

# A Wafer-Scale Neuromorphic Hardware System for Large-Scale Neural Modeling

Johannes Schemmel, Daniel Brüderle, Andreas Grübl, Matthias Hock, Karlheinz Meier and Sebastian Millner  
Kirchhoff Institute for Physics, University of Heidelberg  
Im Neuenheimer Feld 227, 69120 Heidelberg, Germany  
Email: schemmel@kip.uni-heidelberg.de

**Abstract**— Modeling neural tissue is an important tool to investigate biological neural networks. Until recently, most of this modeling has been done using numerical methods. In the European research project "FACETS" this computational approach is complemented by different kinds of neuromorphic systems. A special emphasis lies in the usability of these systems for neuroscience. To accomplish this goal an integrated software/hardware framework has been developed which is centered around a unified neural system description language, called PyNN, that allows the scientist to describe a model and execute it in a transparent fashion on either a neuromorphic hardware system or a numerical simulator. A very large analog neuromorphic hardware system developed within FACETS is able to use complex neural models as well as realistic network topologies, i.e. it can realize more than 10000 synapses per neuron, to allow the direct execution of models which previously could have been simulated numerically only.

## I. INTRODUCTION

Artificial neural systems play an important role in neuroscience. The main reason being the limited access one has to the individual neurons and synapses in vivo. By gathering the knowledge obtained from biological experiments and using them to build an artificial neural system [1], computational neuroscience has emerged as an indispensable part of brain research. In addition, artificial neural systems will enable a lot of new applications in the areas of robotics, ambient intelligence and human-machine interfaces, to name a few of the most prominent examples.

The basis of an artificial neural system is the model. The model captures a certain state of experimental data and also the modeler's beliefs about the biological domain in the language of mathematics. Usually, the equations derived in the modeling process are subsequently numerically solved on a parallel computer. Thus, the artificial neural system derived in this way is a computer simulation.

Regarding the biological basis of neural communication it has been shown [2] that neural systems are well suited to physical modeling as well. In a physical model, the constituents of the system are modeled by physical entities which obey the same equations as the mathematical description of the system. The most successful technology to implement such a physical model is VLSI technology. The physical entities are small transistor circuits, their interconnection uses the available metal wiring as well as programmable switches to control the topology.

Using a physical model keeps a one-to-one relationship between the neurons and synapses of the biological example and the model. Thereby, the fault tolerance concerning the loss of individual neurons and synapses observed in biology is preserved. This is an especially useful property considering the reliability concerns of future CMOS generations [3]. In addition, by using only a few transistors to emulate the neuron's differential equations compared to several millions involved in the same task while solving these equations numerically on a microprocessor core, the power consumption is reduced by several orders of magnitude [4]. Due to the inherent continuous time operation of a physical

model it is much faster than the numerical approach for all but the most simple network configurations. On the other hand, a physical model is much less flexible compared to a generic software module. Taking this into account, numerical and physical models could complement each other in neuroscience research.

Considering future applications of the neural computing paradigm a sophisticated physical model is much more likely to satisfy the power and cost demands, although it is not yet clear if contemporary VLSI technology can reach the necessary complexity.

The FACETS project aims at developing a large-scale physical model capable of implementing most of the neural systems modeled in contemporary computational neuroscience. The following sections will give an overview of the different components of this model and its associated VLSI circuits as well as its system and software components.

## II. NEURON MODEL

At the basis of a neural VLSI implementation is the neuron model itself, i.e. the differential equation governing the temporal evolution of the membrane potential. The FACETS project participated in the development of a new model, the exponential integrate and fire model (AdExp) [5] which contains several additions compared to the standard integrate and fire model (I&F) [6]:

$$-C_m \frac{dV}{dt} = g_l(V - E_l) - g_l \Delta_{th} \exp\left(\frac{V - V_{th}}{\Delta_{th}}\right) + g_e(t)(V - E_e) + g_i(t)(V - E_i) + w(t) \quad (1)$$

The variables  $C_m$ ,  $g_l$ ,  $E_l$ ,  $E_e$  and  $E_i$  are the membrane capacity, the leakage conductance and the leakage, excitatory and inhibitory reversal potentials. The variables  $g_e(t)$  and  $g_i(t)$  represent the total excitatory and inhibitory synaptic conductances. The *exponential* term on the right hand side introduces a new mechanism to the I&F neuron: Under certain conditions, the membrane potential develops rapidly towards infinity. The *threshold potential*  $V_{th}$  represents the critical value above which this process can occur, and the *slope factor*  $\Delta_{th}$  determines the rapidness of the triggered growth. Such a situation, which is detected by a mechanism that permanently compares  $V(t)$  with a critical value  $V_{spike} > V_{th}$ , is interpreted as the emergence of a spike. Each time a spike is detected, a separately generated output event signal is transmitted to possibly connected target neurons (or to recording devices), and the membrane potential is forced to a reset potential  $V_{reset}$  by an adjustable, usually very strong, reset conductance.

A second equation describes the temporal evolution of the so-called *adaptation current*  $w(t)$ :

$$-\tau_w \frac{dw}{dt} = w(t) - a(V - E_l) \quad (2)$$

Additionally, every time a spike is emitted by the neuron,  $w$  changes its value quasi-instantaneously:  $w \rightarrow w + b$ . The time constant and the efficacy of the so-called *sub-threshold* adaptation mechanism are given by  $\tau_w$  and  $a$ , while  $b$  defines the amount of the so-called *spike-triggered* adaptation.

This work is supported in part by the European Union under the grant no. IST-2005-15879 (FACETS).

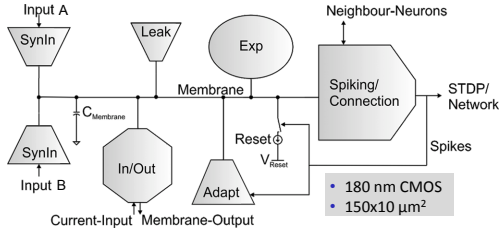


Fig. 1. Schematic diagram of the AdExp neuron circuit.

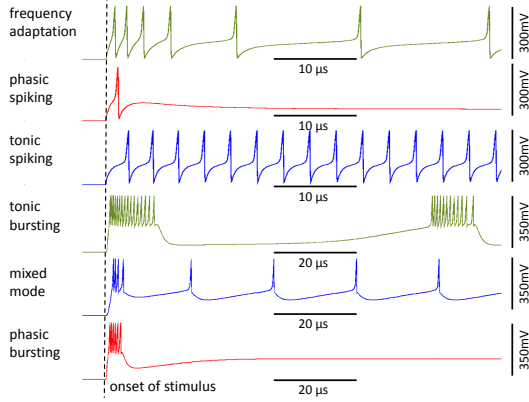


Fig. 2. Exemplary firing modes of the AdExp neuron circuit.

Since both the exponential term in Equation 1 and the adaptation can be deactivated, the AdExp-model can be reduced to the standard I&F model.

The AdExp-model was implemented in a 180 nm CMOS technology. Fig. 1 shows the individual circuit components. Fig. 2 illustrates exemplary firing modes of this neuron circuit. The timescale of the transient simulations shown in the figure reveals one important aspect of the FACETS neuron model: it operates at accelerated biological time. The acceleration factor ranges from  $10^3$  up to  $10^5$  compared to the biological real time (BRT). The neuron time constant is determined by the membrane capacitance and the leakage conductance. Operating the VLSI model at an accelerated time scale allows a reduction of the internal capacitances and an increase of the leakage current compared to the biological model. Thereby the size of the circuits can be decreased substantially. The membrane capacitance is implemented as a MIM-capacitor sitting on top of the circuit, thus occupying no additional silicon area. Most of the internal currents stay in the range from 100 nA to 1  $\mu$ A which avoids deep sub-threshold operation and the strong fixed-pattern noise associated with it.

### III. NEURON INTEGRATION

The neuron circuits are integrated together with their respective synapses in a structure called the Analog Network Core (ANC) as depicted in Fig. 3. To allow neurons with a variable number of synapses, a neuron is built from multiple parts, named dendrite membrane (*DenMem*) circuits. Each *DenMem* circuit is connected to  $224^1$  synapses. In the middle of the ANC is the *neuron builder*. It combines groups of *DenMem* circuits to neurons. It is freely programmable as long as the *DenMem* circuits forming a neuron are adjacent or opposite, regarding the left and right half of the ANC, to each other. Each *DenMem* circuit is configured by 23 individual analog parameter inputs which are generated by

<sup>1</sup>This number is defined by the MPW-size limits of the manufacturer.

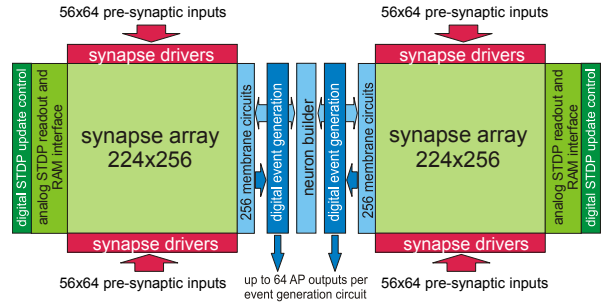


Fig. 3. Schematic diagram of the Analog Network Core (ANC).

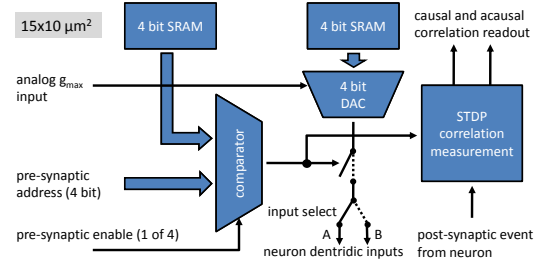


Fig. 4. Schematic diagram of a synapse.

single-poly floating-gate analog memory cells located between the *DenMem* circuits and the neuron builder.

Fig. 4 shows the structure of a synapse. Each synapse contains a four bit address decoder and is connected to one out of four enable signals. Therefore, the synapses located in a certain column can receive as many as 64 different pre-synaptic inputs, but two adjacent columns share the same 64 inputs. The maximum number of pre-synaptic inputs a neuron can receive is therefore  $columns/2 \times 64 \times blocks = 14336$ , with the number of synapse columns = 224 and the number of synapse blocks = 2.

The synapse weight is represented by a current generated by a four bit multiplying DAC. A four bit SRAM stores the individual synapse's weight while the maximum conductance for a column of synapses is controlled by the analog input  $g_{max}$ . The  $g_{max}$  values of two adjacent columns sharing the pre-synaptic input can be programmed to be a fixed multiple of each other. Thereby these columns can combine their synapses to reach a weight resolution of 6 to 8 bits at the expense of a twofold reduction of the synapse number in these columns.

The synapses transmit their post-synaptic messages to the neuron using two shared inputs per *DenMem* circuit. Each column of synapses connects either to input A or B (see Fig.4). The signal is encoded as a current pulse of defined length and an amplitude proportional to the synapse weight. The synaptic input circuit in the neuron clamps the line to a fixed potential and uses the integral of the synaptic currents it receives to generate an exponentially decaying synaptic conductance, which is subsequently used to charge or discharge the membrane, depending on the selected reversal potential. In a simple setup input A could for example be excitatory and input B inhibitory, with different reversal potentials and synaptic time constants. A more elaborate setup with larger neurons built from several *DenMem* circuits may use different kinds of membrane ion channels. This can be emulated by setting up different reversal potentials and synaptic time constants for different *DenMem* circuits.

There are two levels of plasticity implemented in the synapse block: short-term depression and facilitation and spike-timing de-

pendent plasticity (STDP). The former effect is generated similar to [7]. For short-term plasticity the effective weight is modulated by the length of the current pulse. Since there are 64 possible pre-synaptic neurons connecting to a synapse column the plasticity circuit located in the periphery of the synapse block keeps track of the firing history of all 64 channels by means of a capacitive storage array.

STDP uses a two stage algorithm: first, in each synapse the temporal correlation between pre- and post-synaptic signal is measured and the exponentially weighted results of this measurement are accumulated on two capacitances, one for causal and one for acausal correlations. A digital control circuit periodically reads these data and decides utilizing a programmable algorithm if the digitally stored weight should be changed. In case the weight is changed the capacitive storage is cleared and the accumulation process starts again. This is similar to the implementation described in [8].

#### IV. INTEGRATION OF THE ANALOG NETWORK CORE

The ANCs presented in the previous section are the building blocks for the FACETS artificial neural system. To reach a sufficient complexity thousands of these blocks have to be interconnected by a network with sufficient bandwidth and an adequately flexible topology. The following section will give a short summary about the solutions developed within the FACETS project. A more detailed description can be found in [9].

##### A. Event Communication

Operating an ANC at an acceleration factor of  $10^4$  and a mean firing rate of 10 Hz BRT leads to a pre-synaptic event rate of 1.5 Gigaevent/s. This poses two problems: providing sufficient bandwidth in-between the ANCs and limiting the power consumption per event. An asynchronous, serial event protocol operating at up to 2 Gb/s is used to interconnect the ANCs, called Layer 1 (L1) routing. A single L1 event transmits six bits encoding the pre-synaptic neuron number. To limit the power consumption, low-voltage differential signaling is used everywhere outside of the ANCs. Each event is framed by a start- and a stop-bit. Their temporal distance is used as a timing reference for a DLL in each receiver circuit. Therefore, there is no activity outside of an event frame limiting the quiescent power consumption of the receivers. The only part that needs a continuous bias current of about  $100 \mu\text{A}$  is the differential input stage, since it has to detect the start-bit in time which lasts only one bit-period. The signal deteriorates due to the RC-time constant of the on-chip transmission lines and therefore it has to be repeated in regular intervals. To avoid accumulating timing errors active repeater circuits using a DLL for timing recovery have been inserted at each ANC boundary. Fig. 5 shows an event frame (insert) measured after it has traveled a distance of 10 mm across a prototype chip containing said repeaters. The outer part of the figure plots the power consumption as a function of the event rate.

##### B. Wafer-Scale Integration

Each L1 channel carries the pre-synaptic signals from 64 neurons, thus at most 224 channels are needed to feed the ANC. At the nominal acceleration factor of  $10^4$  this allows event rates of about 300 Hz BRT per pre-synaptic neuron, providing ample head-room. 224 differential signal lines need 448 physical wires. If the ANCs are arranged in a two-dimensional grid with L1 channels running horizontally and vertically through the ANCs, each ANC would need at least twice as many connections. A Wafer-Scale integration scheme was selected to implement these channel densities. Fig. 6 shows the resulting structure: Eight individual chips, named HICANN (High Input Count Analog

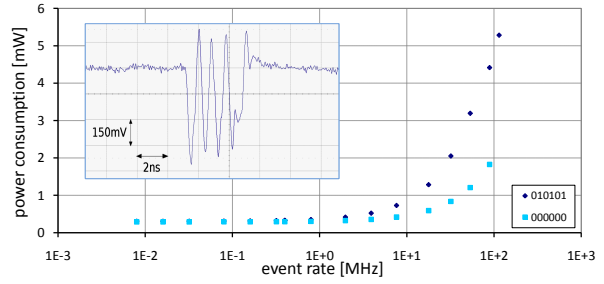


Fig. 5. Event-rate dependent power consumption of the Layer 1 repeater circuits. Insert: Measured Layer 1 event data packet after traveling 10 mm on chip.

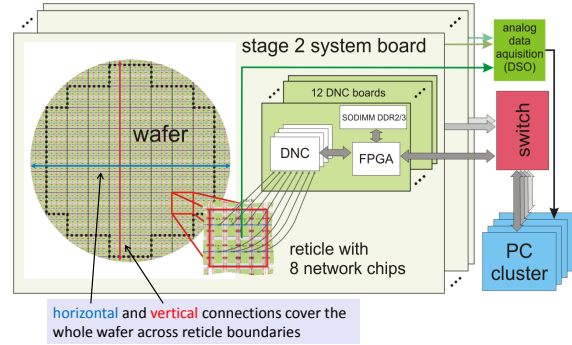


Fig. 6. Overview of the FACETS wafer-scale system

Neural Network), containing one ANC each together with the L1 repeaters as well as the necessary support circuitry, constitute one reticle. This fine granularity allows the production of prototypes by MPW runs. Fig. 7 shows the first prototype of HICANN. 44 reticles containing 352 HICANN chips fit on a 20 cm wafer. Within the reticle, horizontal and vertical L1 channels connect the HICANN chips. An additional metal layer is deposited on top of the wafer by a post-processing step, which allows the interconnection of the individual reticles with a metal pitch well below  $10 \mu\text{m}$ . Since within the reticle this metal layer is not needed for L1 connections, it is used to redistribute the bondpads of the HICANN chips into rows of equally spaced rectangular  $200 \times 1000 \mu\text{m}^2$  pads. These pads connect the wafer to the system board by an array of elastomeric strip connectors. Fig. 8 shows photographs of the inter-reticle connections formed by the wafer post-processing step<sup>2</sup>.

<sup>2</sup>The fan-out structures visible are necessary because the size of the pad-windows is larger than the line pitch.

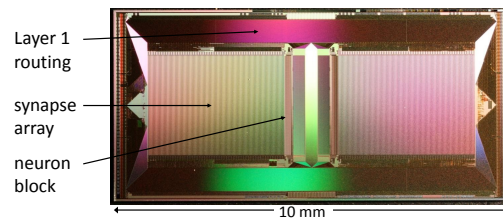


Fig. 7. Photograph of the HICANN die.

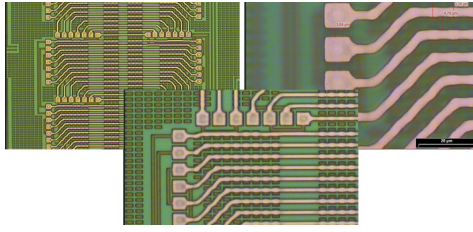


Fig. 8. Photograph of the inter-reticle connections (8  $\mu\text{m}$  pitch).

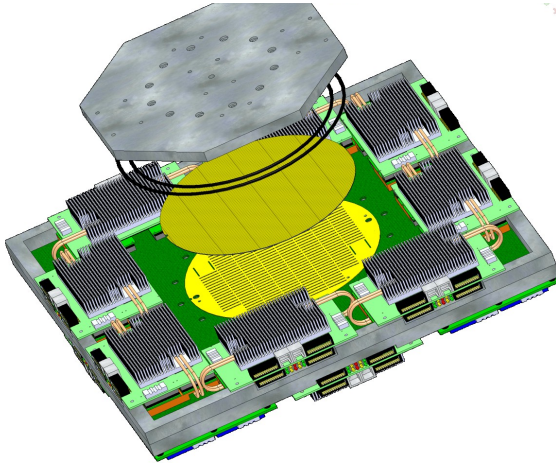


Fig. 9. Drawing of a complete wafer module.

### C. Inter-Wafer connections

A single wafer contains  $4 \cdot 10^7$  synapses and up to 180k neurons. Larger systems can be built by interconnecting several wafer modules. For this purpose a second communication protocol is implemented into the HICANN chips, the Layer 2 routing. Instead of the continuous time L1 protocol, it uses time-stamps to code the precise firing-time of the neuron. A full-duplex 2 Gb/s connection links each HICANN to the system PCB, resulting in a total bandwidth of 176 GB/s for the whole wafer. Since a data packet uses 32 bit, 44 Gigaevents/s can be exchanged between two wafers. The packet-based protocol is handled on the system PCB by a set of specialized ASICs<sup>3</sup> containing dedicated buffer algorithms to sort all arriving packets by their time-stamps. Thereby, the maximum jitter of an L2 connection is 4 ns. The wafer-to-wafer communication is handled by standard FPGAs and OTS switches via 1 or 10 Gbit Ethernet links. Fig. 9 shows the arrangement of the wafer and its surrounding digital communication hardware. The host communication uses specialized packets with lower priority sharing the L2 connections. Due to the standard Ethernet protocol used throughout the wafer-to-wafer communication any amount of computing power necessary to control the system can be incorporated into the network.

### V. SOFTWARE MODEL

The hardware model described in the previous sections allows to perform modeling experiments which previously have been possible only as numerical simulations on supercomputers. But if these experiments should be performed by computational neuroscientists the usage of such a machine has to be similar to the way these experiments are set up on conventional computers. To reach this goal a new software interface based on the script

<sup>3</sup>The Layer 2 circuits have been developed by the TU Dresden.

language Python [10] was introduced by the FACETS project into the neuroscience community, called PyNN [11]. PyNN allows to describe the experimental paradigm, the used models and the evaluation of the experiment's outcome without referring to a specific simulation engine. Therefore, a PyNN setup can be transferred to the hardware model as well. From an internal graph-based representation of the biological network the configuration data has to be calculated similar to the place and route process used for mapping an HDL design to an FPGA. All external stimuli are then sent in real time via the L2 communication links to the hardware neurons while simultaneously the firing pulses of those neurons selected for recording are read back for analysis.

### VI. CONCLUSION

The FACETS project has developed solutions to enable large scale analog hardware to complement numerical simulation as modeling tools for neuroscience. It has successfully addressed the following prerequisites for this endeavor:

- programmability of topology and model parameters,
- a flexible and biologically realistic neuron model,
- a low-power communication technology for neural events,
- a scalable packet-based inter-wafer and host communication which includes the possibility of interactive simulations to close the sensor-actor loop,
- a software framework for the translation of experiments from biology to hardware including a serious attempt at the unification of the electronic specification of such experiments through the PyNN initiative.

What lies ahead of the involved groups now is the consolidation and finalization of the numerous developments made. Concerning the hardware, the first HICANN prototype is in the lab for measurements now while the remaining parts of the system have been tested by numerous prototypes. On the software side, there has been the successful transfer of exemplary neuroscience experiments to the hardware. A behavioral simulation of the wafer-scale system allows to execute them in simulation.

### REFERENCES

- [1] C. Johansson and A. Lansner, "Towards cortex sized artificial neural systems," *Neural Networks*, vol. 20, no. 1, pp. 48–61, 2007.
- [2] C. A. Mead and M. A. Mahowald, "A silicon model of early visual processing," *Neural Networks*, vol. 1, no. 1, pp. 91–97, 1988.
- [3] ITRS, "International technology roadmap for semiconductors," [http://www.itrs.net/Links/2007ITRS/2007\\_Chapters/2007\\_PIDS.pdf](http://www.itrs.net/Links/2007ITRS/2007_Chapters/2007_PIDS.pdf), 2007.
- [4] G. Indiveri, "Neuromorphic vlsi models of selective attention: From single chip vision sensors to multi-chip systems," *Sensors*, vol. 8, no. 9, pp. 5352–5375, 2008. [Online]. Available: <http://www.mdpi.com/1424-8220/8/9/5352>
- [5] R. Brette and W. Gerstner, "Adaptive exponential integrate-and-fire model as an effective description of neuronal activity," *J. Neurophysiol.*, vol. 94, pp. 3637 – 3642, 2005.
- [6] A. Destexhe, "Conductance-based integrate-and-fire models," *Neural Comput.*, vol. 9, no. 3, pp. 503–514, 1997.
- [7] J. Schemmel, D. Brüderle, K. Meier, and B. Ostendorf, "Modeling synaptic plasticity within networks of highly accelerated I&F neurons," in *Proceedings of the 2007 IEEE International Symposium on Circuits and Systems (ISCAS'07)*. IEEE Press, 2007.
- [8] J. Schemmel, A. Grübl, K. Meier, and E. Müller, "Implementing synaptic plasticity in a VLSI spiking neural network model," in *Proceedings of the 2006 International Joint Conference on Neural Networks (IJCNN'06)*. IEEE Press, 2006.
- [9] J. Schemmel, J. Fierens, and K. Meier, "Wafer-scale integration of analog neural networks," in *Proceedings of the 2008 International Joint Conference on Neural Networks (IJCNN)*, 2008.
- [10] H. P. Langtangen, *Python Scripting for Computational Science*, 3rd ed. Springer, February 2008.
- [11] D. Brüderle, E. Müller, A. Davison, E. Müller, J. Schemmel, and K. Meier, "Establishing a novel modeling tool: A python-based interface for a neuromorphic hardware system," *Front. Neuroinform.*, vol. 3, no. 17, 2009.