2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Optimising the Overall Power Usage on the SpiNNaker Neuromimetic Platform

Evangelos Stromatias, Cameron Patterson and Steve Furber

***Abstract*— Simulations of biological tissue have been extensively used to replicate phenomena observed by in-vivo and in-vitro experiments as an alternative methodology for explaining how computations could take place in a brain region. Additional benefits of simulated neural networks over in-vivo experiments include greater observability, experimental control and reproducibility. General-purpose supercomputers provide the computational power and parallelism required to implement highly complex neural models, but this comes at the expense of high power requirements and communication overheads. Moreover, there are certain cases where real-time simulation performance is a desirable feature, for example in the field of cognitive robotics where embodied agents need to interact with their environment through biologically inspired asynchronous sensors. The SpiNNaker neuromimetic platform is a scalable architecture that has been designed to enable energy-efficient, large-scale simulations of spiking neurons in biological real-time. This work is based on a recent study which revealed that while they are generally energy efficient, SpiNNaker chips dissipate significant amount of power whilst in the idle state. In this paper we perform a systematic investigation into the overall energy consumption of a SpiNNaker system and propose a number of optimised suspend modes in order to reduce this. The proposed implementation is 60% more energy efficient in the idle state, 50% in the uploading and 52% in the downloading phases, while the power dissipation of the whole simulation is reduced by 52%. For demonstration purposes, we run a neural network simulation comprising thousands of neurons and millions of complex synapses on a 48-chip SpiNNaker board, generating millions of synaptic events per second.**

I. Introduction

Computational neuroscientists have been using simulations of biological tissue as an approach to understand how neural circuits work. Models of neural networks, based on anatomical data, aim at reproducing phenomena observed in-vivo and in-vitro experiments. If the simulated network shows similar behaviour, it may then be used to describe how computations take place in that particular region. In addition, computational models provide greater observability and reproducibility. The overall benefit is that a sufficiently accurate model can be simulated repeatedly and in high-fidelity, without the noise of a biological recording and at whichever level of detail is required. For these reasons, large-scale simulations of biological tissue are considered promising tools in the challenge of understanding just how the brain functions [1].

In the past, large-scale simulations of spiking neural networks have been successfully executed on general-purpose supercomputers [2], [3], [4]. However, while supercomputers offer signicant parallelism and great opportunity for model exibility, they suffer from their large electrical power demands, which are rarely reported, and from communication bottlenecks when simulating spiking neural networks. Recently, Wong et al. [5] simulated 53×10^{10} neurons with 1.37×10^{14} synapses on a Sequioa - BlueGene supercomputer. The simulation ran 1542 times more slowly than biological real-time and the biggest cost reported was communicating the spikes via MPI messaging. Power dissipation was omitted, but the TOP500 [6] supercomputer list states that the power used by the Sequia - BlueGene/Q is 7,890 kWatts.

There are certain cases where real-time performance of a neural simulation is a desirable feature. Once such example is when cognitive neuroscientists and roboticists would like to test and validate their hypotheses using embodied agents [7], [8] interacting with their environment or by interfacing with biologically inspired sensors [9], [10], [11].

SpiNNaker [12] is a biologically inspired, massively-parallel, scalable computing architecture optimised to simulate very large-scale spiking neural networks in real-time [13]. Each SpiNNaker chip comprises 18 identical low-power fully programmable processors that allow the use of dynamic, arbitrary and heterogeneous models in simulation. Its novel interconnection fabric enables it to cope with the small frequent spiking events, the very same limitations that general-purpose supercomputers struggle with. By connecting SpiNNaker chips and boards together, large machines can be formed, permitting simulations to scale-up seamlessly.

This paper is based on a previous study [14] which investigated the energy consumption of SpiNNaker chips for different stages of neural network simulation. Results revealed that a significant portion of the total power was dissipated during the idle state. The idle state is defined as the power used by the system after booting and while in a state where it is ready to accept a workload. In this work, we systematically investigate which components within a SpiNNaker chip are the most energy-consuming, in order to propose new suspend states that minimise the total energy use. This will have a noticeable impact in the overall energy consumption of larger SpiNNaker machines, especially for the nodes not participating in a simulation. Additional savings are expected during the uploading and downloading phases of a simulation cycle.

We present as contributions an implementation of an improved idle state, one that is 60% more efficient when compared with the current one. Moreover, we show its effect on the different simulation phases and on the average power dissipation. For demonstrating these claims we utilise a

The authors are with the School of Computer Science, The University of Manchester, Manchester, M13 9PL, UK. Email: {evangelos.stromatias, cameron.patterson}@cs.man.ac.uk, steve.furber@manchester.ac.uk.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

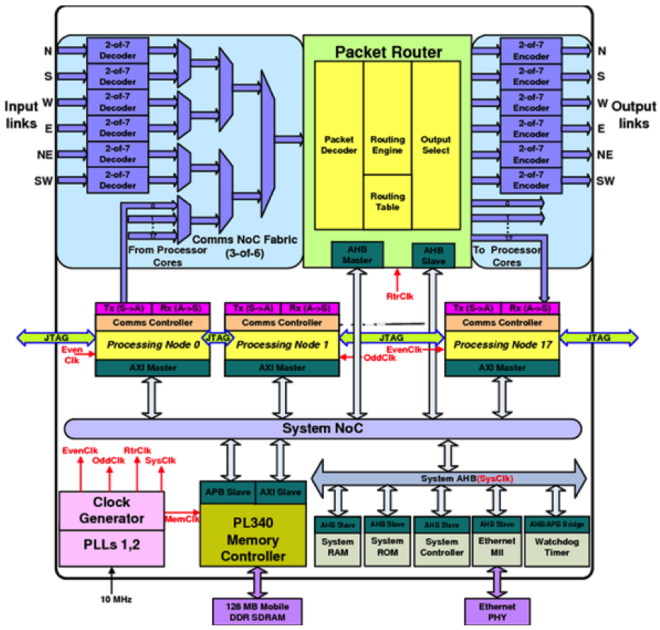

Fig. 1. Block diagram of a SpiNNaker chip.

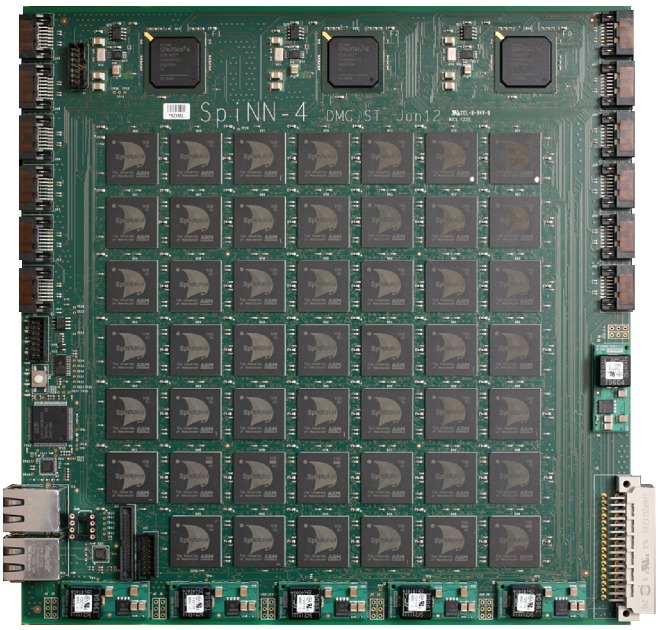

Fig. 2. A 48-node SpiNNaker board.

simulation comprised of thousands of neurons, with millions of complex synapses, generating activity in the millions of synaptic events per second.

II. The SpiNNaker System

A. Hardware

SpiNNaker is an Application-Specific Integrated Circuit (ASIC) designed to enable energy-efficient simulations of heterogeneous models of spiking neurons in real-time [15]. The SpiNNaker chip, Figure 1, which is the building block of the system, consists of 18 homogeneous ARM968 processor cores. Each of these cores has its own 96 KBytes of Tightly-Coupled Memory (TCM) for instructions and data, while they all have access through a self-timed system network on chip (NoC) to a shared 128 MBytes SDRAM where all the relevant synaptic information is stored. At the heart of the SpiNNaker chip lies a packet-based multicast (MC) router, which is responsible for communicating the spikes to its internal cores or to other chips through an asynchronous Communication NoC (Figure 1). Spikes are transmitted as 40 or 72-bit packets implementing the Address-Event Representation (AER) [16] scheme, where the address of the firing neuron is the information transmitted. Each SpiNNaker chip was designed so that each core could simulate up to 1000 real-time neurons, each neuron receiving 1000 connections and generating action potentials with a mean firing rate of 10 Hz [17]. The re-programmability of the SpiNNaker cores however allows different combinations of both the number of neurons and their synaptic connections to be deployed [14].

There are 5 components within a SpiNNaker chip that require a clock source, the ARM968 cores that are divided into two banks based on their physical ID, the router, the System AHB bus and the shared SDRAM memory. Each chip receives a 10 MHz input clock from the Board Management Processor (BMP), that can be used as is, further divided by 4 or used as an input to the 2 Phase-Locked Loop circuits (PLLs). Finally, there is an additional clock divider that can optionally divide the input signal, for each of the 5 clock domains, by 2, 3 or 4. The user can control these parameters through the System Controller registers.

In this study a board with 48 SpiNNaker chips will be used, Figure 2, which is largest system to-date and will be the fundamental component for creating the largest SpiNNaker machines. There are 864 ARM processors available, 768 of which can be utilised for neural applications, 48 as spares for fault-tolerance purposes [18] and 48 for monitoring. The three XILINX Spartan-6 Field Programmable Gate Array (FPGA) chips, found at the top of the board, will be used in the future for board-to-board communications, by taking advantage of high-speed 3.1 Gbps serial interfaces (SATA).

At the lower-right of the board there is a multi-pin connector over which its 12 V DC supply is provided enabling it to be slotted via a frame into the backplane PCB of a multi-board systems. This 12 V supply is distributed to several DC/DC converters which are found just above the multi-pin connector and running along the lower edge of the board (Figure 2). These converters provide the required voltages for the individual components in the system. Three 1.2 V regulators supply batches of 16 SpiNNaker chips, one 1.2 V regulator is provided for the FPGAs and one 1.8 V regulator is used predominantly to supply the SDRAMs but also for input/output purposes of the SpiNNaker and FPGA chips. Finally, a 3.3 V regulator is shared by the BMP, the Ethernet circuitry, the indicator LEDs (1 per chip on the reverse of the board) and once again for the FPGAs.

By connecting multiple SpiNNaker chips together on cir-

cuit boards we form the first level of machine, with boards put together in a rack frame, multiple frames building a rack cabinet and then multiple cabinets forming the largest of machines. The final SpiNNaker machine will comprise approximately 57,600 SpiNNaker chips (over a million ARM processors) aiming to simulate a billion spiking neurons with trillions of synapses in real-time.

B. Software

The SpiNNaker software is divided into two parts. The software running on the SpiNNaker chips and the software running on the host.

1) SpiNNaker side: In each SpiNNaker chip, one core is dedicated for monitoring purposes and some of its responsibilities involve system-wide inter-processor communication, application support and system monitoring. The remaining cores that can be used for neural applications, run a SpiNNaker Run-time Kernel (SARK), which controls the flow of execution and schedules/dispatches application functions when appropriate. On top of SARK, a C-based Application Programming Interface (API) [19] operates that allows users to write call-back functions that respond to events, (Table I), abstracting the hardware complexity. If a core completes the execution of all its scheduled call-backs, and no further events are scheduled, it enters into a power-saving idle mode.

TABLE I
CALL-BACK FUNCTIONS FOR THE NEURAL APPLICATION CORES.

Event	Description
Timer	Neural Equations are solved
Packet Received	Synaptic information is retrieved from SDRAM
DMA Done	Synaptic information is integrated in the neural state

2) Host side: On the host side, which can be a general-purpose computer, PyNN [20] is used to describe a neural network topology. PyNN is a high-level simulator-independent specification language used for building large-scale neural network models using abstractions such as populations and projections. A tool named Partitioning And Configuration MANagement (PACMAN) [21] is responsible for mapping a PyNN description to a SpiNNaker machine based on the available resources.

III. EXPERIMENTAL SETUP

A. Measuring the Power Dissipation

For our experiments we use a SpiNN-4 multi-layer printed circuit board as our fundamental block from which we measure the system (Figure 3). Each SpiNN-4 board has 48 SpiNNaker chips plus ancillary components, and this board configuration is the building block from which larger machines are constructed.

As previously described, the 48-chip SpiNNaker board has 6 DC/DC converters that supply power to the on-board components. In order to measure the power dissipation of the board, a number of shunt resistors were placed in series with the 6 DC/DC converters: 0.03 Ω resistors were placed in series with the 1.2 V regulators (A, B, and C in Figure 3) that supply the SpiNNaker chips. In addition, 0.1 Ω shunt resistors were placed in series with the 1.8 V regulator that supplies the SDRAMs and the inputs/outputs of the chips, the 1.2 V regulator that supplies the 3 FPGA chips, the 3.3 V regulator that supplies voltage to the BMP, the Ethernet circuitry, the indicator LEDs and the FPGAs, and finally with the 12 V supply to the board.

Tenma 72-7750 and Fluke 77 multimeters are used to measure the voltage drop across the shunt resistors, which is proportional to the current flow at that regulator. The 12 V measurement point, as well as an overall measure of the power consumed by the board, serves predominantly as a check and balance, where the heat generated by the shunt resistors and the efficiency of the DC/DC convertors are taken into account.

We also employed two additional measures to assist in the verification process. Firstly, we used a bench meter to validate the overall power supplied to the board based on the load. Secondly, and separately, we used a Model 2000MU-UK Wattmeter connected in-line with the mains supply and before the switched-mode AC/DC adaptor. This meter displays a second by second integrated display of the power passing through it, and therefore gives an overall power rate including all losses in all transformers, shunts and the SpiNNaker board's consumption. While we did not anticipate this would be particularly accurate, by using a straight line 80% efficiency factor for the 12 V AC/DC converter in use, the meter tended to be within a 1 W or 2 W window of the measurements calculated using the more accurate calibrated equipment.

B. Power Profiling SpiNNaker

Parameterised software was developed using the SpiNNaker API software and version 1.09 of SARK. This software made the necessary changes to the SpiNNaker hardware directly, such as peripherals and clocks, resulting in a steady-state environment where accurate and systematic calculations of energy consumption can be made. After the experiment, the configuration reverts to the standard operating parameters for an idle SpiNNaker system, permitting a direct comparison to be made between the new suspend mode under test and the existing software.

Whilst there are chip-level components which may be individually enabled and disabled including some of the controllers, the router, the PLLs and the individual processor blocks, the 'low-hanging fruit' in the experiments was expected to be the dynamic power used by the clocked components. Frequency scaling adjustments should have a big impact on energy use when compared to other components, particularly those that are asynchronous.

At present the run-time software kernel derives all its clock domains from PLL1 set at 400 MHz with the exception of the memory, which PLL2 drives at 266 MHz. Processor domains A and B each divide the incoming 400 MHz by two for a 200 MHz clock, and both the router and System AHB bus divide it by four to supply 100 MHz. By scaling these clocks dynamically while in idle mode, it is expected that the largest

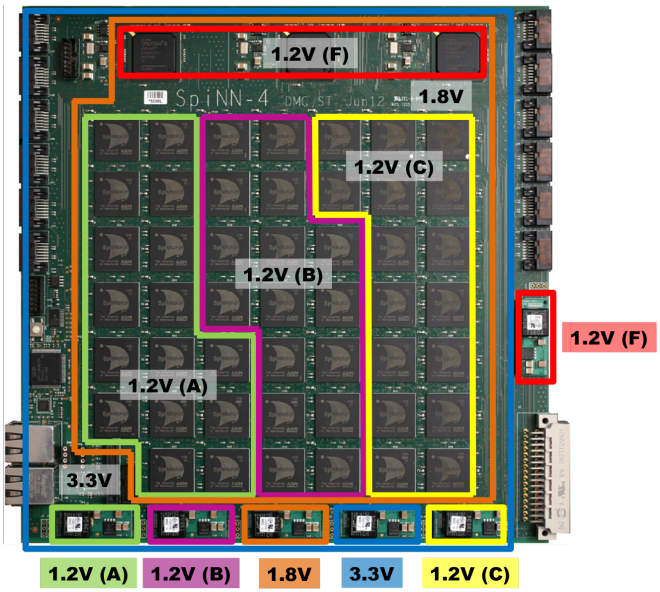

Fig. 3. Power Distribution of a 48-node SpiNNaker board.

savings can be made, even to the extent of shutting down a particular PLL to totally remove the clock from targeted domains. Extreme measures such as these, however, may have adverse impact on the recoverability of the component, for example you cannot remotely command a chip to exit suspend mode if all its processors are de-clocked and cannot respond.

As indicated above, these experiments will concentrate on the dynamic power of the system and in reducing this to a minimum in both recoverable and non-recoverable states. Where it has been possible to characterise particular peripherals and components this will be reported in the results section, so that the maximum potential of the proposed suspend modes can be ascertained.

C. Benchmark Neural Network Topology

A neural simulation will run on SpiNNaker in order to investigate the power dissipation during the three simulation phases: uploading the simulation data, running the simulation and fetching the results. This neural network was designed specifically to stress the intra-chip communications and explore the practical upper-bounds of a 48-chip SpiNNaker board [14]. The neuron model selected for this network is the leaky integrate and fire (LIF) model with current-based exponential synapses, described in the following section.

1) The Leaky Integrate-and-Fire (LIF) Neuron Model: This model captures the fundamental dynamics of biological neurons and represents the cell membrane as a single constant leak conductance and an input current, equation 1.

$$\tau_m \frac{dV}{dt} = E_L - V + R_m I(t) \tag{1}$$

Where I is the input synaptic current, described by equation 2, R_m is the membrane resistance, V is the membrane voltage, E_L is the resting potential and τ_m is the membrane time constant.

An action potential is generated each time the membrane potential V reaches a predefined threshold V_{th} and V is then reset to V_{reset}. An additional parameter, known as the refractory period T_{refrac} simulates the inactivation of ionic channels immediately after a spike has been fired and does not allow a second spike for that period of time.

2) The Synapse Model: In chemical synapses, when an action potential from a presynaptic neuron arrives at the synaptic terminal it causes neurotransmitter release into the synaptic cleft. The neurotransmitter binds to receptors found in the postsynaptic cell and this may initiate an electrical response. If the aforementioned response causes a depolarization in the postsynaptic membrane potential it is called Excitatory Postsynaptic Potential (EPSP), otherwise it is called Inhibitory Postsynaptic Potential (IPSP). In this study current-based instantaneous rise and single-exponential decay synapse models [22] are employed, as described by equations 2 and 3.

$$I(t) = I_{injected}(t) + I_E(t) + I_I(t) \tag{2}$$

Where $I_{injected}$ is the current injected directly to the membrane of the neuron using an electrode and the I_E and I_I terms account for the excitatory and inhibitory currents as described by equation 3.

$$I_{E/I}(t) = \begin{cases} \bar{w} \cdot exp(-\dfrac{t - t_0}{\tau_{E/I}}) & \text{for } t \geq t_0 \\ 0 & \text{for } t < t_0 \end{cases} \tag{3}$$

Where $\bar{w}$ is the peak current amplitude (weight) of the synapse, E or I subscripts represent the excitatory and inhibitory post synaptic currents (PSP). The $\tau_{E/I}$ represents the decay time of the excitatory/inhibitory synaptic currents. The full set of neural and synapse parameters used in the experiments can be found in Table II.

TABLE II
NEURAL AND SYNAPTIC PARAMETERS USED IN THE EXPERIMENTS. VALUES INSIDE THE BRACKETS INDICATE THE RANGE OF RANDOMLY GENERATED VARIABLES BASED ON A UNIFORM DISTRIBUTION. THE RANDOM SEED IS KEPT CONSTANT THROUGHOUT THE EXPERIMENTS.

Parameters	Values	Units
τ_m	64.0	mV
V_{init}	$[-65.0, -125.0]$	mV
V_{reset}	$[-90.0, -125.0]$	mV
V_{thres}	$[-50.0, -60.0]$	mV
$\tau_{I/E}$	10	ms
$\tau_{refract}$	3	ms

3) Benchmark Neural Network Topology: The benchmark neural network consists of a population of neurons where each population resides on a single core, recurrently connected in an all-to-all fashion (Figure 4). Every time a neuron generates an action potential (MC packet) the router redirects it back to the originating core, triggering a packet received

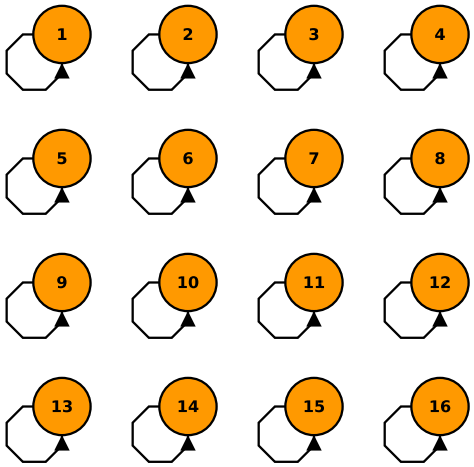

Fig. 4. Topology for the benchmark neural network.

event. As a consequence, a lookup process is initiated, which requests a memory transfer of the relevant synaptic information from the chip's SDRAM. The populations are replicated across the 768 cores (48 chips).

The activity of the network is controlled through two parameters, the injected current $I_{injected}$ and the number of neurons per population, since the synapses increase quadratically with the number of neurons. All synaptic weights are set to zero and since all the processing steps required to handle an incoming spike are the same, regardless of the value of the weight, this technique enables full control of the network dynamics through the current injected to the neurons, $I_{injected}$.

4) Monitoring of the Simulation: In order to verify the fidelity of results and the status of the experiment, extra information is recorded per core during the neural simulation. At the beginning of each timer event a counter stores the cumulative difference between the total MC packets received and DMA Done events. This way it guarantees that all spikes are processed in the correct timer interrupt; if a core is busy it might service a spike in the next timer interrupt. Moreover, an additional counter is incremented each time a neuron fires a spike, so that the total spikes generated per population can be read at the end of the simulation.

IV. Results

A. Measuring the Default Idle State of SpiNNaker

The first step towards optimising the total energy consumption is to measure the default idle state, which will serve as a baseline for remaining experiments. The experimentation is carried out on a single SpiNN-4 board by systematically adjusting a single parameter at a time to ensure that the characteristic information on that component can be gathered. Where it is not possible to alter a single variable at a time, such as where there are combinatorial limitations in PLL assignment, a control experiment is undertaken so that the desired power information can be deduced and recorded.

Using the current SARK software in the idle state, the following table records the power recorded at each of the DC/DC converters, which supply power to the SpiNN-4 board. The results can be seen in Table III.

TABLE III
POWER DISSIPATION IN THE DEFAULT IDLE STATE.

DC/DC Converter	Measured Power
1v2 Bank A	5.23 W
1v2 Bank B	5.17 W
1v2 Bank C	5.52 W
1v8 SDRAM	0.90 W
3v3 Supply	4.67 W
1v2 FPGA	0.50 W
Total	**21.99 W**

Whilst these numbers do not take into account the losses in the converters, they do indicate that the current idle power budget is around 22 W after the conversions to the various required supply voltages. These numbers are used in the experimentation to evaluate the effectiveness of a proposed optimisation.

B. Power Dissipation of Clocked Components

The first set of experimentation is on the five clock domains of all SpiNNaker chips on a board, and the experiments are devised so that PLL1 is used for controlling the component under test, and the remainder of the domains, which are not on test, are clocked from the alternative PLL2.

These clocking domains are as follows: Processor Block A, Processor Block B, The Router, The System AHB (Bus) and Memory.

In the experiments the configuration is replicated across all chips of a 48-chip, single board machine. As a control PLL2 is set to 260 MHz and divided by two for the router, AHB and processors, with the memory controller receiving the 260 MHz directly. PLL1 is used for the component under test and is set explicitly to 200 MHz for all experiments with the appropriate divider to meet the target frequency, with the exception of the memory experiments where this is exceeded. Where the target clock is 10 MHz or less, it is sourced from the 10 or 10/4 MHz clocks directly and PLL1, although unused, remains switched on and at 200 MHz. The results are summarized in Table IV.

If we consider that the total power recorded for the board in the default state, Table III, is 22 W, it is obvious from Tables III & IV that the majority of the idle dynamic power in the SpiNNaker system is taken by the processor blocks, at approximately 70%. The routers and the system busses which also use the 1.2 V supply account for 10% or so, the memory around 8%, and the remainder by the other supporting hardware on the board. Clearly we should be able to attain the largest gains through manipulation of the processor clocks, but it may be possible to make smaller incremental gains by tweaking chip and processor block components when a chip is idle. The 3.3 V supply accounts for the majority of the remaining consumption and thus should also be explored for energy saving opportunities.

TABLE IV
INVESTIGATING THE POWER CONSUMPTION OF CLOCKED COMPONENTS. DEFAULT VALUES ARE IN BOLD FONT.

Component	330 MHz	260 MHz	200 MHz	100 MHz	50 MHz	10 MHz	2.5 MHz	625 kHz
768 Cores			**15.27 W**	8.09 W	4.43 W	1.14 W	0.38 W	0.20 W
48 Routers			2.57 W	**1.34 W**	0.75 W	0.23 W	0.19 W	0.14 W
48 System AHBs			1.02 W	**0.86 W**	0.75 W	0.59 W	0.43 W	0.36 W
48 SDRAMs	2.36 W	**1.81 W**	1.56 W	0.71 W	0.50 W	0.19 W	0.17 W	0.14 W

C. Proposed States of Operation

This section proposes a number of new SpiNNaker Suspend Modes (SSMs), both recoverable and non-recoverable, with results in Table V including the total percentage of power saved. The following subsections provide a brief explanation of each chip suspend mode and its effects to the routing system of a larger SpiNNaker machine.

1) SSM0 - Operational: This is out of scope for idle power saving, but there is potential in the future to explore this state and seek out frequency scaling strategies to minimise wear out, optimise energy use etc.

2) SSM1 - Wait for Interrupt: This is the default mode of operation for a SpiNNaker core when it is not in state SSM0. Its context is saved and recovery is on a per cycle basis. This is the current idle mode pre- and post-simulation and does not attempt any further energy management. Cores are run at 200 MHz, router and system AHB are at 100 MHz and the memory controller 260 MHz.

3) SSM2 - Suspend With SDRAM: This mode clocks down all processors to the minimum possible frequency 625 kHz (10 MHz /4 /4). The router is clocked to 50 MHz (PLL1 /4) as this provides sufficient bandwidth to cope with a full complement of through external traffic. The System AHB bus is also clocked at 50 MHz in the same way as this provides reliable communications for remote mode change commands. This mode maintains the full memory refresh rate of PLL2 at 260 MHz so that when the processors are restored, full context remains available.

4) SSM3 - Suspend Without SDRAM: This mode is identical to SSM2 but stops PLL2, which is fed to the memory controller for refresh. This mode loses external memory context and requires a reconfiguration of the memory controller on recovery.

5) SSM4 - Node Routing Pass-Through: This mode removes the clock from all clocking domains on a chip (set to a stopped PLL2), except the router which is clocked at 50 MHz (PLL1 /4). This way a SpiNNaker board does not become a black hole in the routing fabric since the routers remain in use, full connectivity remains. All context is lost, and the board requires a reset, which may take seconds to initiate and complete, and remote intervention to reboot.

6) SSM5 - FPGA Bypass: This mode removes all clocks from the SpiNNaker chip clock domains including the router (PLL1 and 2 are disabled). The routing logic must now be handled by the FPGAs which sit on the edges of all the boards.

7) SSM6 - Board Power Down: This mode removes the power from the board with the exception of the BMP to allow recovery. Both router pass-though and FPGA bypass are not possible and the board is a traffic black-hole.

D. Power Dissipation During a Full Simulation Cycle

In order to demonstrate the effects of an optimised suspend mode on the simulation cycle, we have implemented SSM2 by adding extra functionality to SARK.

Each time a neural application core participates in a simulation, it sets a bit in a specific place in the shared memory, and resets it as soon as the simulation is over. The monitor core polls that memory location periodically to check the status of the application cores and if all application core bits are reset, it enters SSM2. A chip returns back to SSM1 whenever it receives a message from the host, or another SpiNNaker chip, indicating either the uploading of data to the shared memory, downloading of results or the beginning of a simulation.

A simulation comprising 192,000 neurons with 48,000,000 current-based exponential synapses ran in real-time for 30 seconds and generated 720,000,000 synaptic events per second. For this simulation the PACMAN tool produced 208 MBytes worth of synaptic information, neural parameters, neural/synapse models and routing tables, while the size of the recorded membrane traces was 1.152 GBytes.

Power dissipation was recorded, one sample per second, from the 12 V supply. The same experiment ran four times, twice for the new optimised idle mode and twice for the default SSM1 mode to validate the data.

Figure 5 shows the power dissipated during a simulation cycle for the default idle state and the one with the proposed optimisations, for both trials. A full simulation cycle on SpiNNaker consists of three phases; uploading the simulation data, running the simulation and retrieving the results. As can be seen in the figure, all SpiNNaker chips start in SSM2 and gradually the chips that receive simulation data enter into the SSM1, returning back to SSM2 as soon as the uploading has finished. The next phase is the execution state, SSM0, where the simulation runs for 30 seconds. When the simulation has ended the SpiNNaker chips enter SSM1, which is the standard idle mode, and the next time the monitor core polls their status it sets them to SSM2. The last phase is the downloading of the membrane potentials. During this phase only the chip that sends data is in SSM1 and returns back to SSM2 when all data to retrieve is sent back to the host.

TABLE V

SPINNAKER SUSPEND MODES UNDER INVESTIGATION.

Mode	Description	DC/DC Power Out	DC/DC Loss	Board Power	Power Saving	% Saved
SSM0	Active Operation					
SSM1	Wait for Interrupt	21.99	4.85	26.84		
SSM2	Suspend With SDRAM	7.97	2.72	10.68	16.16	60
SSM3	Suspend Without SDRAM	6.40	2.00	8.40	18.44	69
SSM4	Node Routing Pass-Through	6.23	1.91	8.14	18.71	70
SSM5	FPGA Bypass	5.48	1.63	7.11	19.74	74
SSM6	Board Power Down	0.77	0.29	1.06	25.78	96

TABLE VI

THE POWER DISSIPATION OF A FULL SIMULATION CYCLE FOR THE PROPOSED (SSM2) AND DEFAULT IDLE MODE (SSM1).

Mode	Average Power (W)
SSM1	28.14
SSM2	13.53

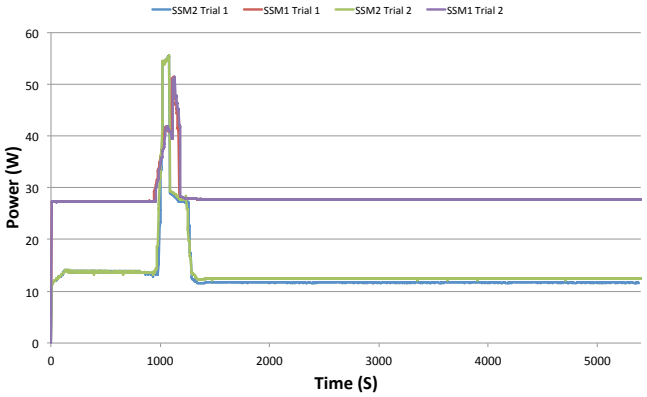

Fig. 5. A full simulation cycle on a 48-SpiNNaker board, which consists of 192,000 neurons in total with 48,000,000 current-based exponential synapses. A simulation cycle comprises three phases: uploading the simulation data, running the simulation and fetching the results. Legend denotes the proposed optimised idle state (SSM2) and the default implementation (SSM1) showing the significantly reduced baseline.

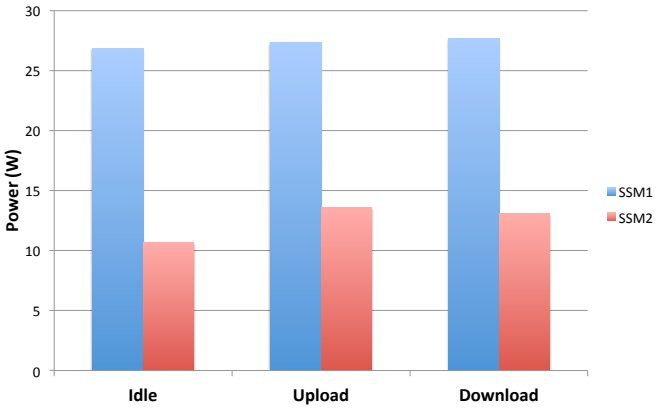

Fig. 6. Power comparison between the proposed optimised idle state (SSM2) and the default one (SSM1).

The energy consumption for each simulation phase, including the idle state, is presented in Figure 6. The simulation that incorporates the new suspend mode, SSM2, is 60% more energy efficient in the idle state, 50% more efficient during the uploading phase and 52% during the downloading phase. Table VI summarises the power dissipation for both simulations. Results, indicate that the simulation with the optimised idle mode (SSM2) is overall 52% more energy efficient than the current default one.

V. CONCLUSIONS AND FUTURE WORK

Large-scale neural simulation is a promising alternative methodology for understanding how brains process information, and this interest is reflected by projects including The SyNAPSE project [23], the Human Brain Project (HBP) [24] and the BRAIN Initiative [25].

In this paper we have investigated different approaches towards optimising the overall power dissipation of a 48 chip SpiNNaker board, which is the building block for creating larger SpiNNaker machines. We mainly focused on recoverable idle states through dynamic frequency scaling, by systematically examining the energy consumption of each clocked component within a SpiNNaker chip. The proposed optimisation was implemented and tested on a large simulation comprising thousands of neurons with tens of millions of synapses with an activity hundreds of millions of synaptic events per second. We were mainly interested to see the effect of the new idle state under a full simulation cycle, which consists of three phases: uploading models and synaptic information, running the simulation and retrieving the results.

The results show that the proposed optimisation is 60% more energy efficient for the idle state, 50% for the uploading and 52% for the downloading phase. Moreover, for the same simulation the power dissipation is reduced by 52%. We expect that these energy savings will have a bigger impact as the size of the SpiNNaker machines scale up.

Future work will include larger multi-board simulations, investigating the on-board FPGAs chips and how the intra-board communications affect the overall power dissipation. Additional idle modes for the application cores participating in a simulation but have not received or generated activity over some period of time. Finally, implementing a deeper suspend mode by powering down a SpiNNaker board and

recovering it using the board management processor.

ACKNOWLEDGMENT

The SpiNNaker project is supported by the Engineering and Physical Sciences Research Council (EPSRC) of the UK, through Grants EP/D07908X/1 and EP/G015740/1, and also by ARM and Silistix. The authors would like to thank Francesco Galluppi, Luis Plana, Steve Temple, James Knight, Steve Rhodes and Jeff Pepper. Cameron Patterson works on the PRiME programme, which is exploring the energy / reliability trade-off of the next generation of computing systems, and this work is supported through ESPRC grant EP/K034448/1. The authors appreciate the support from sponsors, industrial partners and the contributions of the current and former members of the project.

REFERENCES

[1] C. Eliasmith, T. C. Stewart, X. Choo, T. Bekolay, T. DeWolf, Y. Tang, and D. Rasmussen, "A large-scale model of the functioning brain," *Science*, vol. 338, pp. 1202–1205, 2012.

[2] H. Markram, "The Blue Brain Project," *Nat Rev Neurosci.*, vol. 7, pp. 153–160, 2006.

[3] E. M. Izhikevich and G. M. Edelman, "Large-scale model of mammalian thalamocortical systems.," *Proc Natl Acad Sci U S A*, vol. 105, pp. 3593–3598, Mar. 2008.

[4] R. Ananthanarayanan, S. K. Esser, H. D. Simon, and D. S. Modha, "The cat is out of the bag: cortical simulations with 10^9 neurons, 10^{13} synapses," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, SC '09, (New York, NY, USA), pp. 63:1–63:12, ACM, 2009.

[5] T. M. Wong, R. Preissl, P. Datta, M. Flickner, R. Singh, S. K. Esser, E. McQuinn, R. Appuswamy, W. P. Risk, H. D. Simon, and D. S. Modha, "Ten to power 14," Tech. Rep. RJ10502, IBM, April 2013.

[6] "TOP500 Supercomputer Site." `http://www.top500.org`.

[7] F. Galluppi, K. Brohan, S. Davidson, T. Serrano-Gotarredona, J.-A. P. Carrasco, B. Linares-Barranco, and S. Furber, "A real-time, event-driven neuromorphic system for goal-directed attentional selection," in *Neural Information Processing*, pp. 226–233, Springer, 2012.

[8] F. Galluppi, J. Conradt, T. Stewart, C. Eliasmith, T. Horiuchi, J. Tapson, B. Tripp, S. Furber, and R. Etienne-Cummings, "Live demo: Spiking ratslam: Rat hippocampus cells in spiking neural hardware," in *Biomedical Circuits and Systems Conference (BioCAS), 2012 IEEE*, pp. 91–91, IEEE, 2012.

[9] P. Lichtsteiner, C. Posch, and T. Delbruck, "A 128 x 128 120 db 15 us latency asynchronous temporal contrast vision sensor," *Solid-State Circuits, IEEE Journal of*, vol. 43, pp. 566–576, Feb 2008.

[10] J. Leero-Bardallo, T. Serrano-Gotarredona, and B. Linares-Barranco, "A 3.6 us latency asynchronous frame-free event-driven dynamic-vision-sensor," *Solid-State Circuits, IEEE Journal of*, vol. 46, pp. 1443–1455, June 2011.

[11] S.-C. Liu, A. van Schaik, B. Minch, and T. Delbruck, "Event-based 64-channel binaural silicon cochlea with q enhancement mechanisms," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pp. 2027–2030, June 2010.

[12] S. Furber and S. Temple, "Neural Systems Engineering," *Computational Intelligence: A Compendium*, pp. 763–796, 2008.

[13] S. B. Furber, S. Temple, and A. D. Brown, "On-chip and inter-chip networks for modeling large-scale neural systems," in *ISCAS*, 2006.

[14] 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*, pp. 1–8, Aug 2013.

[15] S. Furber and S. Temple, "Neural Systems Engineering," *Journal of the Royal Society Interface*, vol. 4, pp. 193–206, 2006.

[16] M. Mahowald, *An Analog VLSI System for Stereoscopic Vision*. Norwell, MA, USA: Kluwer Academic Publishers, 1994.

[17] S. Furber and A. Brown, "Biologically-inspired massively-parallel architectures - computing beyond a million processors," in *Application of Concurrency to System Design, 2009. ACSD '09. Ninth International Conference on*, pp. 3–12, July 2009.

[18] S. B. Furber, D. R. Lester, L. A. Plana, J. D. Garside, E. Painkras, S. Temple, and A. D. Brown, "Overview of the spinnaker system architecture," *IEEE Transactions on Computers*, vol. 99, no. PrePrints, 2012.

[19] T. Sharp, L. A. Plana, F. Galluppi, and S. Furber, "Event-driven simulation of arbitrary spiking neural networks on spinnaker," in *Neural Information Processing*, vol. 7064 of *Lecture Notes in Computer Science*, pp. 424–430, Springer Berlin Heidelberg, 2011.

[20] A. P. Davison, D. Brderle, J. M. Eppler, J. Kremkow, E. Muller, D. Pecevski, L. Perrinet, and P. Yger, "PyNN: a common interface for neuronal network simulators," *Frontiers in Neuroinformatics*, vol. 2, p. 11, 2009.

[21] F. Galluppi, S. Davies, A. Rast, T. Sharp, L. Plana, and S. Furber, "A Hierarchical Configuration System for a Massively Parallel Neural Hardware Platform," in *(Accepted for) ACM International Conference on Computing Frontiers*, 2012.

[22] E. D. Schutter, *Computational Modeling Methods for Neuroscientists*. The MIT Press, 1st ed., 2009.

[23] J. Arthur, P. Merolla, F. Akopyan, R. Alvarez, A. Cassidy, S. Chandra, S. Esser, N. Imam, W. Risk, D. Rubin, R. Manohar, and D. Modha, "Building block of a programmable neuromorphic substrate: A digital neurosynaptic core," in *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pp. 1–8, June 2012.

[24] "Human Brain Projet." `https://www.humanbrainproject.eu`.

[25] "Brain Research through Advancing Innovative Neurotechnologies (BRAIN) Initiative." `http://www.nih.gov/science/brain/`.