SP9 Neuromorphic Computing Platform - Results from SGA2 year 1 (D9.6.1 -SGA2)

Figure 1: The BrainScaleS (upper image) and SpiNNaker (lower image) machines. These machines form the HBP Neuromorphic Computing Platform, offering accelerated learning and programmable flexibility respectively.

Project Number:	785907	Project Title:	Human Brain Project SGA2

Document Title:	SP9 Neuromorphic Computing Platform - Results from SGA2 year 1
Document Filename:	D9.6.1 (D62.1 D30) SGA2 M13 ACCEPTED 190723
Deliverable Number:	SGA2 D9.6.1 (D62.1, D30)
Deliverable Type:	Report
Work Package(s):	WP9.1-WP9.6
Dissemination Level:	PU = Public
Planned Delivery Date:	SGA2 M13 / 30 Apr 2019
Actual Delivery Date:	SGA2 M13 / 26 Apr 2019; ACCEPTED 23 Jul 2019

Authors:	SP9 contributors
Compiling Editors:	Steve FURBER, Andrew DAVISON, Johannes SCHEMMEL, Sebastian HÖPPNER, Wolfgang MAASS, Michael SCHMUCKER
Contributors:	SP9 members
SciTechCoord Review:	
Editorial Review:	

Description in GA:	For consistent presentation of HBP results, SGA2 M12 Deliverables describing the accomplishments of an entire SP or CDP have been prepared according to a standard template, which focuses on Key Results and the outputs that contribute to them. Project management elements such as Milestones and Risks will be covered, as per normal practice, in the SGA2 Year 1 Report.
Abstract:	This Deliverable describes progress over the first 12 months of SGA2 in the development of the Neuromorphic Computing Platform.
Keywords:	Neuromorphic Computing Platform, SpiNNaker, BrainScaleS,
Target Users/Readers:	Scientists, companies and other potential users of HBP results.

Table of Contents

Table of Tables

Table of Figures

1. Overview

Neuromorphic computing is a term used to describe computing systems that differ from conventional computers in that they incorporate a degree of biological inspiration. SP9 supports two world-leading neuromorphic computing platforms: The BrainScaleS platform displays the principle of physical modelling, where electronic circuits directly implement the equations that describe the physics of biological neurons, whereas the SpiNNaker system employs more conventional massively-parallel digital computation, with a novel communication mechanism inspired by the very high levels of connectivity found in the brain.

During SGA2, SP9 activity has focused on both extending the software support for the first generation BrainScaleS and SpiNNaker systems to support an ever-increasing user base and, at the same time, further developing the technology for the second generation systems, which will offer significantly enhanced capabilities compared with the first generation.

Highlights during the first 12 months of SGA2 included:

- SpiNNaker-1: Doubling the capacity of the first-generation SpiNNaker platform to a million cores - the world's largest neuromorphic computing platform.
- SpiNNaker-1: Improvement of the SpiNNaker software to allow synaptic generation on the machine, reducing the execution time of the cortical microcolumn benchmark simulation from 10 hours to 10 minutes.
- SpiNNaker-2: Fabrication of the Jib1 SpiNNaker-2 prototype chip.
- BrainScaleS-1: Facilitating wafer-scale experiments through further commissioning of both software and hardware.
- BrainScaleS-1: Experimental proof of time-to-first spike coding on analogue accelerated neuromorphic hardware, enabling both time and energy efficient pattern classification. Also a class of generative models were implemented that prove the sampling property of the network and demonstrate its applicability to high-dimensional datasets.
- BrainScaleS-2: Demonstrating neuromorphic advantage with experiments showcasing a virtual environment, learning and motor control loop, accelerated and fully implemented on-chip, as well as the successful application of structural plasticity to networks processing auditory and visual stimuli.

2. Introduction

The field of neuromorphic computing has been attracting increasing interest from industry over the course of SGA2. There are a number of factors contributing to this interest, the primary one being a recognition that, while conventional machine learning has made great strides over the last decade, new ideas are required if this progress is to continue into the future, and the biological brain - the source of inspiration for neuromorphic computing - is a promising potential source of such ideas.

Evidence for this increasing interest comes in many forms, including the entry of Intel into the field with its Loihi research prototype chip, the emergence of start-up companies with business plans to develop new neuromorphic platforms or to build software on existing platforms, and the direct funding of neuromorphic research in academic groups by established industrial players.

In SP9, the HBP is maintaining a European lead in the international neuromorphic computing field by supporting two very different systems: SpiNNaker, the world's largest neuromorphic computing system, based on massively-parallel many-core technology; and BrainScaleS, the world's fastest physical modelling system, based on analogue wafer-scale technology. Although both first-generation machines pre-date the HBP and are based upon ten-year-old technologies, they are still highly competitive. The second-generation systems will build upon extensive user-experience with the first-generation systems within the HBP and significantly extend the lead. SGA2 will see the

development of these second-generation systems, which should be ready for production and deployment in SGA3.

Of course, the hardware is only part of the story, and software support for the systems is also a major focus during SGA2. This includes the software required to integrate the two systems into the HBP Neuromorphic Computing Platform, which is used by an ever-increasing number of users, and specific demonstrators for new approaches from theoretical neuroscience and new application domains.

Of particular interest in SGA2 are developments in BioAI that are closing the gap between the learning capabilities of neuromorphic systems and those of conventional deep networks. A number of these developments are being implemented on early prototypes of the second-generation neuromorphic systems, thereby proving their capabilities ahead of the commitment (in SGA3) of significant resources to their manufacture and deployment.

3. Key Result KR9.1: Both first-generation machines integrated into joint platform

3.1 Outputs

3.1.1 Overview of Outputs

- Neuromorphic Computing Platform Monitoring Service (C1637, C3042)
- BrainScaleS-1 Neuromorphic Computing System (version 1=NM-PM1) (C1)
- SpiNNaker-1 Neuromorphic Computing System (version 1=NM-MC1) (C2)

3.1.2 Neuromorphic Computing Platform Monitoring Service

The HBP Neuromorphic Computing Platform consists of a number of services (collectively forming component C3042), in addition to the large-scale neuromorphic computing systems in Manchester and Heidelberg.

In order to maximise the availability of the Platform we have developed a monitoring service (C1637) which tests the accessibility of all services, alerts platform developers in case of unexpected downtime, and provides a web page for users where they can check on platform availability (http://status.hbpneuromorphic.eu). Release of the first version of this service was reported as Milestone MS9.1.1.

3.1.3 BrainScaleS-1

3.1.3.1 Integration into the Joint Platform

See Key Result 9.2, BrainScaleS, Resource Management.

3.1.3.2 BrainScaleS-1 — Internal Alerting and Monitoring

The NMPM1-platform needs a continuous monitoring and alerting system for a stable and reliable operation. The information flow of this process is shown in Figure 2.

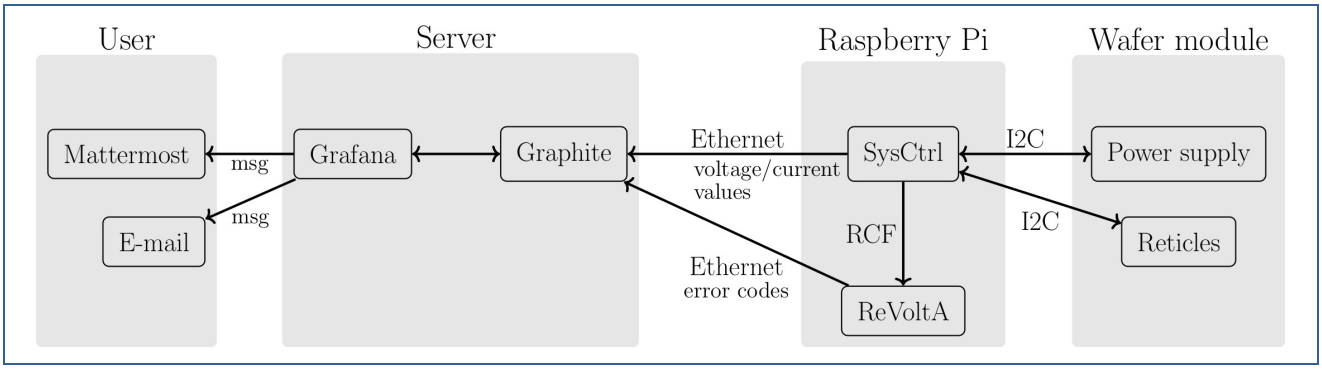

Figure 2: BrainScaleS-1 monitoring and alerting system information flow

The sysCtrl software, which runs on a RaspberryPi, collects the wafer module observables, mainly voltages and digital logic states. The storage of data is done on a Graphite server on a conventional x86-computer. For the presentation of the data to the user, the software tool Grafana is used. Furthermore, Grafana analyses the data and checks it against thresholds, such as wafer temperature limit. If a threshold is crossed, Grafana informs the users and system operators via e-mails and Mattermost messages. Complex checks combining different data types are hard to implement into Grafana. Therefore, an additional module on the RaspberryPi is installed. This is the Reticle Voltage Alerting (ReVoltA) python script. It takes the voltages and the power state of all reticles from sysCtrl and checks if the voltage levels are in the correct range depending on the power state. The simplified result is sent again to Graphite and in case of an error informs the users. A screenshot showing the wafer monitoring dashboard is shown in Figure 3 (contact questions@neuromorphic.eu for access).

Figure 3: Screenshot of the BrainScaleS-1 wafer monitoring dashboard

3.1.3.3 New Analogue-to-Digital (ADC) Subsystem

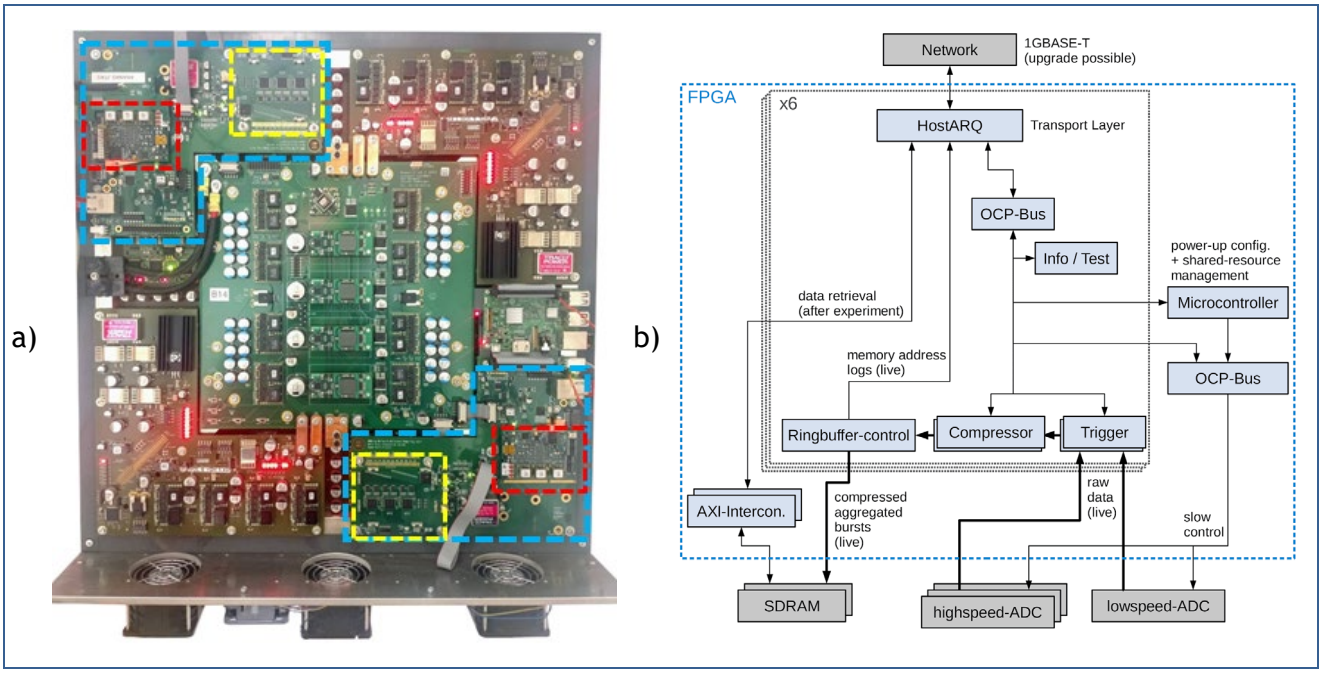

Figure 4:The new Analogue-to-Digital (ADC) Subsystem

a): Backside view of a wafer module, with the two boards for the new ADC subsystem (blue). ADC chips (yellow) are controlled by a plug-in FPGA board (red). b): Structure of the FPGA firmware that is used to operate the new ADC subsystem. All visible electronic modules have been developed and built in Heidelberg in HBP.

An upgrade to the analogue-to-digital conversion (ADC) subsystem has been developed for calibration of the analogue neuromorphic circuits and for membrane voltage digitization purposes (see Figure 4, a)). It eliminates the need for external analogue cabling. The FPGA design of the ADC subsystem has been reworked during the reporting period (see Figure 4, b)). The upgraded analogue-to-digital subsystem now allows for parallel readout of all 96 analogue channels per wafer module. To manage shared resources (such as ADC configuration) and system start-up configuration, a Soft-Microcontroller is integrated into the FPGA.

The raw live data exceeds 36Gbit/s per wafer module. It is delta-compressed, aggregated for each trigger group and stored in local SDRAM ring buffers. A memory aggregation scheme has been developed maximising the throughput to the buffer memory. The software to read out the data and re-assemble the traces from the aggregated format is currently under development.

3.1.3.4 New Wafer Set

Twenty-five wafers have been produced during SGA1, containing improved analogue neuromorphic circuitry ("HICANN v4.1"). After post-processing 19 wafers and testing four of these, no significant production errors were found. The platform is currently being upgraded with these wafers.

3.1.3.5 Improved FPGA firmware of main communication modules

These communication modules implement the interface between the wafer-level communication circuits and the control PC cluster running the software (48 per wafer module). Stability improvements for more reliable platform operation include:

- Upgraded transport layer implementation for communication with control cluster with our current HostARQ implementation.
- Secured JTAG-based communication using this transport layer.

- Added physical link characterisation feature for FPGA-HICANN connection. These links operate at 1GBit/s per HICANN chip.
- Source code now also developed using the continuous integration methodology (cf. KR 9.2).

3.1.4 SpiNNaker-1

3.1.4.1 Integration into the Joint Platform

The SpiNNaker-1 machine has been extended to contain over 1 million cores. Jobs submitted to the NMC resource manager (NMPI) are executed on SpiNNaker by the SpiNNaker job execution engine. This execution engine has been augmented to send information to the NMC monitoring service to allow monitoring of the SpiNNaker service. The same execution engine has been updated to allow users to use the git version of the SpiNNaker software, using any branch or tag available.

3.1.4.2 Integration with the HBP Neurorobotics Environment

We have now installed a local version of the Neurorobotics environment. We have made this work better with SpiNNaker by decoupling the executions; this means that the platforms can drift in terms of which timestep they are on, but this doesn't seem to be too much of a problem in practice so far. We have made it possible to run up to 30 simultaneous NRP experiments with SpiNNaker. We are planning a neurorobotics workshop/hackathon in September 2019.

3.2 Validation and Impact

3.2.1 Actual Use of Output(s) / Exploitation

The monitoring service has already proven useful in alerting platform developers when services are down.

The new BSS-1 ADC subsystem improves analogue readout quality and readout bandwidth. 96 signals can be digitised in parallel (compared to 12 with the previous setup).

New BSS-1 wafers provide improved calibration of analogue synaptic inputs and power supply of the on-wafer network circuits. Both increase usability and stability of the NM-PM1 system.

Improved BSS-1 communication module firmware allows a more stable platform operation.

The 1-million core SpiNNaker machine is now servicing jobs submitted through the HBP Collaboratory.

3.2.2 Potential Use of Output(s)

The monitoring service is intended to provide the basis for the Neuromorphic Computing Platform load-balancing and scaling service (C1638), which will be developed in the second year of the HBP SGA2 grant. This new service will be alerted by the monitoring service when one of the Platform Components is down or overloaded, and will automatically deploy additional instances of the relevant Component. This will increase the overall availability and reliability of the Platform.

The new BSS-1 ADC subsystem allows for more parallel and faster readout of analogue quantities. This results in faster calibration and experiment turnaround times on the Platform and allows a more fine-grained evaluation of results. It supports now simultaneous measurements on all wafer modules of the BSS platform.

The availability of the 1-million core SpiNNaker machine will allow users to submit larger jobs to the platform. Additionally, the service can now run more small jobs at the same time. The ability for users to use the git versions of the software means thatthey can access newly implemented features and bug fixes before they are released.

3.2.3 Publications

3.2.4 Measures to Increase Impact of Output(s): Dissemination

- The SpiNNaker and BrainScaleS platforms have been presented at the HBP booth at the EU's ICT2018 event in Vienna.
- Tutorials on the usage of the SpiNNaker-1 system and the BrainScaleS-1 system have been given during the 2019 NICE workshop. Agenda: http://niceworkshop.org/nice-2019/agenda/.
- https://twitter.com/BrainScaleS (823 followers) and HBPNeuromorphic (159 followers) are used to promote results.

4. Key Result KR9.2: Comprehensive software suite for the operation of neuromorphic machines

4.1 Outputs

4.1.1 Overview of Outputs

- SONATA support in PyNN - C349, C1718
- Graphical tools for building neuronal network models - C1655
- BrainScaleS-1 and -2 Operation Software — C2787, also contributes to C1, C454, C457
- Neuromorphic benchmarks C2575, C2576, C2735
- MUSIC 3.0 - C347
- SpiNNaker-1 and -2 Operation Software - C2

4.1.2 Interoperability of PyNN and SONATA

Sharing and reuse of models is essential for ensuring the validity and reproducibility of results in computational neuroscience. The Blue Brain Project (EPFL) and the Allen Brain Institute jointly launched an effort to develop a community standard data format for sharing large-scale neuronal network models with explicit connectivity (connections provided as an explicit list, rather than generated algorithmically). This effort was taken up by other members of the INCF Special Interest Group on Standardised Representations of Network Structures.

SONATA is being adopted as the standard model description format for the HBP Brain Simulation Platform (BSP). As part of the pipeline being developed for simulating BSP models on neuromorphic hardware, we have extended the PyNN software package (Components C349 and C1718) so that (i)

SONATA models can be imported and simulated with PyNN and (ii) PyNN scripts can be exported in SONATA format. This work was done jointly with SP6, and was made available in PyNN v0.9.4.

4.1.3 Graphical tools for building neuronal network models

The Neuromorphic Computing Platform executes simulations for which the model description and simulation protocol must be written as Python scripts, using the PyNN application programming interface (API). However, PyNN necessarily requires familiarity with computer programming, which makes both the models and the modelling approach less accessible to researchers and students lacking programming experience. We have developed a Collaboratory app which allows users to create models of point spiking neurons using a graphical interface, without any programming, and then run simulations of these models on the HBP's BrainScaleS and SpiNNaker neuromorphic computing systems. The app can also be used to generate and download scripts for the NEST and NEURON simulators. A screenshot of the app is shown in Figure 5 below. More information is available in the public HBP report "Working prototype of a Collaboratory app for graphical model building" (SGA2 D9.1.1 (D57.1, D90)).

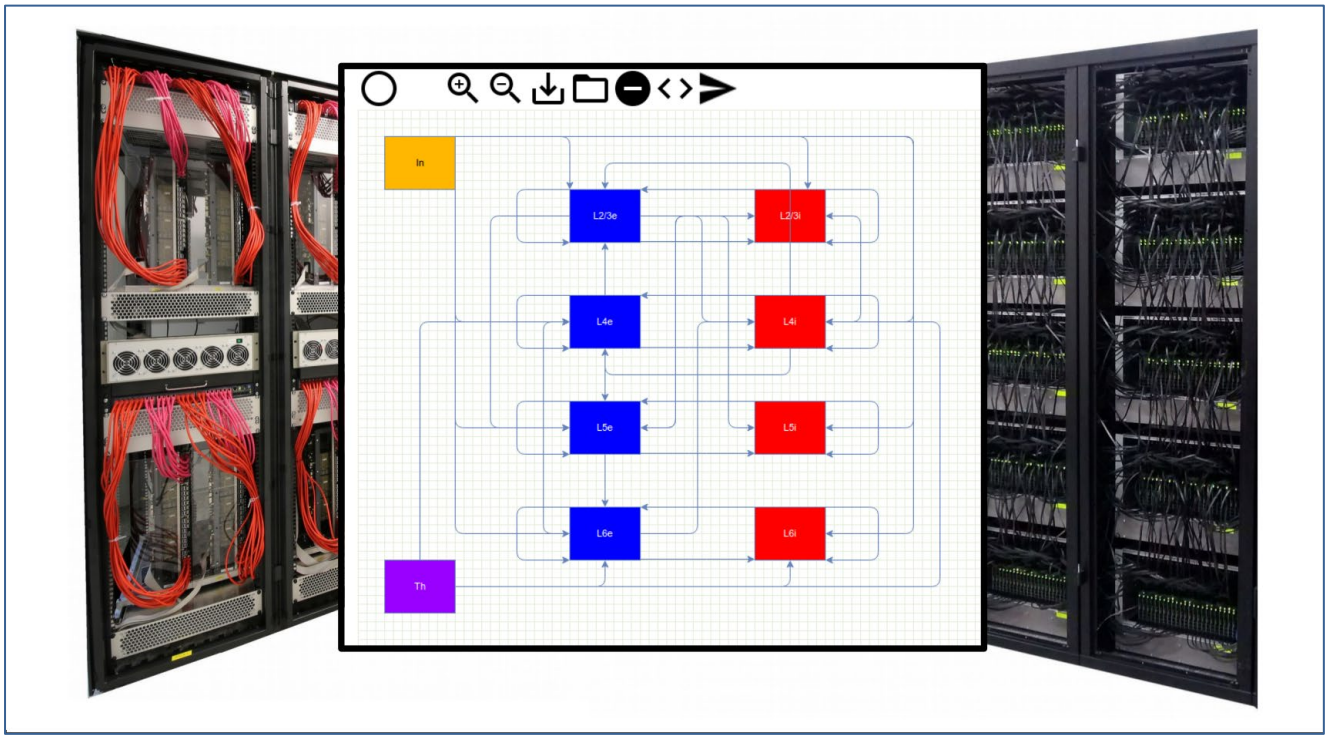

Figure 5: Screenshot of the PyNN model builder app, showing a model of a cortical column.

The model can be emulated on the BrainScaleS system (left) and simulated on SpiNNaker (right), with simulations launched directly from the app.

4.1.4 BrainScaleS Operation Software

4.1.4.1 BrainScaleS Common

4.1.4.1.1 Software Development Process

The development process involves code review, continuous integration and a reproducible container-based software environment. The process is described in SGA2 Deliverable D9.2.1 (D58.1, D91) and is also valid for BrainScaleS-1.

4.1.4.1.2 Software Testing and Deployment

The general idea of continuous integration and automatic software deployment is described in Deliverable SGA2 D9.2.1 (see "Continuous Integration", Chapter 2.4.2).

Figure 6 shows the development of CI-related jobs in the resource management system since the start of SGA2.

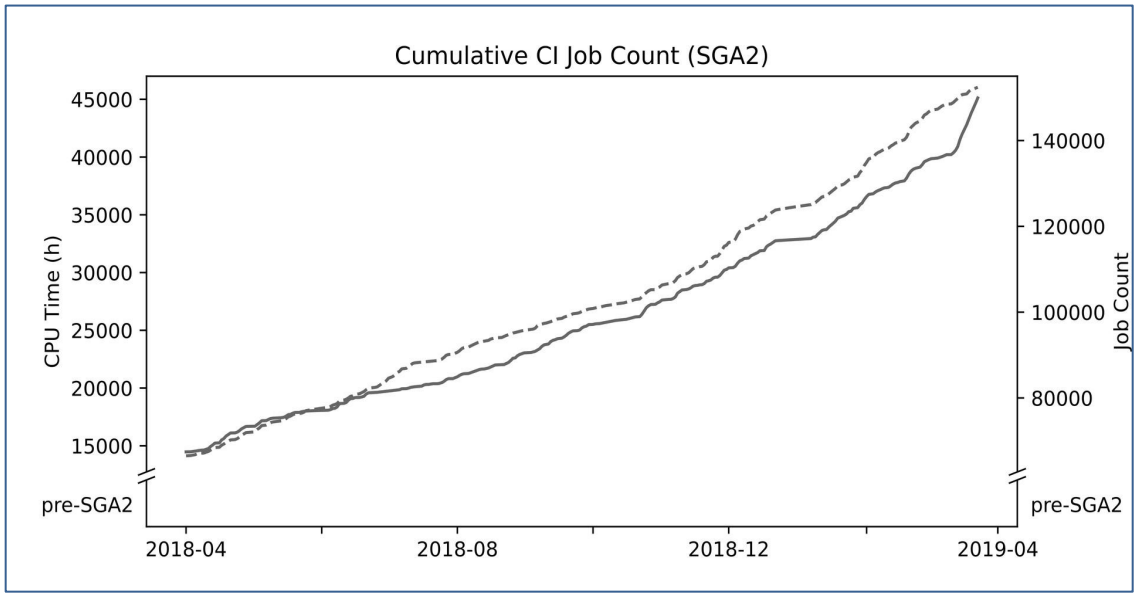

Figure 6: Development of CI jobs in the BrainScaleS queuing system since the start of SGA2 (Key: dashed: job count, solid:CPU time).

4.1.4.1.3 Resource Management

Access to compute resources (including regular compute nodes, BrainScaleS-1 wafer systems and BrainScaleS-2 prototype systems) is governed by slurm, a workload manager. Several plugins for slurm were developed (automatic data network configuration, spawning control daemons for prototype systems) and made available as open source on github.

The NMC resource management system (NMPI) is linked to the BrainScaleS-internal slurm-based resource manager. A per-job mapping between NMPI and internal slurm jobs is handled by a python-based adapter which synchronizes the job states between the queuing systems.

4.1.4.2 BrainScaleS-1

4.1.4.2.1 Configuration Visualisation

Further development of the web-based visualisation (webvisu on github) allows for full-wafer visualisation of on-wafer ("Layer-1") routes, neuron distributions and synapses (Figure 7). The latest version is available online.

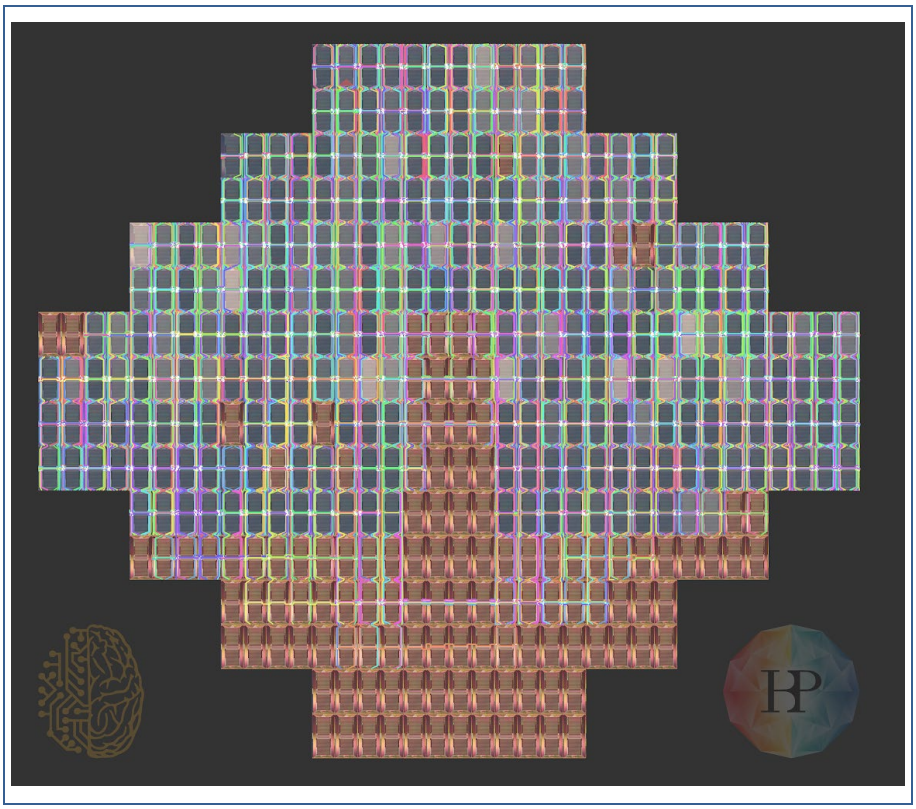

Figure 7: Demo output of the Wafer Visualisation tool.

4.1.4.2.2 Configuration Software

The BrainScaleS-1 software stack now supports parallel configuration of FPGAs and wafers.

Configuration and hardware component selection was optimized to allow for better utilisation of wafer resources, thus allowing larger networks to be emulated on a single wafer.

All features combined, the full configuration of a full wafer is reduced from more than 10 to below 2 minutes with the configuration of the floating gates being the dominant contribution. For small differential configuration changes, the experiment rate can now reach up to multiple Hertz.

4.1.4.2.3 Map & Route Software

The BrainScaleS-1 system uses the "marocco" layer to place and route user-defined neural networks topologies onto the neuromorphic system. Based on the latest user feedback, several map & route algorithms have been modified to improve the result: clustering dendritic as well as axonal communication partners during neuron placement. A reduction of Layer-1 bus usage prevents exhaustion of Layer-1 routing resources, while also needing fewer synapse drivers at the targeted synapse arrays, as they can be shared by a larger number of presynaptic neurons. Most of the place and route algorithms have been improved to keep the electrical load within the constraints. Figure 8 depicts a simple, single-source, multi-target routing result on the left, and an improved routing result on the right. The latter takes advantage of all three possibilities to inject into a target synapse array.

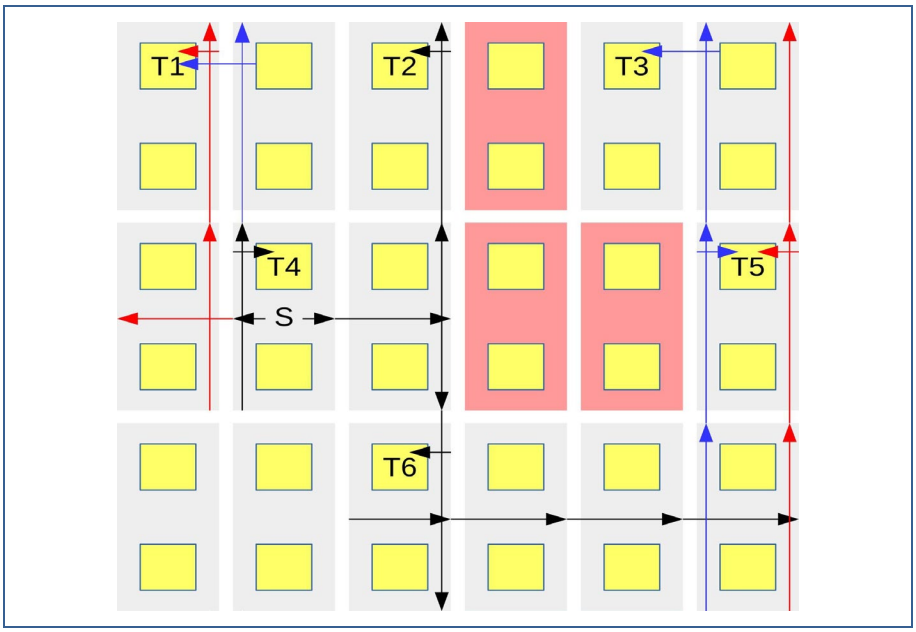

Figure 8: Layer-1 routing improvements in BrainScaleS-1

Key: red: old, blue: new. The routing algorithm now supports injection into neighbouring chips (e.g. T1 from right neighbour).

4.1.4.3 BrainScaleS-2

SGA2 Deliverable D9.2.1 covers all BrainScaleS-2 related software components.

4.1.5 Neuromorphic Benchmarks

The benchmark framework SNABSuite (C2735) was extended to 57 benchmark tasks, providing measures for direct comparison of performance indicators (e.g. resource efficiency, quality of the result, robustness) for both HBP neuromorphic platforms SpiNNaker and BrainScaleS (inputs to C2576). Most of these have been scaled to cover all sizes of networks (from single processing core to full board size). Several more prototypes for benchmarks exist and will be finalised in the next months, covering the field of solving constraint satisfaction problems and classification problems. Accompanying with these benchmarks, the suite provides tools for analysis and visualisation of benchmark results (C2575), which have been extended together with the respective benchmark tasks. Furthermore, the suite has been integrated into a web service, which tracks the development of performance indicators on all target platforms over time.

4.1.6 MUSIC 3.0

MUSIC (C347) is a communication framework in the domain of computational neuroscience and neuromorphic computing which enables co-simulations, where components of a model are simulated by different simulators. It consists of a software API and C++ library which can be linked into existing software with minor modification and enables the communication of neuronal spike events, continuous values and text messages, while hiding the complexity of data distribution over ranks, as well as interlocking free scheduling of communication in the face of loops.

MUSIC is, so far, the only available tool in the domain of computational neuroscience which enables co-simulation with largely independent participating simulation software applications.

The MUSIC interface previously provided two phases:

- Setup - port creation and configuration

- Runtime - simulation loop.

New ports could only be created during the Setup phase.

The MUSIC interface version 3.0 (C1806; https://github.com/incf-music/music-api-3.0) replaces these phases with two states:

- STOPPED - where port creation/deletion and configuration is possible
- RUNNING - simulation loop where data is streamed.

Since new ports can be created and deleted in the STOPPED state and, since switches between the states can be made during the full run of the simulation, this allows for reconfiguration of port connectivity during simulation (C1807).

ControlPorts have been introduced which can switch a MUSIC-aware application between the two states.

Normally, port connectivity is specified outside of applications/models (important to preserve modularity and re-usability). The new interface contains a PortConnectivityManager which can be used to write a port managing application (such as the master in the Neurorobotics Platform CLE).

4.1.7 SpiNNaker Operation Software

4.1.7.1 Booting the Million Core SpiNNaker Machine

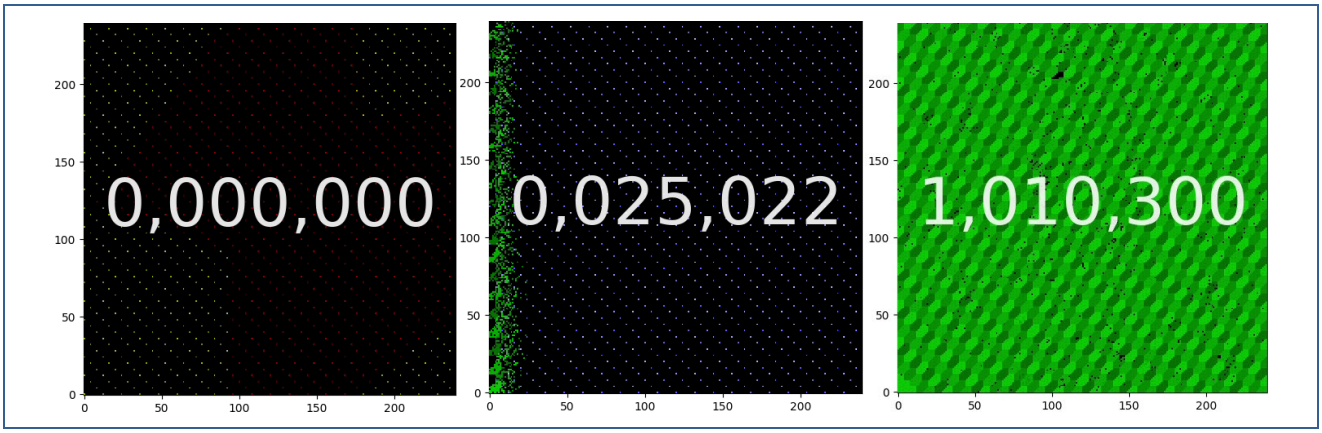

Figure 9: Booting the SpiNNaker 1Million Machine

There were some issues with the boot process when the SpiNNaker machine was scaled up to 1 million cores that meant that the machine would not boot consistently or reliably. These were resolved in time for the “Million Core switch on” event, and a visualisation of the boot process was also created to show that the booted machine had over 1 million cores in a single machine (Figure 9).

4.1.7.2 Synaptic Expansion on Machine

Previous versions of the SpiNNaker software convert the PyNN network connectors (such as OneToOneConnector and FixedProbabilityConnector) into the data structures required by the SpiNNaker binaries on the host machine. This data is then transferred to the machine in fully expanded form, so the generation of data and the loading of data were both slow. This has been replaced by a process in which the connector descriptions for certain connectors are now loaded on to the machine rather than the expanded matrix. A SpiNNaker application is then run in advance of the neural network application binaries; this application performs the expansion of the synapses on the machine. This greatly reduces the amount of data generated on the host machine, meaning the data generation process is faster and the loading of the data is faster. Additionally, the actual

conversion of the connector descriptions in to synaptic matrices can now happen in parallel on the machine itself, again reducing the time to perform this operation. In practice, this has reduced the overall execution time of the benchmark cortical microcircuit simulation from 10 hours to 10 minutes.

4.1.7.3 Faster extraction of data

We have made the reading of results from the machine faster by changing how the communication works. Instead of using SpiNNaker SDP and P2P messages with state and confirmations per message, we now use a single SDP message to trigger the transfer of large quantities of data, which are sent to the Ethernet chip using Fixed Route messages (similar to multicast, but with only one destination), which then sends out the message without further use of network communication. This allows the transfer to reach up to 40Mbits/s from any chip in the machine, where the previous methods only allowed for 8Mb/s maximum from non-Ethernet connected chips. We have also now implemented code in Java to run on the host machine, which allows us to achieve parallel extraction of the data over multiple boards. This has shown to be highly scalable, up to the limit imposed by the network bandwidth available.

4.1.7.4 PyNN 0.9 compatibility

We have made the software compatible with PyNN 0.9. This includes adding some features such as PopulationView support, and the ability to record data (such as spikes and membrane voltage) from a subset of the neurons in a population, as well as adding support for additional PyNN connectors, and adding an “alpha” synapse implementation.

4.1.7.5 Neuron Model Implementation Changes

We now support “general neuron” implementations, which means SpiNNaker implementations of neuron models that include multiple components in a single implementation (where components are the synapse model, the neuron membrane model, the input conductance or current model and the threshold model). This will allow users to support more complex neuron models that don’t fit into our componentisation model.

4.1.7.6 Code Evaluation

The Neuron code has been evaluated with respect to the performance on the machine. This has given us a better understanding of how different parameters affect the running of the code. This is leading to some promising work on optimising the various parts of the SpiNNaker code, which is still in progress.

4.1.7.7 Logging Changes

The SpiNNaker logging code has been modified to reduce the impact of log messages on the instruction memory. This dynamically updates the C code and creates a dictionary of log messages, replacing the actual message with an ID. During extraction of the log data, these are then converted back in to the original message, so the process is transparent to the user.

4.1.7.8 Integration Testing

We have installed a local server and Jenkins service on which we can now run integration tests of the software against the 1-million core SpiNNaker machine.

4.1.7.9 Nengo

Work has started on moving the Nengo implementation on SpiNNaker on to the main tool chain. This work was previously supported by PhD students, but we are working to make it work on the same set of tools as is used for the SNN software.

4.1.7.10 Neuron Routing Filtering

We have updated the code to better filter out unused spikes on reception at a core. The multicasting of spikes means that each core that makes up a population receives all spikes from the sending core even if no synapses are formed from the sending neuron. This means the core has to do work to read an empty synaptic row from SDRAM. These changes add a layer that can look up if each row has data in it from local memory before performing the DMA transfer, meaning that more spikes overall can be processed.

4.1.7.11 Code Quality Reviews

A number of reviews of the code have taken place, and this has been updated to improve the quality of the code, making maintenance easier in the future.

4.2 Validation and Impact

4.2.1 Actual Use of Output(s) / Exploitation

PyNN support for SONATA was also released only a short time ago, so we have received limited feedback. It has been successfully used by other members of the INCF SIG on Standardised Representations of Network Structures, and has been used to produce a figure for an in-preparation publication about SONATA.

The PyNN model builder app has been released as a public app in the Collaboratory, where anyone with an HBP Identity account can use it. It was released only a short time before the writing of this report, so we do not yet have any user numbers or user feedback.

The BrainScaleS-1 and -2 software stacks are used for all experiments performed on the Neuromorphic Computing Platform; all software repositories are open source and published on https://github.com/electronicvisions.

The benchmark suite is published on https://github.com/hbp-unibi.The suite is integrated into the benchmark web service https://benchmarks.hbpneuromorphic.eu and executed and evaluated on a regular basis.

Uses of MUSIC include the NEST module Spore that provides a simulation framework for reward-based learning with spiking neurons (https://github.com/IGITUGraz/spore-nest-module; Kaiser *et al.* (2019), Kappel *et al.* (2018)) and in a toolchain connecting neuronal simulations with the ROS environment (Weidel *et al.* (2016), Jordan *et al.* (2017), Bahaguna *et al.* (2018)).

The SpiNNaker-2 software stack is used for all simulations executed on the platform. The git version of the software is used by various researchers, particularly those at UMAN. The software is open source and available from https://github.com/SpiNNakerManchester/.

4.2.2 Potential Use of Output(s)

PyNN support for the SONATA format, together with PyNN's pre-existing support for the NeuroML format, should increase the ease of reproducibility and reuse of large-scale, detailed neuronal network models. In particular, this functionality helps form a pathway from the detailed, biophysical

network models being developed on the Brain Simulation Platform (and elsewhere, outside the HBP) to simulations on neuromorphic hardware.

The PyNN model builder app is only at the prototype stage, but is nevertheless already useful. At this stage, it could potentially be used already by students in summer schools, MOOCs, tutorials, etc. As it is further developed, it will cover an increasing proportion of the functionality of PyNN and the underlying neuromorphic systems, and will include features for working with larger-scale, more complex models and for viewing/analysing the results of neuromorphic simulations. Once these more advanced features are available, the app will potentially be able to replace/augment Python scripting for active research with the neuromorphic systems.

The BrainScaleS-1 and -2 software stacks are open for experimenter feedback and present a living code base. Optimisations are being integrated, as they become available, to increase the level of experiment complexity that they can support. Examples for future developments are code generation supporting the embedded plasticity processor on the BrainScaleS-2 platform, or the placement of structured neurons, using a future release of the PyNN API.

The benchmark suite provides measures for comparison of performance and efficiency of neuromorphic platforms. It can drive the active development and reveal potential improvements in the current software stack, as well as contribute to the development of future hardware by defining a set of typical workloads. Furthermore, the suite provides a tool to potential users for performance estimation of their specific application.

With the rise in interest in multi-scale modelling, both in HBP and in the wider community, tools for coordinating multiple simulation engines are likely to increase in importance. MUSIC is a proven technology, and the new version, 3.0, increases its flexibility and hence increases the range of domains in which it can be used. New uses of MUSIC in the next stage of HBP, in the areas of brain simulation, neurorobotics and neuromorphic computing, are already being planned.

The SpiNNaker-1 software stack is in constant development to address the needs of the user base. The improvements made are mostly driven by the user experience of the software; the changes described above will allow users to run their simulations in less overall time on the platform. Additionally, the writing of new models for the platform has been simplified and the ability to debug these models has been improved.

4.2.3 Publications

- Rhodes, O. *et al.* (2018). sPyNNaker: a software package for running PyNN simulations on SpiNNaker. *Front. Neurosci.* 12:816. doi: 10.3389/fnins.2018.00816 (HBP publication P1805)
- Rowley, A. G. D. *et al.* (2018). SpiNNTools: the execution engine for the SpiNNaker platform. *Front. Neurosci.* 13:231. doi: 10.3389/fnins.2019.00231 (HBP publication P1806)

4.2.4 Measures to Increase Impact of Output(s): Dissemination

- The release of PyNN 0.9.4 was announced on Twitter, where it was viewed over 800 times, and had 35 interactions (likes, retweets) to date.
- The Wafer Visualisation tool is used in the hardware system tutorials (e.g. at NICE 2019)
- HBP CodeJam #9 in Palermo

5. Key Result KR9.3: Operational prototype of a second generation BrainScaleS chip featuring on-chip local plasticity and non-linear dendritic processing

5.1 Outputs

5.1.1 Overview of Outputs

Output refers to C454 "BrainScaleS-2 standalone, single-chip, physical model system" and Milestone MS9.2.1 "BrainScaleS-2 Single Chip System".

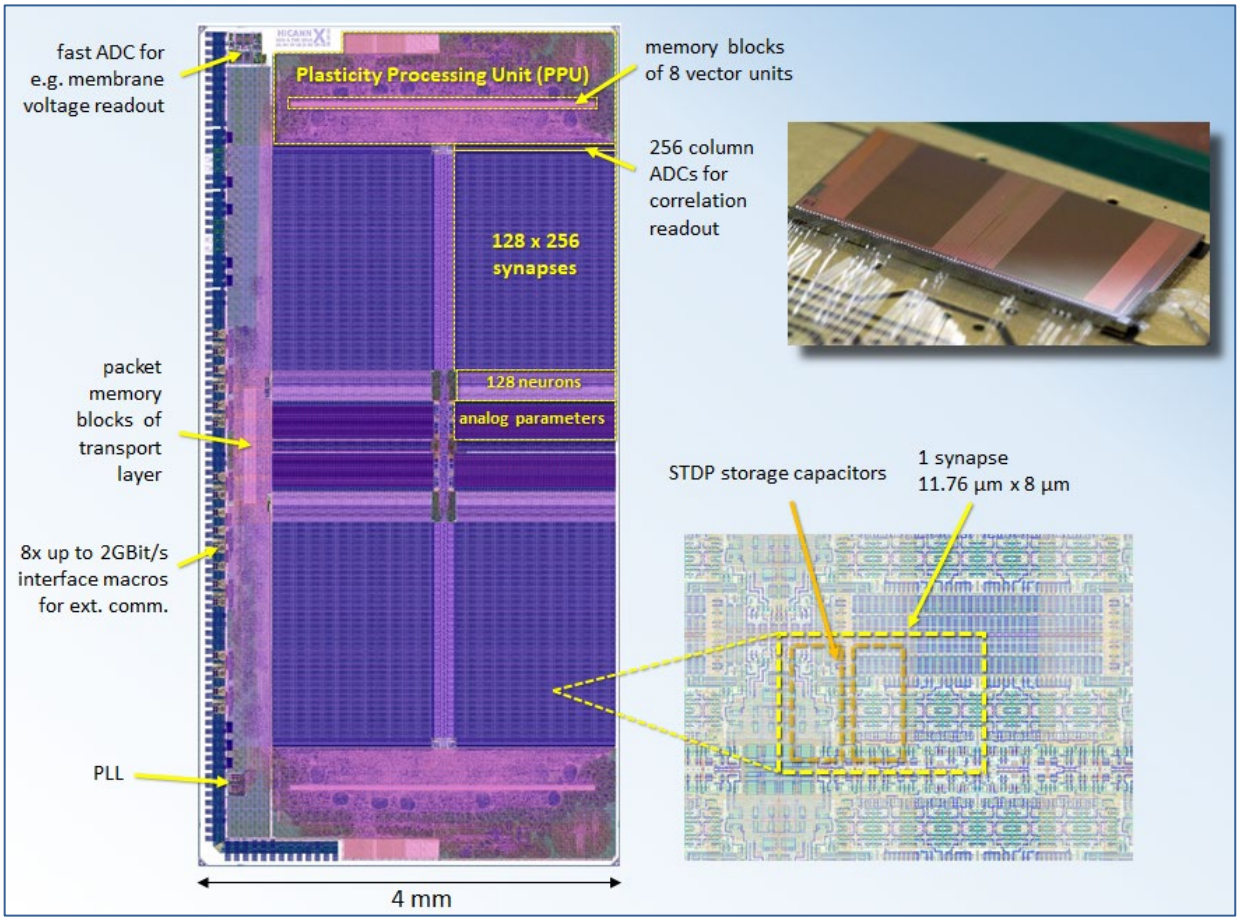

Figure 10: The HICANN-X microchip.

Annotated layout drawing and photograph (insert, top right) Mirrored elements are annotated only once.

Based on small-scale prototype chips, a single-chip system was delivered in SGA1, MS9.2.1. Based on these results, a full-scale chip (HICANN-X, 512 neuron circuits, 128k synapses, acceleration factor of 1,000 compared to biological real-time, on-chip local plasticity and non-linear dendritic processing) has been developed during SGA1 which will be the basis for the upgraded BrainScaleS-2 single chip system (see Figure 10). The required hardware, firmware and software to test the HICANN-X chips was developed during the reporting period. All test system components are designed to be reusable in the single-chip system.

5.1.2 Output 1: BrainScaleS 2 standalone, single-chip, physical model system

Refers to Component C454. In this output, we summarise results that are required for operation of the prototype chip and first test results.

5.1.2.1 Test system

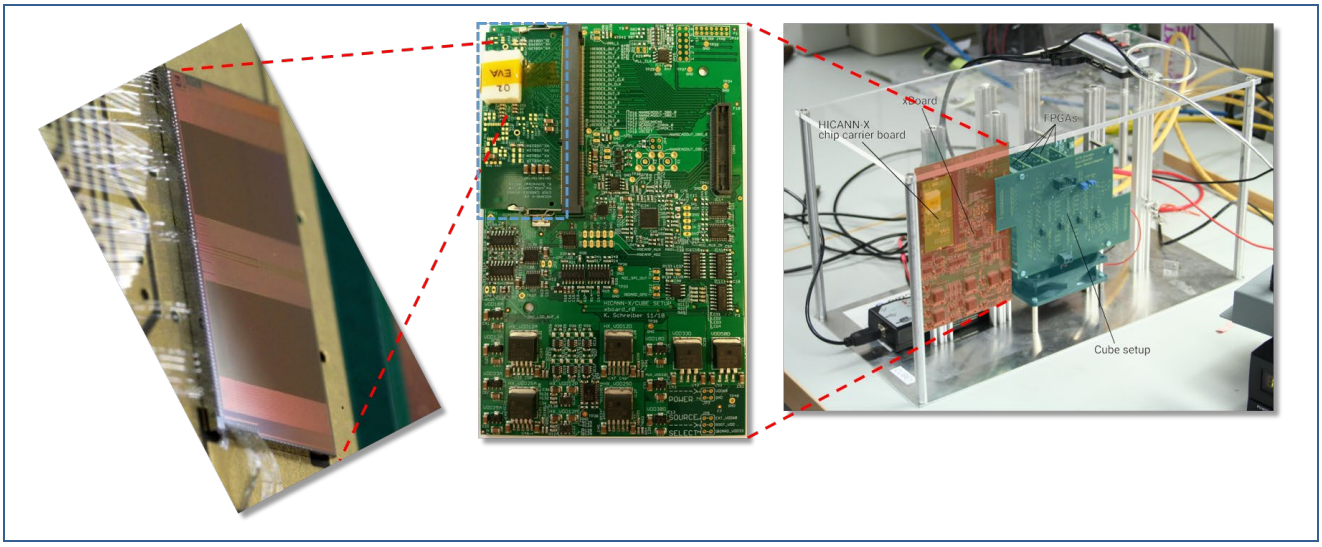

Figure 11:: HICANN-X test system and BrainScaleS-2 single chip system.

The "cube setup" (right, green overlay) interfaces with HICANN-X chips (left) via the xBoard (right, red overlay) and chip carrier board (middle, blue).

Figure 11 shows the entire HICANN-X test setup consisting of the “cube setup” (right, green overlay) which has already been used as a prototyping platform for BSS-1 single chip system, the xBoard (red), and the HICANN-X chip carrier board (middle, blue), both developed and fabricated in SGA2. The cube setup contains four FPGA communication modules. Identical FPGA boards are used in the BSS-1 wafer system. Therefore, only minor firmware and software changes will be required to operate the final BrainScaleS-2 wafer, which will contain the wafer-version of the HICANN-X chips.

Each HICANN-X chip is connected to one of these communication module by a custom 2x8-pair differential high-speed link and the necessary control signals. HICANN-X chips can be identified by a unique label that is stored together with chip-related meta-information on a permanent memory on their carrier boards.

The xBoard contains analogue and digital periphery, as well as circuits for supply voltage generation and monitoring. It generates all supply voltages required by HICANN-X. For test and experimental purposes, these voltages and the corresponding supply currents can be set and monitored from the host computer. Moreover, all analogue bias currents and reference voltages for the neuromorphic HICANN-X components are generated on the board and can be configured by the FPGA as well. A fast analogue-to-digital converter (ADC) with an input multiplexer can further capture all signals that are relevant for commissioning, calibration, parameter sweep response, or any other test purposes, including analogue neuron membrane voltages.

5.1.2.2 FPGA firmware

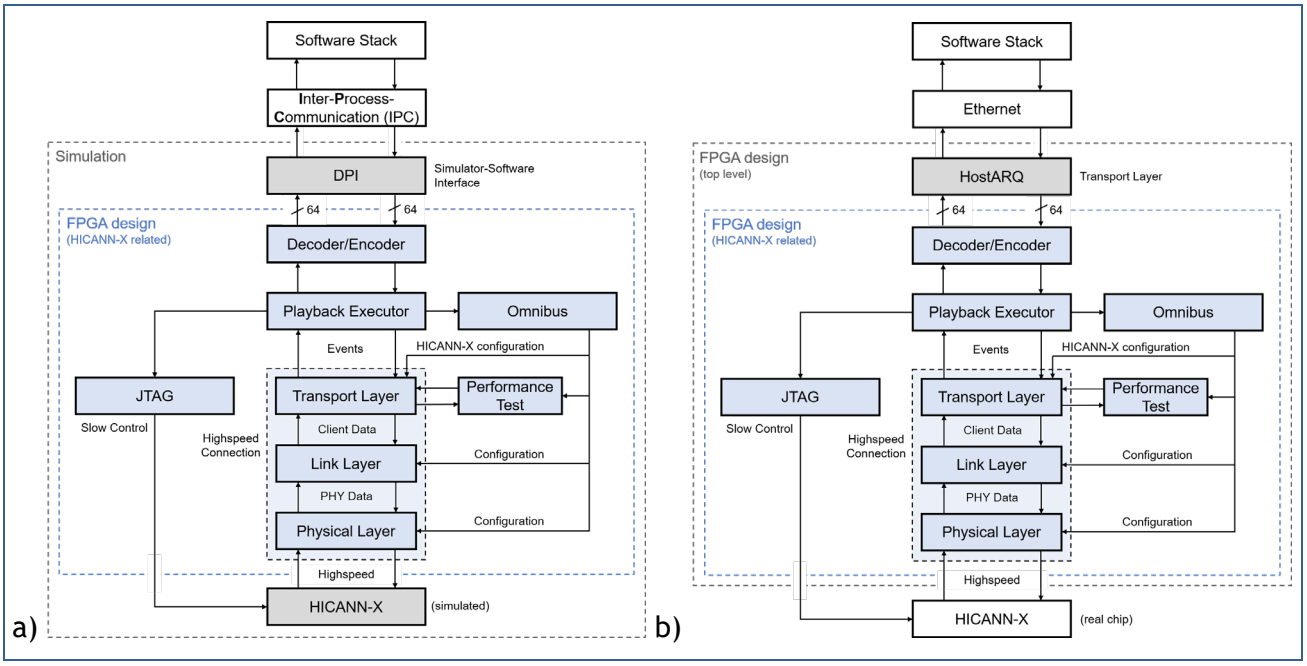

Figure 12: Structure of FPGA firmware.

Key: a) Simulation-only in verification environment. b) With connection to real chip, on single-chip system.

The BSS-1 communication modules are used for communicating with HICANN-X. This ensures reusability of all wafer-scale system components during upgrading to BSS-2.

The main functionality of the FPGA is to transmit configuration data and provide precisely timed release for neuronal events to HICANN-X by eliminating host network jitter, as well as a buffer for recording neuronal event traffic from HICANN-X. During the reporting period, firmware development has progressed to a state where chip tests and experiments with low memory requirements are possible. The firmware structure is shown in Figure 12 b).

Verification of the FPGA design is done in simulation together with HICANN-X, i.e. all involved code can be verified in a system level testbench. This allows for software development in parallel with the FPGA development because both can be tested against each other in simulation with minimal mocking. The simulation set-up is shown in Figure 12 a).

5.1.2.3 Chip test

The HICANN-X microchip was received from the manufacturer in December 2018. Initial tests showed no measurable functionality; not even leakage currents from the IOs to the supply could be observed. The reason for this error was identified, but a second production run was necessary and we are currently (March 2019) waiting for the corrected chips to arrive for testing. Please refer to SGA2 MS9.2.1 for a detailed error description.

5.1.2.4 Verification environment

The system-level test bench implements a remote procedure call-based interface to the simulator with tight coupling using SystemVerilog's Direct Programming Interface (DPI) mechanism (see Figure 12 a)). Within every simulator clock cycle, data can be exchanged with the software: instructions or spike data can be sent to the simulated FPGA and HICANN-X, or answers can be received from the FPGA. As the simulation runs very slowly and with a large timing jitter compared to the hardware, the timing control has to be implemented in software.

5.2 Validation and Impact

5.2.1 Actual Use of Output(s) / Exploitation

The small-scale prototype chips (HICANN-DLS) were used for the publications and events listed in the following sections.

The verification setup described is being used for FPGA firmware and software development. In this way, usability of the prototype chip can be improved while waiting for re-manufacturing.

5.2.2 Potential Use of Output(s)

The operational prototype system will be used to upgrade the BrainScaleS-2 single-chip system with the HICANN-X chip, providing for the first time the 2nd generation BrainScaleS platform to HBP users.

5.2.3 Publications

The following papers were published during the reporting period. Results are based on the existing BrainScaleS-2 single-chip system, operating the HICANN-DLS prototype chips. These contain a reduced feature set of the HICANN-X chip.

- Syed Ahmed Aamir*, Yannik Stradmann*, Paul Müller, Christian Pehle, Andreas Hartel, Andreas Grübl, Johannes Schemmel and Karlheinz Meier: An Accelerated LIF Neuronal Network Array for a Large Scale Mixed-Signal Neuromorphic Architecture. IEEE Transactions on Circuits and Systems I: Regular Papers. DOI: 10.1109/TCSI.2018.2840718 (HBP publication P1448)
- Syed Ahmed Aamir, Paul Müller, Gerd Kiene, Laura Kriener, Yannik Stradmann, Andreas Grübl, Johannes Schemmel and Karlheinz Meier: A Mixed-Signal Structured AdEx Neuron for Accelerated Neuromorphic Cores. IEEE Transactions on Biomedical Circuits and Systems. DOI: 10.1109/TBCAS.2018.2848203 (HBP publication P1335)
- Timo Wunderlich, Akos F. Kungl, Eric Müller, Andreas Hartel, Yannik Stradmann, Syed Ahmed Aamir, Andreas Grübl, Arthur Heimbrecht, Korbinian Schreiber, David Stöckel, Christian Pehle, Sebastian Billaudelle, Gerd Kiene, Christian Mauch, Johannes Schemmel, Karlheinz Meier, Mihai A. Petrovici: Demonstrating Advantages of Neuromorphic Computation: A Pilot Study. Frontiers in Neuroscience, DOI: 10.3389/fnins.2019.00260 (HBP publication P1721)

5.2.4 Measures to Increase Impact of Output(s): Dissemination

- A live-on-tape recording of fully local unsupervised learning on the BrainScaleS-2 prototype chip HICANN-DLS has been uploaded (data otherwise not yet published): https://youtu.be/APQYykiJeKk; similarly, a desktop recording demonstrates reinforcement learning and a virtual environment: https://youtu.be/LW0Y5SSIQU4
- A tutorial on the usage of the BrainScaleS-2 hardware has been given during the 2019 NICE workshop. Agenda: http://niceworkshop.org/nice-2019/agenda/
- https://twitter.com/BrainScaleS (823 followers) and HBPNeuromorphic (159 followers) are used to promote results

6. Key Result KR9.4: Operational prototype of a second generation SpiNNaker chip featuring 10-fold improved energy efficiency, 144 Cortex M4F and 36 GIPS/Watt per chip

6.1 Outputs

6.1.1 Overview of Outputs

- Output 1: SpiNNaker next generation (NM-MC2, SGA2) chip (Component C453), demonstrating new key chip hardware feature for >10-fold energy efficiency improvement.

6.1.2 Output 1: SpiNNaker next generation (NM-MC2, SGA2) chip

A new processing element (PE) for the next generation SpiNNaker2 system has been developed. Its structure is shown in Figure 13:

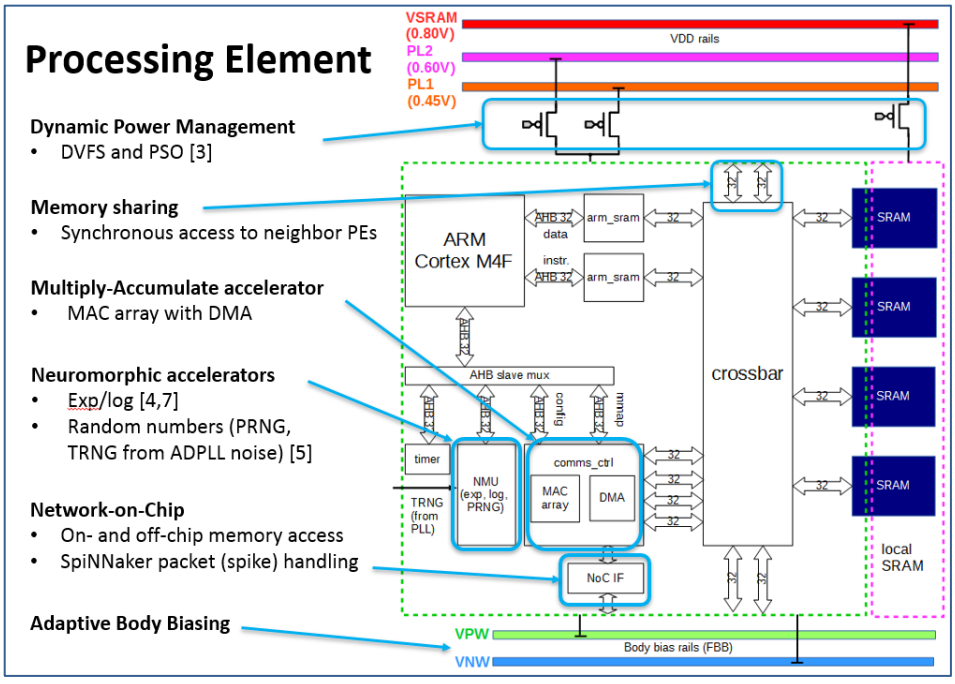

Figure 13: SpiNNaker-2 processing element

The main extensions to enhance the neuromorphic compute performance and energy efficiency are a neuromorphic math unit (NMU) with exponential function and pseudo random number accelerators and a multiply-accumulate (MAC) accelerator which can offload matrix multiplication and convolution from the processor core. It has been implemented in a test chip in 22nm FDSOI CMOS (JIB) which contains 8 PEs organized in 2 quad PEs (QPE). The QPEs are to be scaled in an array in the SpiNNaker2 chip with up to 136 total PEs. A key feature of the PE in GLOBALFOUNDRIES 22FDX technology is the possibility to operate at ultra-low supply voltages (down to 0.40V) to maximise its energy efficiency. This is enabled by an adaptive body biasing approach. On top of this, dynamic voltage and frequency scaling (DVFS) is implemented to trade-off the temporal peak performance requirements (e.g. in simulations time steps with spike bursts) versus the demand for low power

consumption in simulation cycles with low activity. The results of SpiNNaker2 prototype from SGA1 [2] (Component id: C467) have been used to optimizes the DVFS architecture. As result 2 DVFS power levels are used as a compromise between energy efficiency enhancement and the overhead for multiple voltage level supplies. The layout of the JIB testchip in GLOBALFOUNDRIES 22FDX technology is shown in Figure 14:

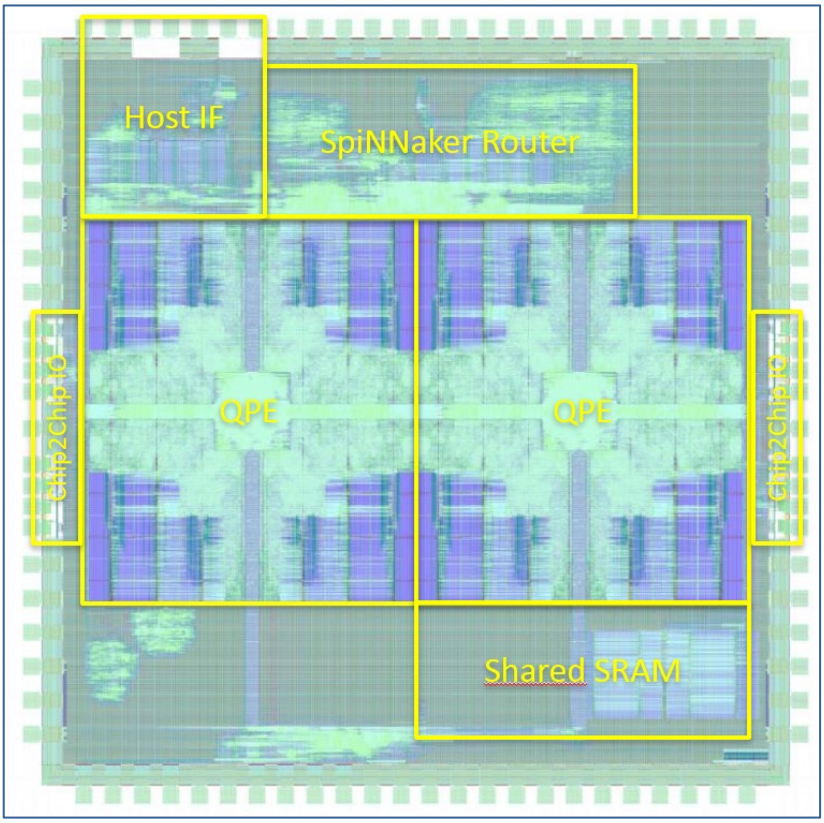

Figure 14: Layout of the JIB testchip

The chips have been taped out at GLOBALFOUNDRIES. A lab evaluation setup PCB with detailed measurement circuits is underdevelopment. Prior to tape-out, extensive performance analyses and power analyses were performed. This included the prototyping of a complete QPE on an FPGA prototype (Xilinx Virtex7) that allows for neuromorphic software execution and performance profiling. Table 1 compares the processing element performance and energy efficiency compared to SpiNNaker1.

Table 1: SpiNNaker-2 vs. SpiNNaker-1: processing element performance and energy efficiency

	SpiNNaker 1 [1]	SpiNNaker2 prototype from SGA1 [2] (C467)	SpiNNaker2 prototype from SGA2 (this deliverable) (C2628)
Processing Element Features			
Technology	130nm	28nm	22nm FDSOI
PE clock frequency	200MHz	125MHz to 500MHz	200 to 400MHz
PE MAC accelerator	no	no	Yes
PE neuromorphic accelerators (exp, PRNG)	no	yes	Yes
Power Management	no	Dynamic voltage and frequency scaling (3 levels)	Dynamic voltage and frequency scaling (2 levels), adaptive body biasing
Nominal Supply Voltage	1.2V	0.70V to 1.0V	0.40V to 0.60V

PE processor energy efficiency [pJ/operation] at room temperature	130 (measured)	45 (measured)	11 (estimated from sign-off power simulations with neuromorphic testcase) Efficiency enhancement compared to SpiNNaker1: 11.8x
Peak throughput MAC per second	0.10G	0.25G	25.60G
Neuromorphic Benchmark Example			
Relative time per 1 Nengo time Step Update (relative), 100 input dimensions	1	0.4	<0.1 Performance enhancement compared to SpiNNaker1: >10x

The Neural Engineering Framework (NEF) uses vector-matrix multiplication in the encoding phase, which can be accelerated by the machine learning accelerator in the SpiNNaker 2 prototype. To demonstrate the speedup, a neural network of 128 neurons is simulated on one PE in the FPGA prototype (Nengo-NEF for SpiNNaker (Component id: C1873)). For comparison, the same neuron type and the same firing rate is used as previously reported in SpiNNaker 1 [3]. The total number of clock cycles for one-time step in SpiNNaker 1 and SpiNNaker 2 prototype is compared. As the dimension increases, the speedup factor brought by the machine learning accelerator also increases, and for high dimensions, a speedup factor of more than 10 can be achieved, as shown in Figure 15:

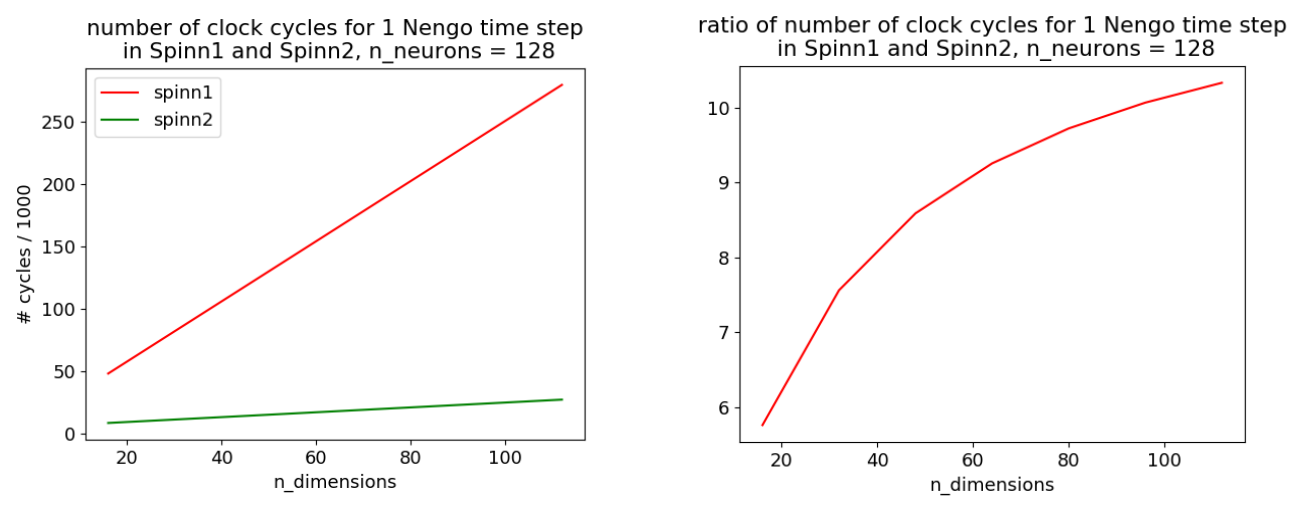

Figure 15: speedup factor brought by SpiNNaker-2 machine learning accelerator with increasing dimensions

As key result 10-fold improved energy efficiency and performance could be demonstrated.

References:

[2] E. Stromatias, F. Galluppi, C. Patterson, and S. Furber, “Power analysis of large-scale, real-time neural networks on SpiNNaker,” in Neural Networks (IJCNN), The 2013 International Joint Conference on, Aug 2013, pp. 1-8.

[3] Mundy, A., Knight, J., Stewart, T. C., & Furber, S. (2015, July). An efficient SpiNNaker implementation of the neural engineering framework. IJCNN 2015.

6.2 Validation and Impact

6.2.1 Actual Use of Output(s) / Exploitation

The primary use of the SpiNNaker-2 prototype devices is to reduce the technology risk of the (expensive) final SpiNNaker-2 chip fabrication. A secondary use is to support software and demonstrator application development ahead of the availability of the final chip.

6.2.2 Potential Use of Output(s)

The Output could be used to build small-scale SpiNNaker-2 systems ahead of the availability of the final chip.

6.2.3 Publications

[1] Dynamic Power Management for Neuromorphic Many-Core Systems, Sebastian Hoeppner, Bernhard Vogginger, Yexin Yan, Andreas Dixius, Stefan Scholze, Johannes Partzsch, Felix Neumaerker, Stephan Hartmann, Stefan Schiefer, Georg Ellguth, Love Cederstroem, Luis Plana, Jim Garside, Steve Furber, Christian Mayr; arXiv:1903.08941 (HBP publication P1793)

6.2.4 Measures to Increase Impact of Output(s): Dissemination

The SpiNNaker-2 chip design was presented and described at a number of venues, including the NICE 2018 workshop in Portland, Oregon.

7. Key Result KR9.5: Two novel theories of computational principles: Learning-to-learn (L2L) and network learning based on dendritic computation

7.1 Outputs

7.1.1 Overview of Outputs

In the first Output, we show that spiking networks can learn fast without synaptic plasticity, using the recurrent dynamics of the network through learning to learn (L2L). In addition, L2L allows us to install prior knowledge into the networks, which aids in this fast learning.

In the second Output, we developed a coincidence-dependent plasticity rule for layer-5 pyramidal neurons that is well supported by biological data and has a clear theoretical interpretation as a biological approximation to logistic regression. Using this rule, the L5P neurons learn to predict the somatic firing on the basis of inputs to the distal dendritic compartments.

7.1.2 Principle 1: Learning to learn with networks of spiking neurons

Recurrent Spiking Neural Networks (RSNNs) in the brain have been optimised through a host of preceding processes, from evolution to prior learning of related tasks, for their learning

performance. We emulate this using the L2L setup. It has been argued (Wang *et al*. 2018) that the pre-frontal cortex (PFC) accumulates knowledge during fast, reward-based learning in its short-term memory, without using dopamine-gated synaptic plasticity. In (Perich *et al*. 2018) a prominent role of short-term memory for fast learning in the motor cortex was suggested as well.

The setup of L2L involves an infinitely large family F of learning tasks C. Learning is carried out simultaneously in two loops (see Figure 16 1A). The inner loop learning involves the learning of a single task C by a neural network N, in our case by an LSNN (= RSNN that also contains adapting neurons). Some parameters of N (termed hyper-parameters) are optimized in an outer loop optimization to encourage fast learning of a randomly drawn task C from F. The outer loop training - implemented here through backpropagation through time (BPTT) - proceeds on a much larger time scale than the inner loop, integrating performance evaluations from many different tasks C of the family F. One can interpret this outer loop training as mimicking longer term processes, as well as prior learning, in brain networks. All synaptic weights of N are hyper-parameters, optimized only through the outer loop. Hence the network is forced to encode all results from learning the current task C in its internal state. Thus, the synaptic weights of the neural network N are free to encode an efficient algorithm for learning arbitrary tasks C from F.

In (Bellec *et al*. 2018) we considered the task of learning complex non-linear functions from a teacher. Specifically, we chose the family F of tasks to be the family of all functions that can be computed by a 2-layer artificial neural network of sigmoidal neurons (the Target Network, TN) with two inputs, 10 neurons in the hidden layer, and weights and biases from [-1, 1], see Figure 16 B.

After a few thousand training iterations in the outer loop, the LSNN achieves low MSE for learning new TNs from the family F, significantly surpassing the performance of an optimal linear approximator (Figure 16 C). The LSNN was able to learn a TN with substantially fewer trials than the best learning algorithm for learning the TN directly in an artificial neural network as in Fig. 1A: backpropagation with a prior that favored small weights and biases (Figure 16 E,F).

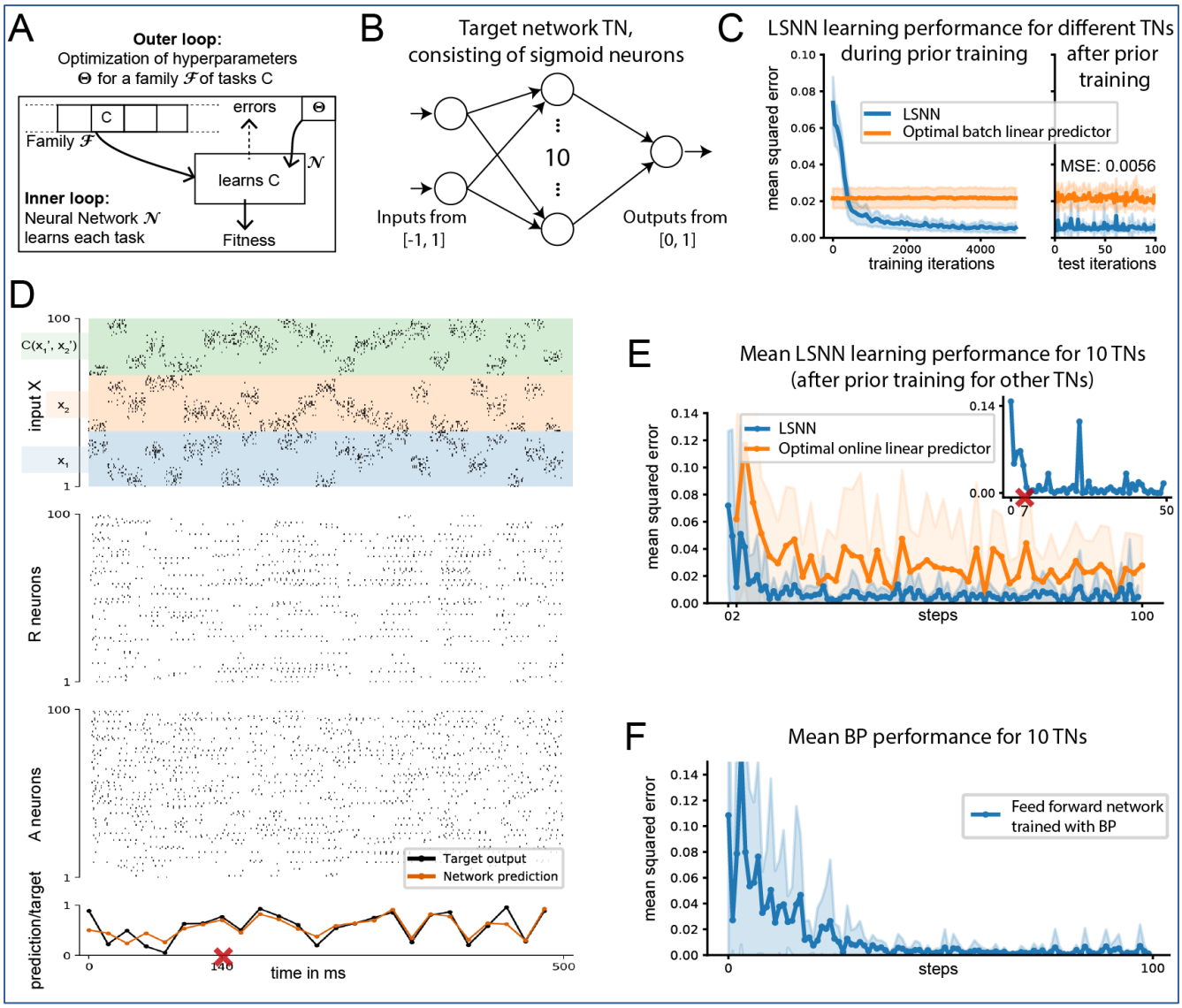

Figure 16: LSNNs learn to learn from a teacher

A) L2L scheme for an SNN N. B) Architecture of the two-layer feed-forward target networks (TNs) used to generate nonlinear functions for the LSNN to learn C) Performance of the LSNN in learning a new TN during (left) and after (right) training in the outer loop of L2L. Performance is compared to that of an optimal linear predictor fitted to the batch of all 500 experiments for a TN. D) Network input (top row, only 100 of 300 neurons shown), internal spike-based processing with low firing rates in the populations R and A (middle rows), and network output (bottom row) for 25 trials of 20 ms each. E) Learning performance of the LSNN for 10 new TNs. Performance for a single TN is shown as insert, a red cross marks step 7 after which output predictions became very good for this TN. The spike raster for this learning process is the one depicted in C. Performance is compared to that of an optimal linear predictor is fitted to the batch of all preceding examples. F) Learning performance of BP for the same 10 TNs as in D, working directly on the ANN from A, with a prior for small weights.

References:

(Bellec *et al*. 2018) Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., and Maass, W. (2018). Long short-term memory and Learning-to-learn in networks of spiking neurons. In Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds. (Curran Associates, Inc.), pp. 795-805. (HBP publication P1449)

(Wang *et al*. 2018) Wang, J.X., Kurth-Nelson, Z., Kumaran, D., Tirumala, D., Soyer, H., Leibo, J.Z., Hassabis, D., and Botvinick, M. (2018). Prefrontal cortex as a meta-reinforcement learning system. Nature Neuroscience 21, 860-868.

(Perich *et al*. 2018) Perich, M.G., Gallego, J.A., and Miller, L. (2018). A Neural Population Mechanism for Rapid Learning. Neuron 964-976.e7.

7.1.3 Principle 2: Supervised learning using Ca+2 Spikes in Layer 5 Pyramidal Neuron

Experimental studies on Layer 5 pyramidal (L5P) neurons show that, when a somatic spike coincides within some time window with distal dendritic input that triggers NMDA spikes, they together trigger a long-lasting calcium spike in the dendrite which causes a burst of spikes in the soma [1, 2, 3]. Thus, the L5P neuron acts as a "coincidence detector". There is further experimental evidence that these calcium spikes play an important role in the synaptic plasticity at distal dendrites [4].

We have developed a simple model of these coincidence detection dynamics, along with a theoretically grounded plasticity rule, gated by coincidence triggered calcium spikes. This rule, called Spike-based Logistic Regression (SLR), is both functional and matches experimental data. It also has a clear theoretical interpretation as a biological approximation to logistic regression.

Our simplified multi-compartment model (Figure 17) consists of somatic, apical dendritic and distal dendritic compartments, where sodium, calcium and NMDA spikes respectively are triggered. The spike rate of the NMDA compartment is given by $\rho(t) = \sigma\left(\mathbf{w}(t)^T\mathbf{x}(t)\right)$, where $\mathbf{w}$ is the weight vector of the incoming synapses incident on the distal dendritic compartment, and $\mathbf{x}(t)$ is the input to the dendritic compartment. The calcium spikes are triggered when somatic and NMDA spikes occur within τ_c ms of each other, and extend over an interval of length τ_{Ca}

Activity in the distal apical dendrite in the absence of a calcium spike leads to synaptic depression [4, 6]. In contrast, robust potentiation is observed in the presence of a calcium spike [7]. Thus, upon each NMDA Spike, we perform the following weight update:

$$\Delta\mathbf{w} = \left(\frac{z(t)}{\sigma\left(\mathbf{w}(t)^T\mathbf{x}(t)\right)} - 1\right)\mathbf{x}(t)$$

Here, z(t) is 1 if a calcium spike is active at time t and 0 otherwise. We demonstrate that the above rule performs stochastic gradient descent on a logistic regression loss with the targets represented by the somatic firing rate.

As a first test, we used it to classify MNIST digits, in which we achieved a performance of 12% Error (matches best performance for a linear classifier [8]). Secondly, we constructed a spiking auto-encoder for MNIST digits with this model as the decoder. Encoding was done using the model described in [5]. Random examples of reconstructed digits are shown in Figure 18.

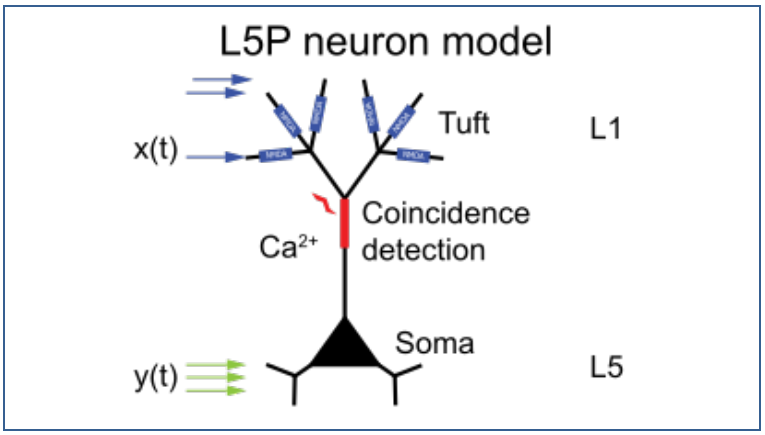

Figure 17: Neuron showing the soma, apical (for Ca2+Spikes), and tuft (with NMDA spikes) compartments.

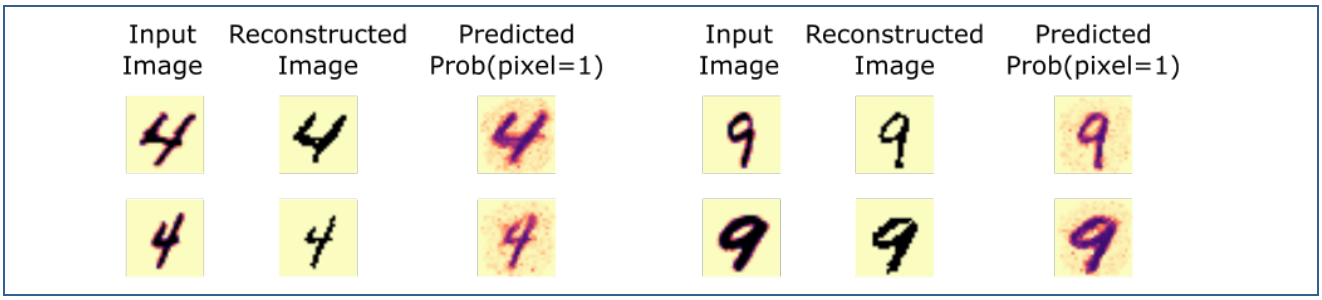

Figure 18: Samples of MNIST digits reconstructed from a sparse encoding using WTAs.

Each Pixel is reconstructed by one L5P Neuron based on its dendritic activity.

References

1) Larkum, M. Trends in Neurosciences 36, 141-151. ISSN: 0166-2236, 1878-108X (Mar. 2013).
2) Major, G., Larkum, M. E. & Schiller, J. Annual Review of Neuroscience 36, 1-24 (2013).
3) Larkum, M. E. & Zhu, J. J. Journal of neuroscience 22, 6991-7005 (2002).
4) Kampa, B. M., Letzkus, J. J. & Stuart, G. J. Trends in Neurosciences 30, 456-463. ISSN: 0166-2236 (Sept.2007).
5) Nessler, B., Pfeiffer, M., Buesing, L. & Maass, W. PLOS Computational Biology 9, e1003037.1553-7358 (Apr. 2013).
6) Sjöström, P. J. & Husser, M. Neuron 51, 227-238. ISSN: 0896-6273 (July 2006).
7) Kampa, B. M., Letzkus, J. J. & Stuart, G. J. The Journal of physiology 574, 283-290 (2006).
8) LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Proceedings of the IEEE 86, 2278-2324 (1998).

7.2 Validation and Impact

7.2.1 Actual Use of Output(s) / Exploitation

None yet.

7.2.2 Potential Use of Output(s)

Both Outputs are suitable for implementation on neuromorphic hardware such as BrainScaleS and SpiNNaker, since CDLR is a local, event-based plasticity rule, while the L2L inner loop learning doesn't require plasticity for learning.

7.2.3 Publications

(Bellec *et al.* 2018) Bellec, G., Salaj, D., Subramoney, A., Legenstein, R., and Maass, W. (2018). Long short-term memory and Learning-to-learn in networks of spiking neurons. In Advances in Neural Information Processing Systems 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds. (Curran Associates, Inc.), pp. 795-805. (HBP publication P1449)

7.2.4 Measures to Increase Impact of Output(s): Dissemination

Conference attendance, e.g. NIPS 31

8. Key Result KR9.6: Applications exploiting the new features of second generation systems (Michael Schmuker)

8.1 Outputs

8.1.1 Overview of Outputs

Applications on SpiNNaker 2

The following applications and benchmarks are running on either the current SpiNNaker2 silicon prototype (Santos) or on the FPGA prototype:

- Deep rewiring
- NEF framework for robotic control
- Neuromorphic power management
- DVS sensor connectivity with power management

Each exploits two or more of the 2nd generation SpiNNaker features, i.e. neuromorphic power management, the floating point unit, accelerated exponentials, random number generators and the multiply-accumulate array for deep neural networks (DNN).

Applications on BrainScales 2

The following applications and benchmarks have been implemented on BrainScaleS-2 prototype system:

- On-chip reinforcement learning benchmarks: Information-theory learning rules for supervised learning
- Neuroscience-inspired structural plasticity to optimise synaptic connectivity
- Expectation-maximisation for unsupervised learning.

8.1.2 SpiNNaker 2

Deep rewiring

Spike-based synaptic sampling and its DNN equivalent, deep rewiring, have been mapped to the prototype. It requires the floating point unit on SpiNNaker2 and uses the exp and random generators for increased efficiency. This plasticity rule shows biological features (rewiring around lesions) are on a par with backpropagation in its DNN version and have very little memory usage. It demonstrates the state of the art in small embedded neuromorphic devices [Liu 2018, Yan 2019].

Neural Engineering Framework for robot control

In cooperation with Chris Eliasmith at the University of Waterloo, we ported the neural engineering framework (NEF) to the FPGA prototype. NEF on SpiNNaker2 opens a range of applications, from robotics to large scale brain models. We are currently running the same NEF robotics controller as the one shown on Intel's Loihi, aiming at side-by-side comparison of the platforms (with a paper in preparation; see robotic arm prototype in Fig. 19). NEF uses the multiply accumulate arrays on the prototypes, significantly improving network capacity compared to SpiNNaker1.

Figure 19: Robot control setup

Neuromorphic power management and event-based vision

In cooperation with Germain Haessig/Giacomo Indiveri at INI Zurich, we connected a dynamic vision sensor to the Santos prototype. A framework for centre-of-mass tracking/prediction is under construction that uses neuromorphic power management to run the processors (and thus power consumption) proportional to the spike activity coming in from the vision sensor (Figure 20).

The neuromorphic power management on the prototypes has been tested on a variety of networks: asynchronous irregular, bursting, synfire chain [Hoeppner 2019].

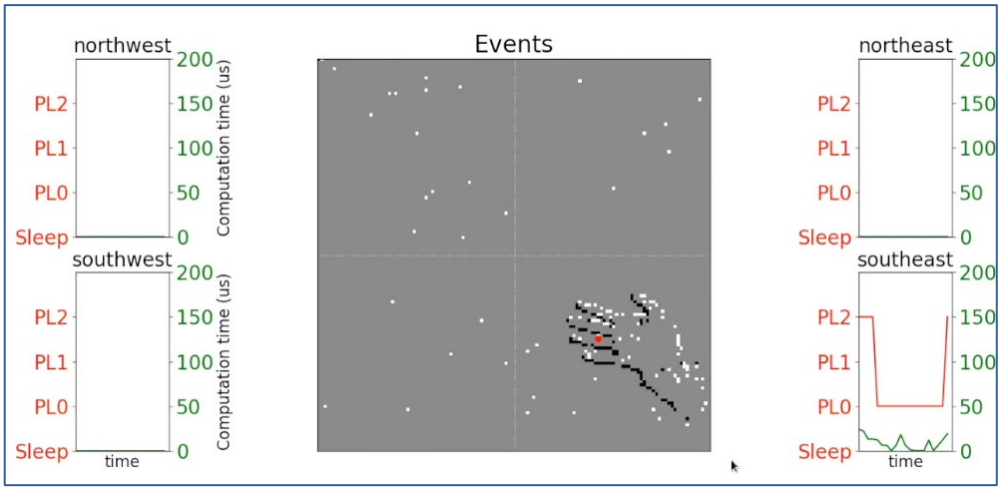

Figure 20: Power-status of processors assigned to the 4 quadrants of the image

Processors with low spiking activity in their quadrant of the image use "sleep" and thus consume much less energy than the processor for the active quadrant.

Assessment against state of the art

SpiNNaker-2's main competitors are Intel's Loihi in the domain of spiking accelerators, and GPUs and other deep neural network accelerators e.g. by NVIDIA. For both domains, SpiNNaker-2 is level with the current state of the art in terms of energy efficiency, while offering more flexibility and programmability than the competition. Publications that back these claims are planned for later this year.

8.1.3 BrainScaleS-2 Prototype System

On-chip reinforcement learning

We used a BSS-2 prototype to solve reinforcement learning problems using spiking neurons [Bohnstingl 2019], while applying the "Learning to learn" framework [Bellec 2018]. Accelerated spiking network emulations produce solutions for Markov decision processes (MDP) and agents solving the multi-armed bandit problem (MAB). The plasticity processor on the BSS-2 prototype system computes accelerated network learning. In the case of MAB, it also simulates the environment in which the agents operate.

Another demonstrator of reinforcement learning implements a simulation of the Pong video game, where a two-layer spiking network learns to control the racket using Reward-modulated Spike-Timing-Dependent Plasticity (R-STDP). A reward is provided in accordance with the aiming accuracy of the neural network (Figure 21). This demonstrator was shown at the Capocaccia workshop 2018 and published [Wunderlich 2019].

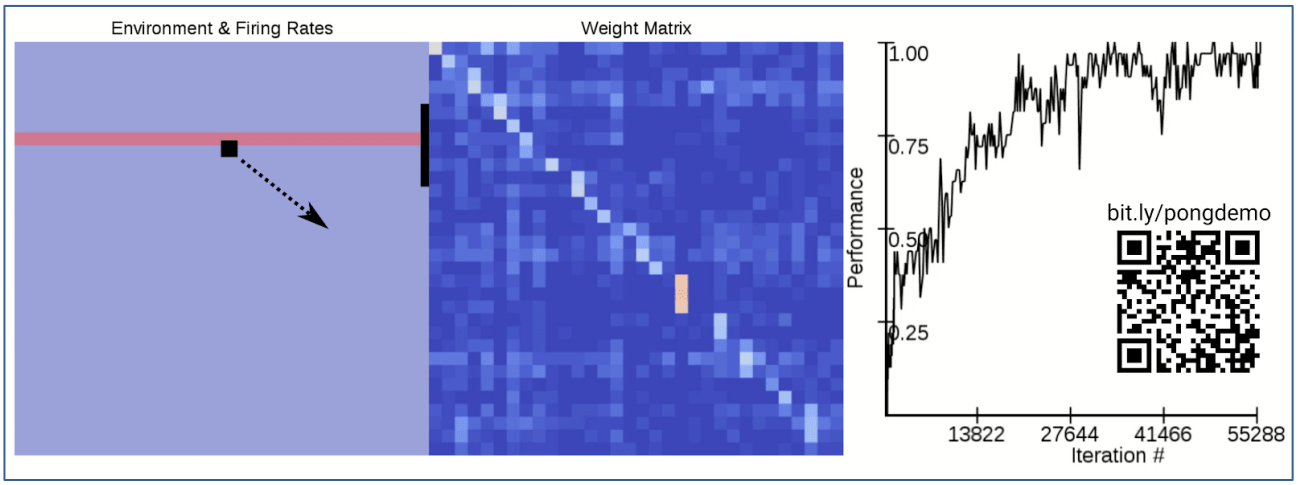

Figure 21: Playing field and Weight Matrix after 5k iterations.

Right: Performance improves with iteration count. See https://youtu.be/LW0Y5SSIQU4.

Information-theoretic learning rules on BrainScaleS-2 prototype system

During SGA1, WP4.3 (PI Grüning, SURREY) developed biologically plausible learning rules for spiking networks based on information theory (e.g. INST/FILT1). In SGA2, they implemented them on the plasticity processor. They successfully solved the Iris and Wisconsin breast cancer benchmarks in software simulations. Success on MNIST was limited by the restriction to 32 neurons on the prototype.

Structural plasticity

Structural plasticity (SP) describes the biological process of constant rewiring of synaptic connectivity to reduce spatial and energetic footprint by limiting the number of synapses. In neuromorphic systems, SP can mitigate effects of limited synapse availability. We implemented an SP algorithm on a BrainScaleS-2 prototype, selecting suitable synapses out of a set of potential

1 doi.org/10.1371/journal.pone.0161335

connections to achieve good performance in the Iris classification task, while maintaining connectome sparseness (Figure 22). Furthermore, SP learned the information content of different filter bank outputs in an auditory digit recognition task (Figure 22 B).

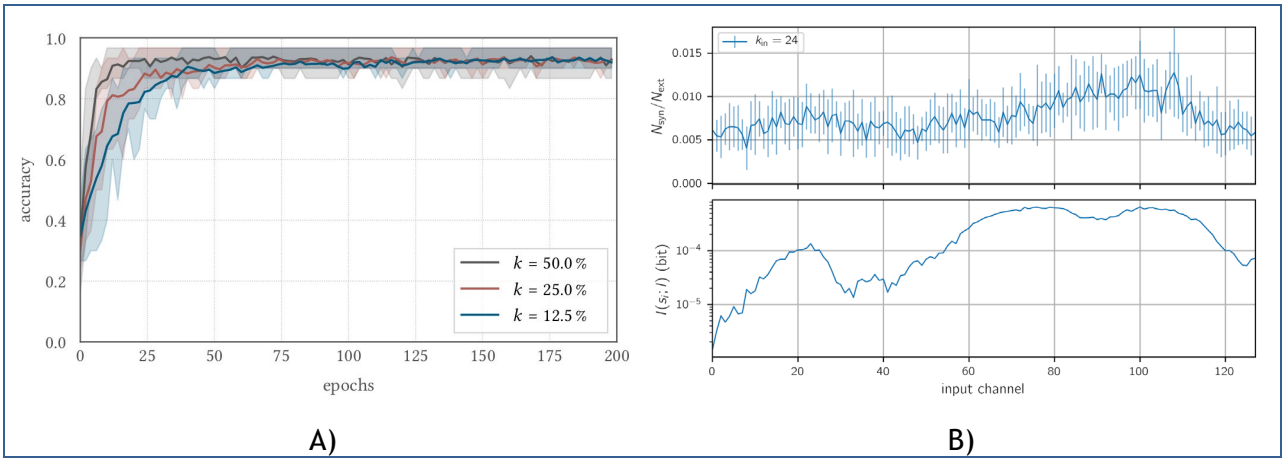

Figure 22: Structural plasticity example

Left: SP algorithm maintains Iris classification accuracy while reducing the number of synapses. Right: The distribution of realized synapses to auditory inputs (top) matches the mutual information between input channels and labels of spoken digits, reflecting a preference for correlated and therefore informative channels.

8.1.4 Spike-based Expectation Maximisation

We implemented unsupervised learning on-chip performing spike-based expectation maximisation (SEM). Images of three digits are presented in a random manner to a network of three neurons that are connected in a winner-take-all fashion (Figure 22). Input connections are subject to a neuromorphic SEM learning rule. The full experiment is described in [Spilger 2018].

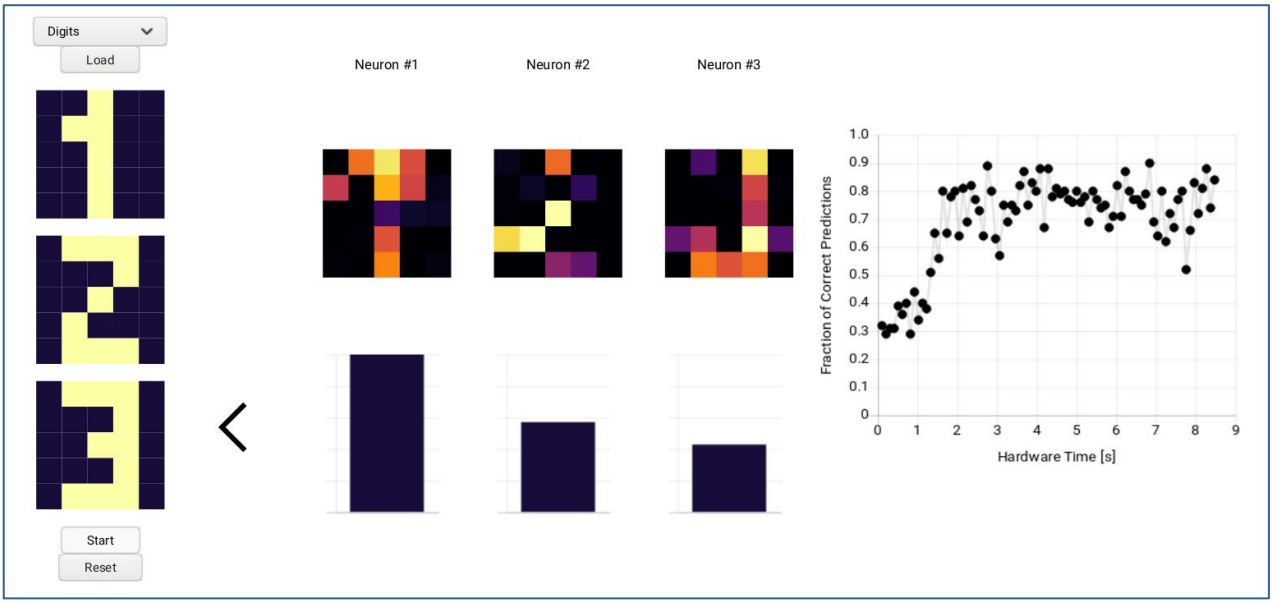

Figure 23: Unsupervised on-chip learning experiment

Input pattern (left), receptive fields & spike-rate (middle), classification performance (right). Video at https://youtu.be/x3l1xl8orhQ.

8.2 Validation and Impact

Applications of the 2nd generation neuromorphic system features serve as demonstrators for the technology, showcasing scenarios where users can benefit from the new technology.

For SpiNNaker 2, deep rewiring enables accelerated low-power deep learning and inference through sparse matrix support, and low power consumption of event-based hardware potentially enables new application scenarios in mobile computing and for biological implants.

For the BrainScaleS-2 prototype, the on-chip plasticity processor enables complex learning rules that go beyond classical STDP supporting bio-inspired AI. Examples are: 3rd factor learning rules like reward-based learning rules, homeostasis and structural plasticity. It also allows other aspects of computational problems to be simulated alongside the accelerated spiking emulation. Combined with the high speedup factor, this potentially enables solution of far more complex computational problems than is possible with the state of the art.

8.2.1 Actual Use of Output(s)/ Exploitation

Since the 2nd generation systems are still in the prototype stage, and the completion of those systems is currently planned for the beginning of SGA3, it is too early to expect a huge uptake of the technology outside the neuromorphic groups within the HBP. However, it should be noted that extra-HBP collaborations are underway, such as the collaboration with the Eliasmith lab on SpiNNaker 2.

8.2.2 Potential Use of Output(s)

The implemented applications are intended as technology demonstrators and their primary impact is in drawing interested users to the platforms and encourage implementations of novel solutions on the hardware, which will generate further scientific, medical or commercial impact down the road.

8.2.3 Publications

[Bohnstingl 2019] Neuromorphic Hardware learns to learn. ArXiv 2019, https://arxiv.org/abs/1903.06493. (HBP publication P1801)

[Bellec 2018] Bellec *et al.*, "Long short-term memory and Learning-to-learn in networks of spiking neurons." Proceedings of NIPS 2018, http://papers.nips.cc/paper/by-source-2018-442. (HBP publication P1449)

[Hoeppner 2019] Hoeppner *et al.*, "Dynamic Power Management for Neuromorphic Many-Core Systems". Arxiv 2019, https://arxiv.org/abs/1903.08941 (HBP publication P1793)

[Liu 2018] Liu *et al.*, "Memory-Efficient Deep Learning on a SpiNNaker 2 Prototype". Front. Neurosci. 2018, https://doi.org/10.3389/fnins.2018.00840 (HBP publication P1624)

[Spilger 2018] Philipp Spilger. “Spike-based Expectation Maximization on the HICANN-DLSv2 Neuromorphic Chip”. Bachelor thesis. Heidelberg University, 2018 (HBP publication P1803)

[Wunderlich 2019] Wunderlich *et al.*, “Demonstrating Advantages of Neuromorphic Computation: A Pilot Study”, in Frontiers in Neuromorphic Engineering, 2019, doi:10.3389/fnins.2019.00260 (HBP publication P1721)

[Yan 2019] Yan *et al.*, "Efficient Reward-Based Structural Plasticity on a SpiNNaker 2 Prototype". Arxiv 2019, https://arxiv.org/abs/1903.08500 (HBP publication P1804)

8.2.4 Measures to Increase Impact of Output(s): Dissemination

The primary way of disseminating these applications is through publication in peer-reviewed journals and via reputable conferences. Demonstrators based on these applications have been shown at targeted workshops, such as the SpiNNaker 2 and BrainScaleS-2 prototype system demonstrations at the Capocaccia workshop for neuromorphic engineering in 2018. Showcases are already planned for the Capocaccia workshop in 2019. We will target the wider AI community (e.g. NeurIPS conference) via additional publications and dissemination venues, as well as more specific ones targeted on narrower audiences in relevant fields (Frontiers in neuromorphic engineering, Telluride workshop).

9. CDP5 Results: links to KRc5.1, KRc5.2, KRc5.3 and KRc5.4 in Deliverable D9.4.1 (D60.1, D27)

SP9 partners contributed to CDP5 results, as described in the Deliverable D9.4.1 "CDP5-Biological deep learning: Results for SGA2 Year 1", especially under the Key Results KRc5.1: O1, KRc5.2: O1 und KRc5.3: O2, O5. The short summaries for these outputs are copied here from D9.4.1 as reference. For the details please see D9.4.1

9.1 KRc5.1: Output 1: Adaptive closed-loop control in a virtual environment using local reinforcement learning on a BrainScaleS-2 prototype

Timo Wunderlich, Akos Kungl, Eric Müller, Johannes Schemmel, Mihai A. Petrovici

CDP5 collaboration between SP9 (UHEI, P47) and SP4 (UBERN, P71)

For the first time, we demonstrated that a prototype of the BrainScaleS-2 neuromorphic system can be used to implement neuromodulated plasticity to solve a closed-loop learning task in a virtual environment. Our experiments show that the BrainScaleS hardware can be used to evaluate learning processes in an accelerated and energy-efficient fashion.

9.2 KRc5.2: Output 1: Spatio-Temporal Predictions with Spiking Neural Networks

Maximilian Zenk, Dominik Dold, Mihai A. Petrovici

CDP5 collaboration between SP9 (U Heidelberg) and SP4 (U Bern)

In previous work, it has been shown that stochastic spiking neural networks sample from posterior probability distributions parametrized by the network's synaptic connections. So far, this has been used for static inputs only, e.g., to perform classification after training such networks to approximate some underlying data distribution. In this work, we extended the approach to time-continuous problems, allowing the network to act and react in a time-dependent environment. The presented model is portable to the neuromorphic hardware currently developed in the HBP.

9.3 KRc5.2: Output 2: Sequence learning by shaping hidden connectivity

Kristin Völk, Mihai A. Petrovici, Walter Senn

CDP5 collaboration between SP4 (UBERN, P71) and SP9 (UHEI, P47)

A cortical developmental model is suggested for learning spatio-temporal patterns based on 2-compartment neuron models and dendritic plasticity. The model shapes an appropriate connectivity pattern in a pool of hidden neurons that allows the memorization of non-Markovian sequences in visible neurons. The model is portable to the neuromorphic hardware currently developed in the HBP.

9.4 KRc5.3: Output 1: Natural gradient for spiking neurons

Elena Kreutzer, Mihai A. Petrovici, Walter Senn

CDP5 collaboration between SP4 (UBERN, P71) and SP9 (UHEI, P47)

We show that natural-gradient-based learning for spiking neurons predicts counterbalancing of homo-and heterosynaptic plasticity, thereby aligning models of error-correcting synaptic plasticity to experimental evidence. In addition, we demonstrate that in many cases, it can be approximated by a simpler rule, enhancing biological plausibility and facilitating implementation on neuromorphic hardware.

9.5 KRc5.3: Output 2: Lagrangian neurodynamics for real-time error-backpropagation across cortical areas

Dominik Dold, Akos F. Kungl, Joao Sacramento, Mihai A. Petrovici, Walter Senn

CDP5 collaboration between SP4 (UBERN, P71) and SP9 (UHEI, P47)

A major driving force behind the recent achievements of deep learning is the backpropagation-of-errors algorithm (backprop), which solves the credit assignment problem for deep neural networks. Its effectiveness in abstract neural networks notwithstanding, it remains unclear whether backprop represents a viable implementation of cortical plasticity. In CDP5, we developed a new theoretical framework that uses a least-action principle to derive a biologically plausible implementation of backprop.

9.6 KRc5.3: Output 3: Error-driven learning supports Bayes-optimal multisensory integration via conductance-based dendrites

Jakob Jordan, João Sacramento, Mihai A. Petrovici, Walter Senn

CDP5 collaboration between SP4 (UBERN, P71) and SP9 (UHEI, P47)

Animals receive information about their environment through a variety of senses that need to be integrated to form a coherent percept. To combine information from multiple senses meaningfully requires a representation of the reliability of each source. We developed a formal framework that maps such types of probabilistic computations to the biophysical dynamics of multi-compartment neurons with conductances-based synapses.

9.7 KRc5.3: Output 5: Training deep networks with time-to-first-spike coding on the BrainScaleS wafer-scale system

Julian Göltz, Oliver Breitwieser, Sebastian Schmitt, Johannes Schemmel, Mihai A. Petrovici

CDP5 collaboration between SP9 (U Heidelberg) and SP4 (U Bern)

We established a framework that learns to recognize patterns on the BrainScaleS (BSS) using time-to-first-spike coding. Such learning with single spikes promises to be an energy efficient and fast approach to machine learning on neuromorphic hardware.

9.8 KRc5.4: Output 2: Interaction between sleep and memory in a thalamo-cortical model performing visual classification (MNIST)

Cristiano Capone, Elena Pastorelli, Bruno Golosio, Pier Stanislao Paolucci, Maurizio Mattia, Mihai Petrovici, Johannes Schemmel

CDP5 collaboration between SP3 (INFN, P92 and ISS, P96), SP4 (UBERN, P71) and SP9 (UHEI, P47)

For the first time, we demonstrated two beneficial effects of deep-sleep thalamo-cortical oscillations: the creation of categories from examples and a generalized homeostasis of synaptic weights, with an improvement in post-sleep classification rates and a normalization of firing rates during post-sleep wake activity. This mechanism could play an essential role in bio-inspired learning for neurorobotic applications based on neuromorphic computing.

10. Conclusion and Outlook

The first year of SGA2 has gone as planned with concrete progress in all targeted areas of development, from integrating the two first generation systems into the Neuromorphic Computing Platform through to the development of the second generation systems.

The two HBP neuromorphic computing (NMC) systems continue to represent the EU leadership position in NMC despite their increasing venerability - both systems were designed before the start of the HBP in 2013 - and both have given their respective design teams extremely valuable experience in supporting and understanding user requirements. This understanding is feeding directly into the design of the second generation systems, which will compound these benefits with those of more advanced semiconductor technology to deliver much higher capabilities to users within similar physical and power envelopes to the current systems.

The world is increasingly sensing the convergence between neuromorphic technology and conventional AI, which is based upon non-spiking deep and convolutional neural networks. The advantages of spiking networks have yet to be convincingly demonstrated, but the expectation is that they will be manifest in more energy-efficient, event-based systems (since a spike is simply a pure asynchronous event) that have capabilities that exceed those of conventional AI in areas such as one-shot and few-shot learning, continuous on-line learning. The proof that spiking networks have these capabilities is in biological brains, but we have yet to understand these sufficiently to be able to engineer those capabilities into artificial systems. We are seeing real progress in closing this gap here, particularly with the work towards KR9.5. The second year of SGA2 should see more dramatic progress in this direction, preparing the ground for major breakthroughs in SGA3.

Annex: a list of some SP9 "components"

ID	Name
C1	SP9 BrainScaleS 1 Neuromorphic Computing System (version 1 = NM-PM1)
C2	SP9 SpiNNaker Neuromorphic Computing System
C347	SP9 MUSIC library
C349	PyNN
C453	SP9 SpiNNaker next-generation (NM-MC2, SGA1) chip
C454	SP9 BrainScaleS 2 standalone, single-chip, physical model system
C457	SP9 BrainScaleS 2 Neuromorphic Computing System (version 2 = NM-PM2) -- system under development
C1608	SP9 accelerated simplified environment simulation for BrainScaleS closed loop experiments (mainly SGA2)
C1609	SP9 SP4 SP3 software part of the agent (sensors and actors interacting with the accelerated virtual environment) (SGA2)
C1637	Neuromorphic Computing Platform Monitoring Service
C1638	Neuromorphic Computing Platform load-balancing and scaling service
C1640	SP9 Developer and Operations Guidebook
C1655	SP9 Graphical neuromorphic model-building app
C2575	SP9 Tools for analysing NM benchmark runs
C2576	SP9 Benchmark set with representative set of network architectures studied in HBP
C2735	SP9 SGA2 T9.5.4 SNABSuite
C2787	SP9 Software for BrainScaleS Systems (SGA2)
C3042	SP9 Neuromorphic Computing Platform Remote Access Service