# Abstract

SpiNNaker is an asynchronous, event-driven parallel architecture designed to simulate the human brain. It has been designed to operate as a large scale neural in real-time network using a System-on-Chip multicore system. Its architecture is different from usual parallel computers, by only using spikes to communicate, with no notion of sequential programming existing. That way it avoids usual pitfalls of parallel computing, such as race conditions and deadlocks. So far the most prominent uses of this architecture have been in neuroscience and robotics. The aim of this project is to put into use SpiNNaker's architecture by bringing it closer to classic computer science problems. The given problem is the conjugate gradient method, an iterative way of solving particular systems linear equations.

# Introduction

SpiNNaker is an architecture inspired by the biology of the human brain. Its optimal configuration can use over a million cores[2], which can be used to simulate the neurons that exist in the human brain. It is a System-on-Chip machine that provides massive parallelism to the user, which escapes most norms of parallel computing, without the main pitfalls of parallel computing, such as race conditions, deadlocks, mutual exclusion etc. [3]. The cores in the system communicate with each other with spikes(packets), in the same fashion neurons communicate and are event-driven. A computation can be done, in the case of an event, such as the delivery of a packet, or a DMA completion.

The Conjugate Gradient Method is an iterative way of solving symmetric and positive definite systems of linear equations. Its iterative property makes it easy to even solve sparse systems, that other methods might have problems solving.

The aim of this report is to explain in more detail the aforementioned concepts, as well as give an account of the background research done to launch the project. It will also try to sketch out some aspects of the final design of the system, as well as the reasons why certain choices in the design were made. Finally, it will show the tasks complete so far, along with a plan for the remaining work.

# Project Description

## Neuron functionality

In order to explain the SpiNNaker architecture some basic functionality of the human neuron should be presented. A standard human neuron can be divided into three parts, the dendrites, the soma and the axon. Each neuron can be connected to a number of different other neurons, throughout the entire human body. The dendrites are the parts of the neuron, which receive information from other neurons. The axon is the part that delivers information to other neurons and the soma is the processing unit of the neuron.[4 Introduction]

Each time a neuron wants to send information it does so by using a spike through its axon and the targeted neuron’s dendrite. The part of the connection where the dendrite connects to the axon is called synapse.

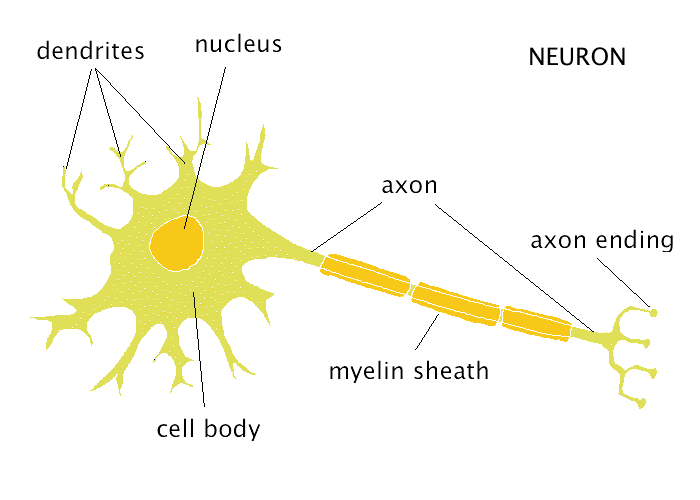


Figure 1: A neuron

## SpiNNaker architecture

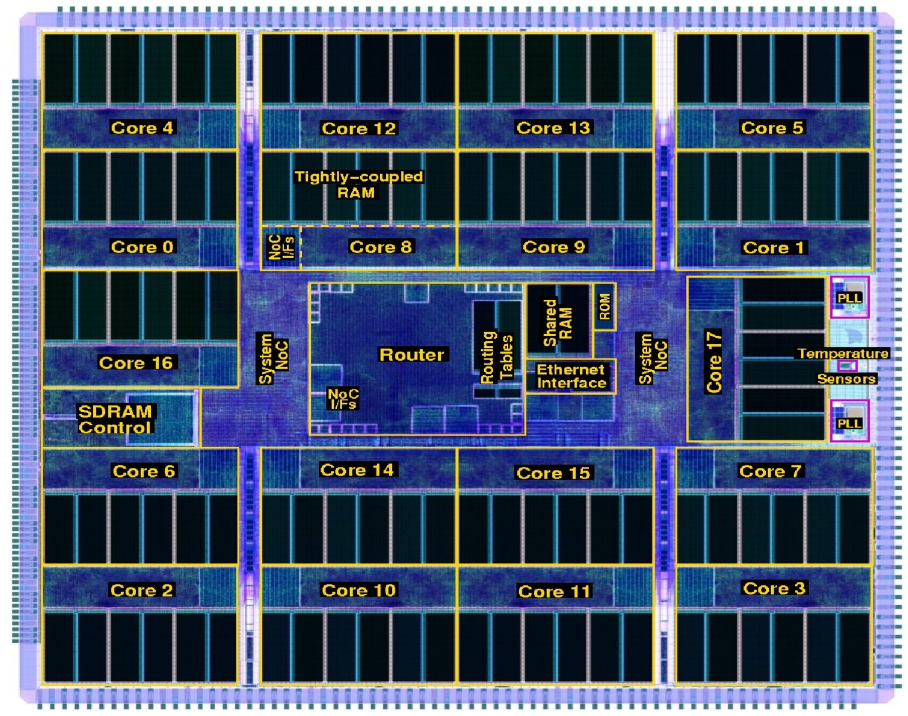
As mentioned before SpiNNaker is a massive parallel computer able to host over a million cores. At the heart of the machine the SpiNNaker chip can be found.

Figure 2. The SpiNNaker chip[5]

As can be seen in Figure 2, the main components of a SpiNNaker chip are 18 ARM cores, a router and an SDRAM. At start-up, after each core finishes its test, a Monitor core is selected, with a responsibility of monitoring the other cores. Another core is used as a backup core and all the other are used for computational purposes[6].

Following the example of neurons, a core in the chip uses spikes to communicate with the other cores. Each spike is relatively small (70 bits at most), thus following the example of neurons[5]. Each core can only communicate to 6 other adjacent cores, but by configuring the connectivity each core can communicate with any core in the system. To make this communication possible, each chip incorporates a router that distributes these messages to other cores. Its main purpose is to direct the messages sent by some cores to other cores correctly. To achieve that the router supports multicast communication, which helps the network run faster by reducing the number of packets sent[2]. The way the multicast packets work, is that each packet consists no information about the destination core, but rather only information about the core that the packet was sent from. If for some reason some of the packets fail to send, or fail to arrive, the system has the ability of replicating these messages and resending them.

As mentioned before, a SpiNNaker machine divides its cores into monitor cores and application cores. However, there is another type of core that its main use is to handle Input and Output, program loading etc. The official name of these types of host cores, is ybug. Furthermore a monitor core is called ybug and an application core is called scamp[5].

The programming model of a SpiNNaker machine is an event-driven model. That means that a computation can only happen following an event. That event can be a packet having sent or arrived, a DMA access, or a timer event. There is no way for an application to control the flow of a program. The programmer can only design event handlers that will be executed when an event occurs and give them a certain priority[3]. When that event occurs, the kernel will execute these callbacks and immediately, or put them in a queue, depending the priority they have.

## Conjugate Gradient Method

As explained before, the Conjugate Gradient Method(CGM) is an iterative way of solving systems of linear equations, whose matrix is symmetric and positive definite. Its pseudo code can be seen in Figure 3.[7]

PUT PSEUDO CODE

CGM based on its layout needs storing of many variables, along with many matrix multiplications. That makes the algorithm by default resource and computationally heavy. The most expensive calculation of the algorithm seems to be the vector-matrix multiplication, which has complexity O(N^2).

## Final Description and Goals

To sum up, based on the description of the constitutes of the problem, the problem itself has as a goal not only to solve the CGM in a SpiNNaker machine, but also to reduce the time needed to solve the problem by doing computations concurrently.

# Background Research and Literature

To begin the project some background reading about the main aspects of it was done in the beginning of the year. The first pieces of information about the project came from studying the contents of the website of the SpiNNaker project[5].

After studying the website’s contents some more general papers were studied in order to establish better foundations for the project[1][6][Overview of the SpiNNaker chip]. To acquire further understanding of architecture of the SpiNNaker chip, papers which describe in more detail certain aspects of the project were studied. These would describe the connection and communication between cores and different SpiNNaker chips[1] and others would describe the process of creating event handlers[3].

Furthermore, some papers were studied to view the uses of the SpiNNaker chip so far.PES KAPIA PAPERS KAI TI KANOUN.

Finally some research was done in order to understand the CGM. This was accomplished by studying several sources, but the main one being a book[8].

# Final Design

# Justification for the Approach

# Completed Work

# Plan for remaining work

# Conclusion

# References

[1]Neural Systems Engineering(Steve Furber)

[2]Understanding the Interconnection Network of SpiNNaker

[3]Event Driven Simulation of Arbitrary Spiking Neural Networks on SpiNNaker

[4]Neuron book

[5]SpiNNaker website(Architectural overview)

[6] A GALS infrastructure for a Massively parallel processor

[7] Reference a book

[8]Conjugate Gradient Method book