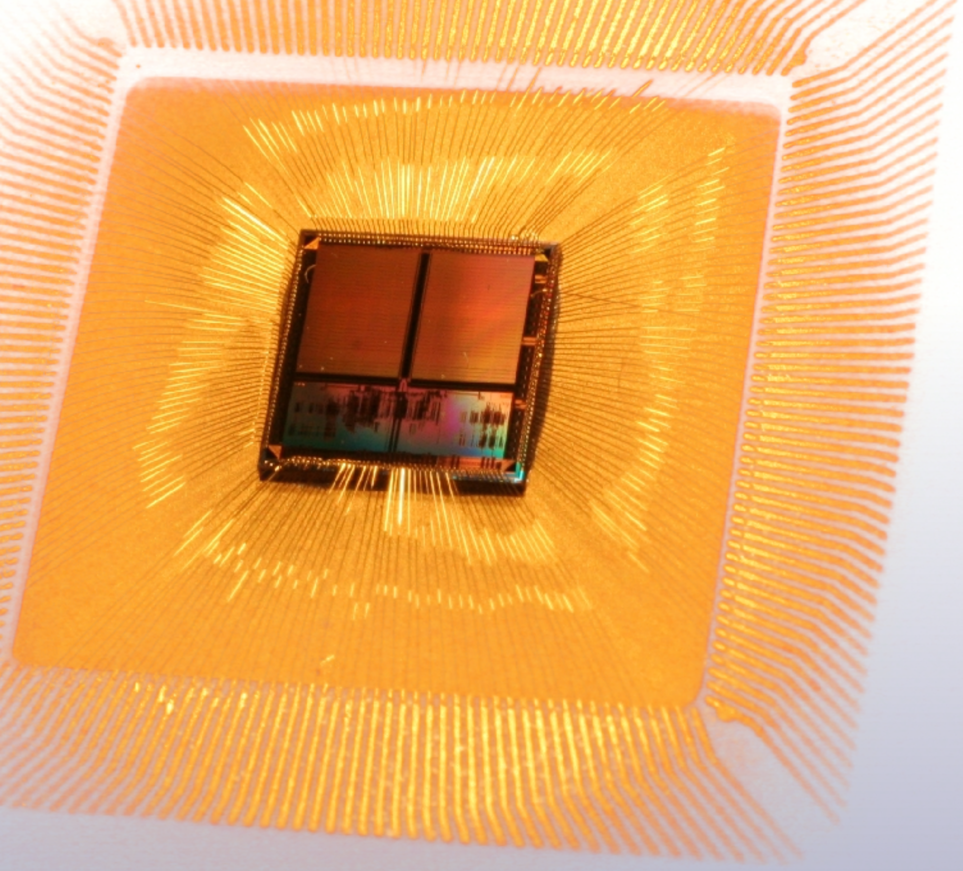


PyNN and the FACETS Hardware

Daniel Brüderle
Heidelberg

Electronic Vision(s)

FACETS Hardware: Recap

„Neuromorphic“ Hardware: A physical model, not a simulation

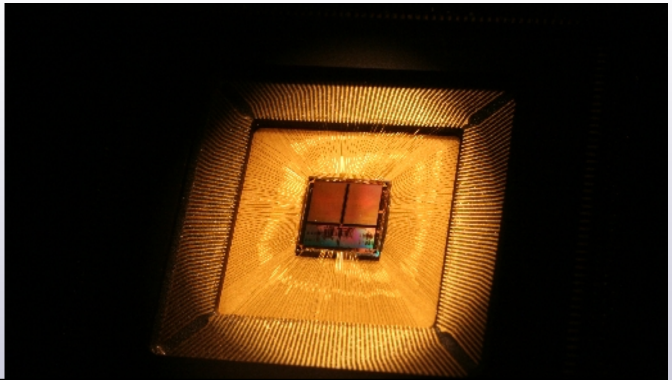
- Intrinsically parallel, scalable, fast, ...
- Not an arbitrarily flexible substrate
  - fixed neuron model
  - limited ranges for neuron and synapse configuration parameters
  - limited resources
    - neuron number
    - connectivity / synapse number
    - max. firing rates
    - individual configurability

FACETS Hardware: Recap

- Three FACETS groups design and build neuromorphic hardware
  - Bordeaux: High-precision systems
    - mixed-signal VLSI HH model
    - real-time
    - ~ 10^0 - 10^2 neurons
  - Heidelberg / Dresden: Large-scale accelerated system
    - mixed-signal VLSI I&F model
    - highly accelerated (speedup factor ~ 10^4 - 10^5)
    - ~ 10^2 – 10^6 neurons
    - 2 stages of development...

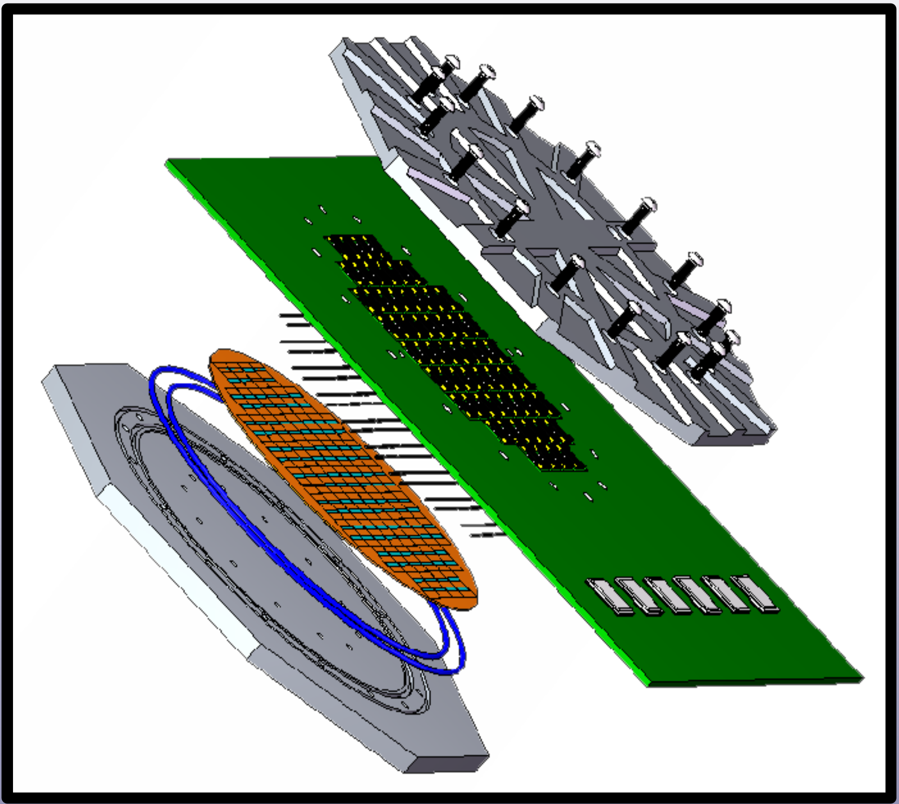
Accelerated FACETS Hardware

- Stage1 (chip-based):
  Conductance-based I&F neurons
  - 384 interconnectable neurons on each chip
  - programmable connectivity
    - source, target, weight, tau_syn
  - chips interconnectable
  - STDP (analog, on-chip)
  - short term dep / fac
  - no spike-frequency adaptation



Accelerated FACETS Hardware

- Stage 2 (waferscale integration):
  Adaptive EIF a la Brette & Gerstner
  - ~ 10^5 dendritic building blocks and ~ 10^7 synapses per wafer
  - wafers interconnectable
  - STDP (digital, on-chip)
  - short term dep / fac



Status Hardware (May 2008)

- Stage 1: up and running
- Problems with analog parameter storage
  - temperature dependent leakage currents at synapses, also at voltage and current memory
  - hard to control e.g. STDP
  - hard to quantitatively compare results to e.g. NEST
- Only subset of neurons readable at the same time
- New, bug-fixed chip available since 1st of May
  - stable parameters
  - all neurons recordable at the same time

Status Hardware (May 2008)

- Stage 2: final stage of development
  - neuron and synapse model decided, prototype for parts of the model available (Stage 1 chip)
  - connectivity and routing issues decided, methods for network mapping existing, under further development and testing
  - new analog floating gate memory developed and successfully tested
  - wafer post-processing successfully tested
  - prototype for digital long-distance and off-wafer communication available
  - first full system expected during 2009

Why PyNN for the FACETS Hardware?

- Only little neuroscientific expertise in hardware groups
- Plan: Hardware as a useful research tool for modelers' community
  - statistics-intensive, large parameters sweeps, long-term learning, etc
  - interweaved hardware software co-simulation
- Needed: Access and usability for every FACETS member

Why PyNN for the FACETS Hardware?

- Python and PyNN provide
  - easy-to-learn, well documented user interface for non-hardware-specialists
  - experiment porting
  - quantitative result comparisons
    - e.g. for hardware model verification
  - analysis and post-processing tools
- Plans to adopt PyNN also for the Bordeaux hardware system and by e.g. Giacomo Indivery (DAISY)

Status *PyNN.hardware*

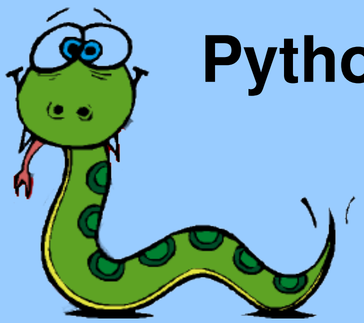
- Started with basic interface: Very hardware-specific C++ API
- At CodeJam #1: plain Python interface (boost), no connection to PyNN
- Now: PyNN supported as far as possible („*pyNN.hardware.stage1*“)
  - procedural API
    - hardware well hidden – seems to behave like e.g. NEST, just faster ;)
    - reasonable default values for hardware parameters
    - voltage recording via oscilloscope + c++-sockets + boost.python
  - after Code Sprint in Debrecen: Populations / Projections
  - standard output formats
  - neuron model IF_facets_hardware1
  - warnings / errors for constraints

PyNN

procedural and object-oriented API supported
own neuron model

PyHAL

Hardware abstraction layer
Object-oriented, user friendly, full chip functionality



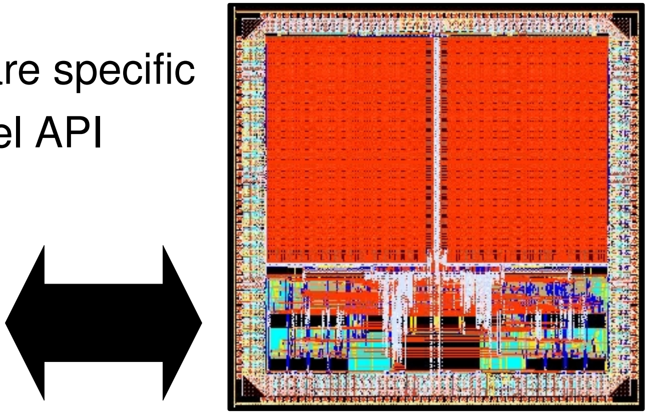
Python

Boost.Python
1:1 translation of all relevant classes

C++

Hardware specific
low-level API

config
input
output



Status *PyNN.hardware*

- Drawbacks:
  - no voltage recording of all neurons at the same time
  - limited parameter ranges (weights, voltages, time constants...), hardly handled so far
  - every run is different...
    - leakage
    - temperature
    - crosstalk
    - power supply
    - ...

Hardware Specific Implementations

- Temporal resolution („timestep“): Sampling rate oscilloscope
- Additional parameters
  - work station (chip) selection
  - translation factors
    - weights
    - temporal speedup
  - mapping parameters
  - calibration data (files for every workstation)
- Unused parameters
  - min_delay, max_delay

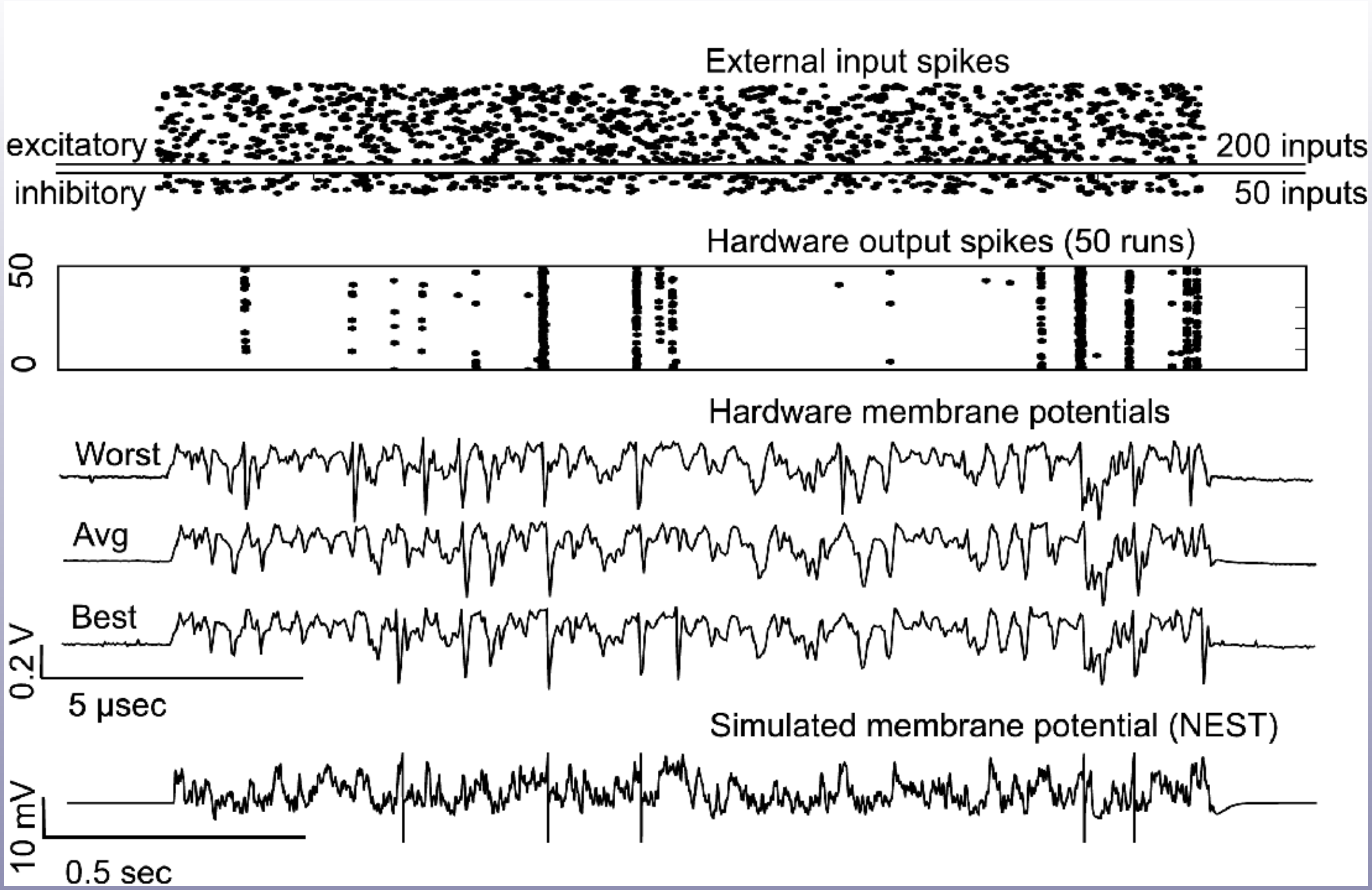
PyNN.hardware in the Official Trunk?

- Plans as decided in Debrecen:
  Provide everyone with a lightweight dummy PyNN.hardware module
  - full PyNN.hardware module necessary only in Heidelberg
  - dummy module implements all errors and warnings that arise due to hardware specific constraints
  - for offline testing of scripts
  - run routine returns only „script executable“ or „script not executable“

Further Plans

- Include Graph Model (for mapping networks to the hardware configuration space, see lightning talk by Johannes Bill)
- Clean handling of limited parameter ranges
- Memory management for large numbers of experiments
  - Direct correspondence / mapping from high level data structures to allocated experiment objects in hardware playback memory

Hardware Model Verification



presented at the IWANN 2007