MANCHESTER 1824
The University of Manchester

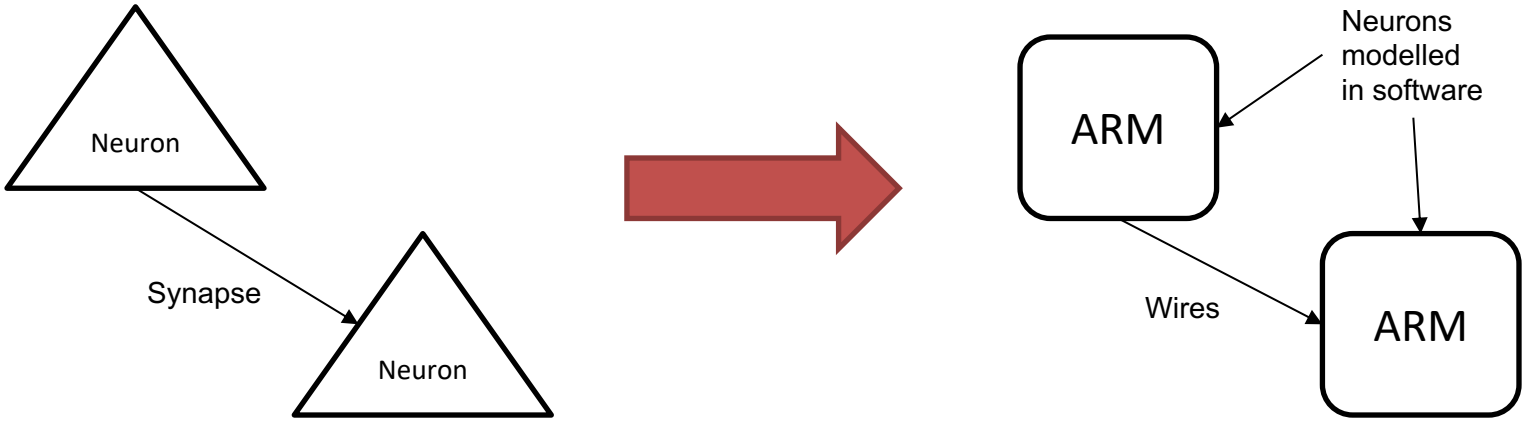
Improving performance of synapse processing in massively parallel neuromorphic systems

Mantas Mikaitis
Supervisor: Dr David R. Lester
APT group

MANCHESTER 1824
The University of Manchester

SpiNNaker Neuromorphic Computer

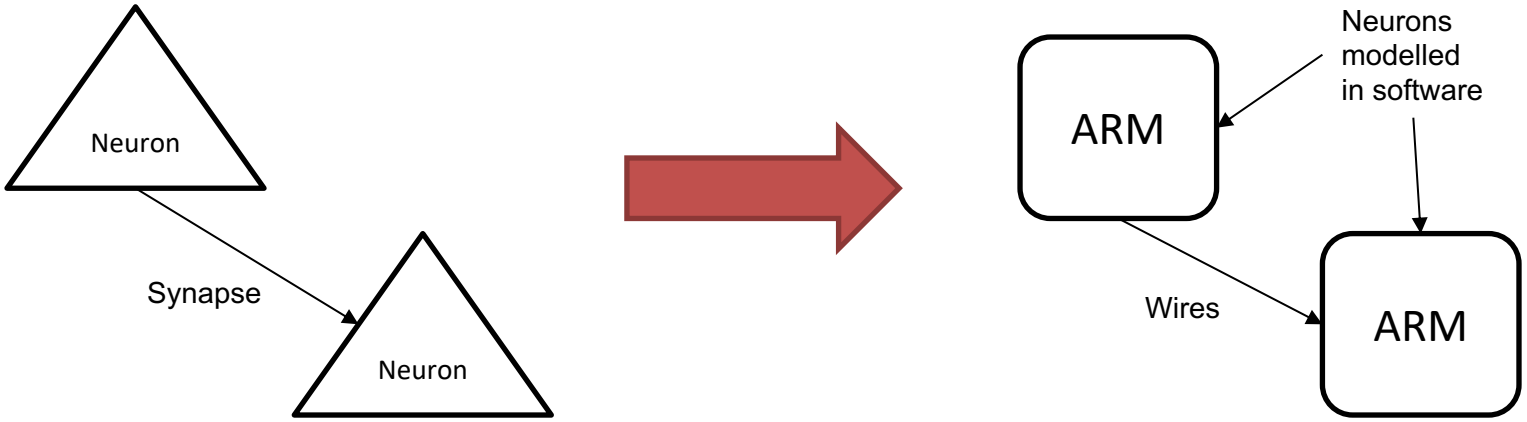
- Neuromorphic – resembling brain interconnect



MANCHESTER 1824
The University of Manchester

SpiNNaker Neuromorphic Computer

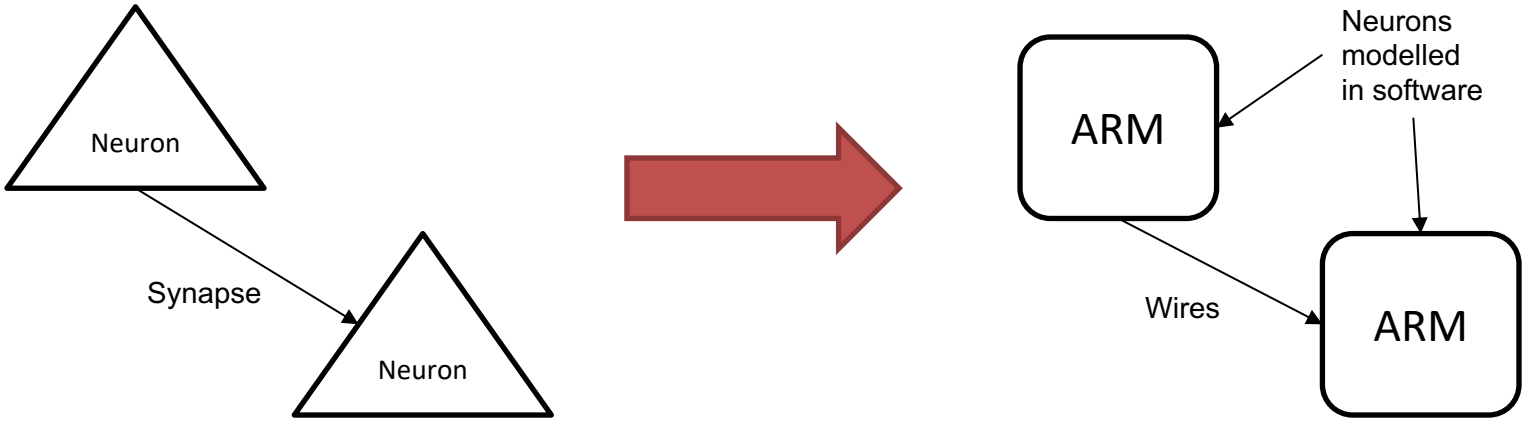
- Neuromorphic – resembling brain interconnect
- SpiNNaker – A name of a million core machine developed in Manchester.



MANCHESTER 1824
The University of Manchester

SpiNNaker Neuromorphic Computer

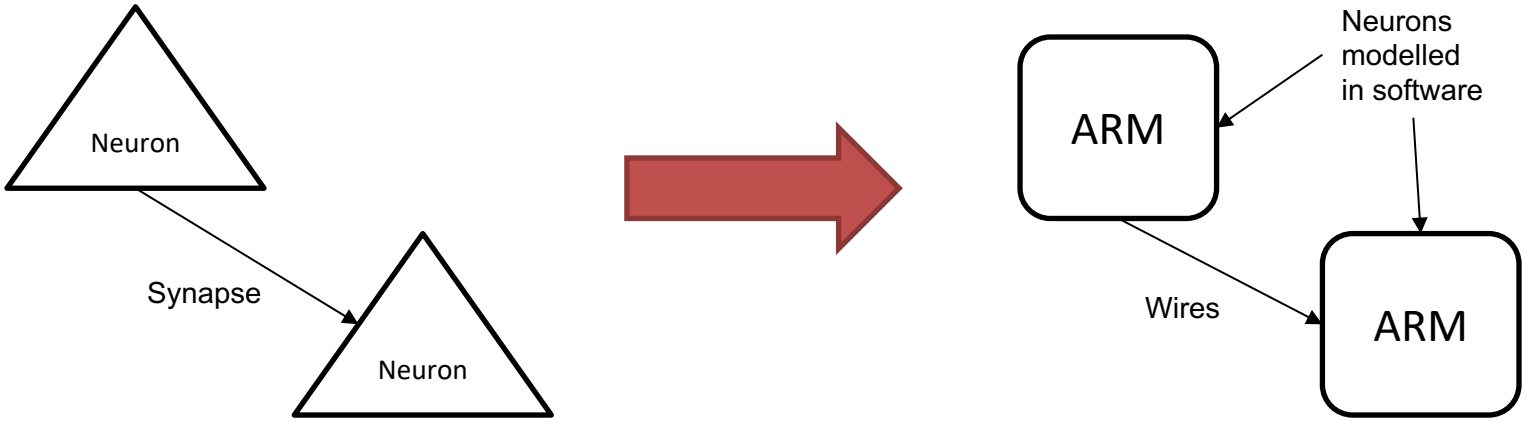
- Neuromorphic – resembling brain interconnect
- SpiNNaker – A name of a million core machine developed in Manchester.
- Main goal: Simulate 1% of human brain.



MANCHESTER 1824
The University of Manchester

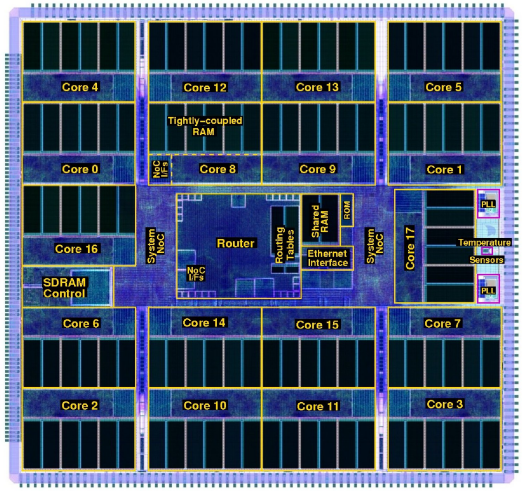
SpiNNaker Neuromorphic Computer

- Neuromorphic – resembling brain interconnect
- SpiNNaker – A name of a million core machine developed in Manchester.
- Main goal: Simulate 1% of human brain.
- Made of ARM cores (Same cores that are in mobile phones)

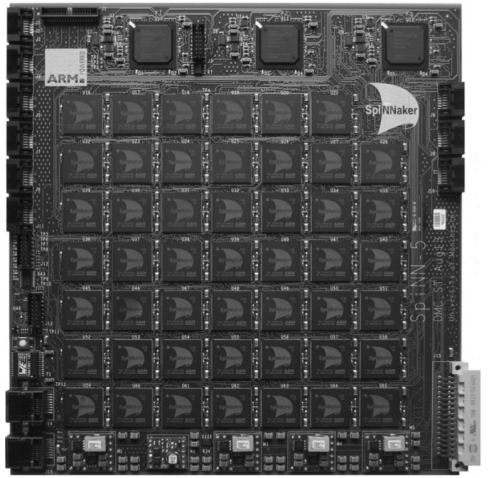


SpiNNaker Neuromorphic Computer

~4000 neurons



SpiNNaker chip



A board with 48 chips

720 boards (1200 boards in final stage for 1% of brain)

MANCHESTER 1824
The University of Manchester

Main challenges of the SpiNNaker project

Hardware:

- Processing millions of signals through million cores
- Power efficiency

Software:

- Biological plausibility – does it simulate accurate brain behavior?
- Real-Time running constraint – can ARM cores run simulation in real-time (Not slowed down).

MANCHESTER 1824
The University of Manchester

What I intend to address in my PhD

Most of the time SpiNNaker spends on processing synapses.

If an algorithm is **learning**, synapse strengths are changed which poses even more challenge.

I am interested in improving this using hardware and software optimizations in order to simulate **larger** neural networks in **real-time**.

MANCHESTER 1824
The University of Manchester

Learning on SpiNNaker-1

- Learning in brain is called **synaptic plasticity**.

MANCHESTER 1824
The University of Manchester

Learning on SpiNNaker-1

- Learning in brain is called **synaptic plasticity**.
- Strengths of synapses are changing so that some **neurons learn to fire a signal (spike) more often or less often**.

MANCHESTER 1824
The University of Manchester

Learning on SpiNNaker-1

- Learning in brain is called **synaptic plasticity**.
- Strengths of synapses are changing so that some **neurons learn to fire a signal (spike) more often or less often**.
- On SpiNNaker, a mathematical learning rule can be injected to describe synaptic change.

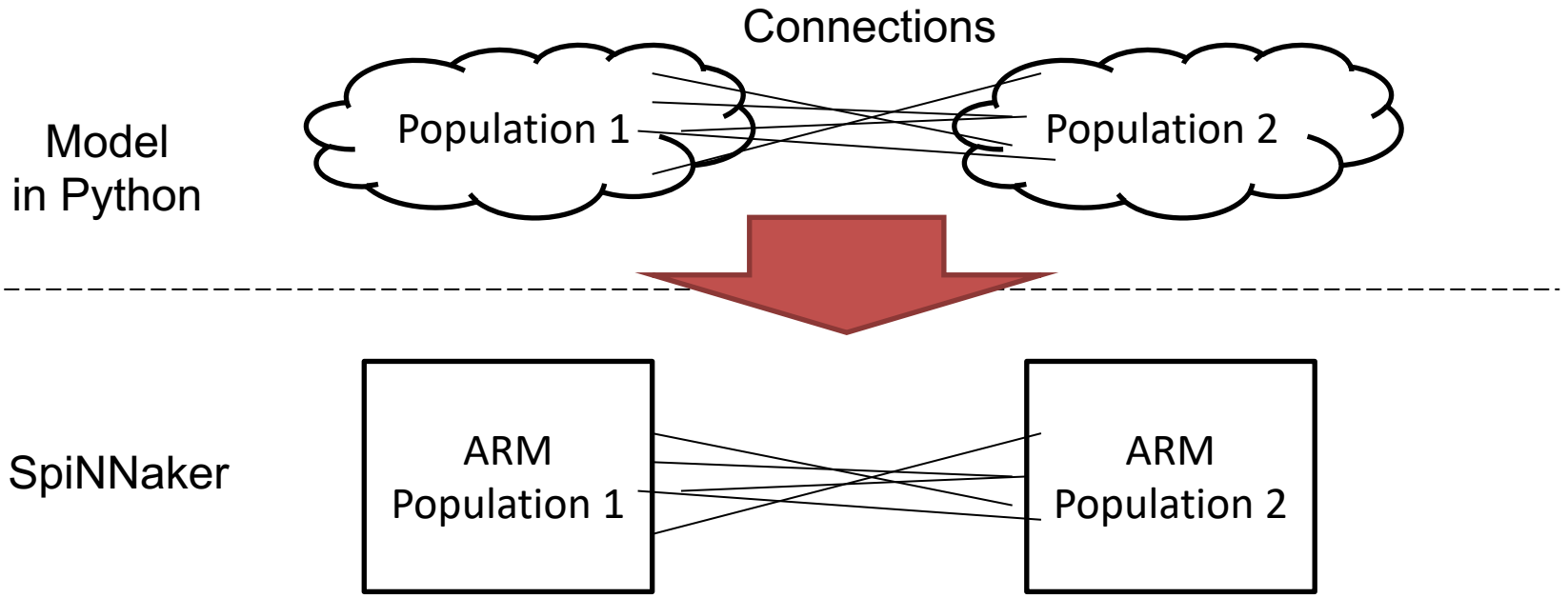
MANCHESTER 1824
The University of Manchester

Learning on SpiNNaker-1

- Learning in brain is called **synaptic plasticity**.
- Strengths of synapses are changing so that some **neurons learn to fire a signal (spike) more often or less often**.
- On SpiNNaker, a mathematical learning rule can be injected to describe synaptic change.
- Most rules involve analyzing a **history of spikes of two neurons connected by a synapse** – slow, memory-heavy process.

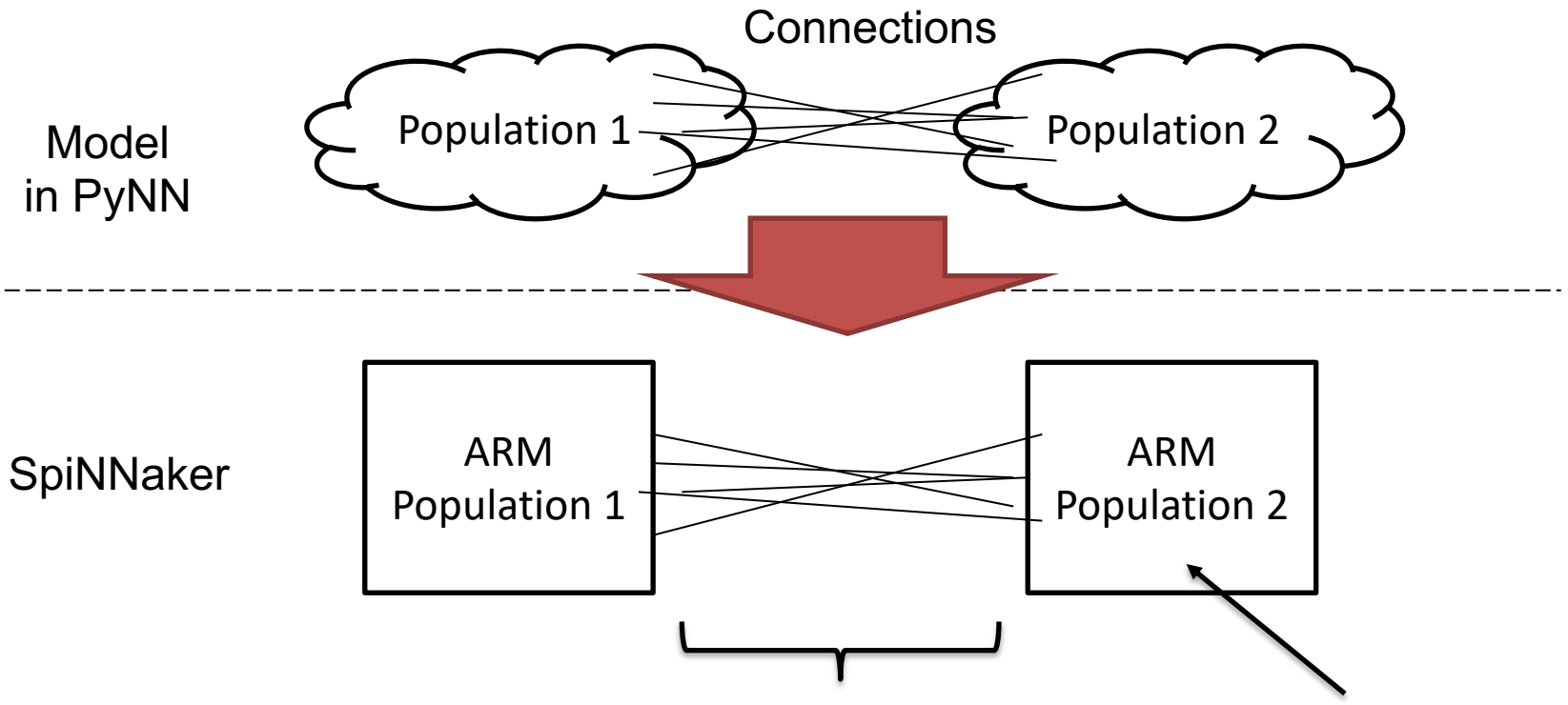
Simulating plastic neural networks on SpiNNker

- Each ARM core gets assigned a set (of some controlled size) of neurons that it is responsible for.
- When one neuron spikes, ARM core has to receive or send out a spike.



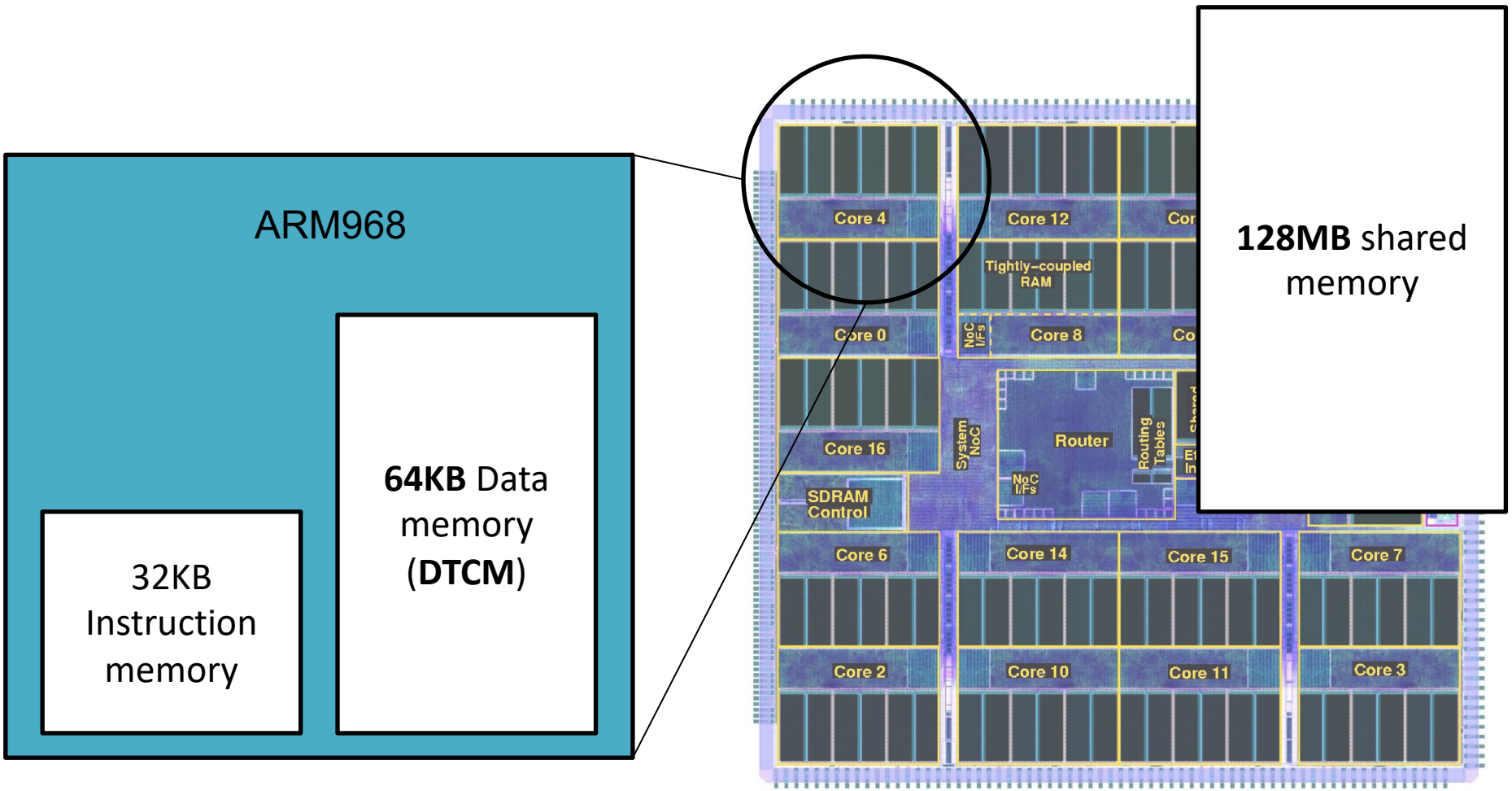
MANCHESTER 1824
The University of Manchester

Simulating plastic neural networks on SpiNNker



MANCHESTER 1824
The University of Manchester

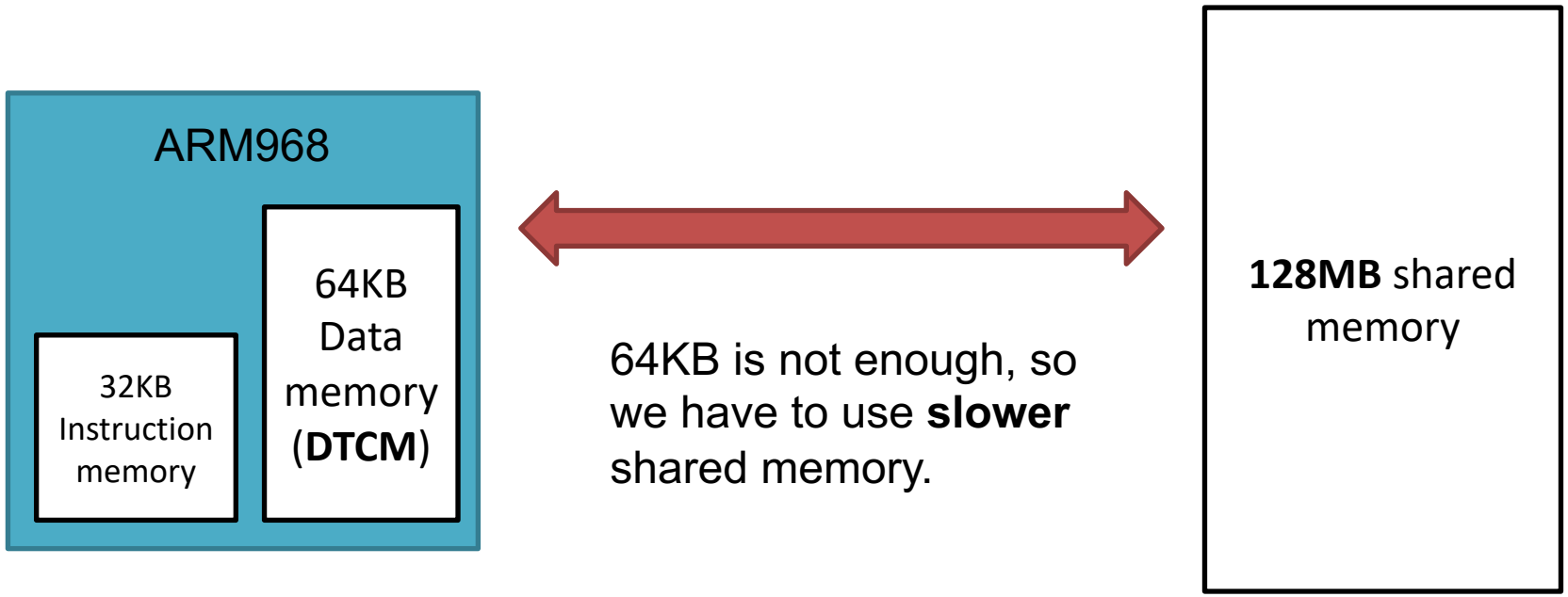
SpiNNaker-1 Chip Tear-Down



Single chip: 18 ARM968 cores and 128MB shared SDRAM

MANCHESTER 1824
The University of Manchester

Simulating plastic neural networks on SpiNNker



MANCHESTER 1824
The University of Manchester

What ARM core spends time on when receiving a spike

- **Copying** neuron spike history from SDRAM to internal memory.

MANCHESTER 1824
The University of Manchester

What ARM core spends time on when receiving a spike

- **Copying** neuron spike history from SDRAM to internal memory.
- **Copying** neuron parameters from SDRAM to internal memory.

MANCHESTER 1824
The University of Manchester

What ARM core spends time on when receiving a spike

- **Copying** neuron spike history from SDRAM to internal memory.
- **Copying** neuron parameters from SDRAM to internal memory.
- **(For each neuron (~255))** Separating packed data into different parameters using **bit-masking and shifting instructions**.

MANCHESTER 1824
The University of Manchester

What ARM core spends time on when receiving a spike

- **Copying** neuron spike history from SDRAM to internal memory.
- **Copying** neuron parameters from SDRAM to internal memory.
- **(For each neuron (~255))** Separating packed data into different parameters using **bit-masking and shifting instructions**.
- Apply spike using **mathematical** model specified.

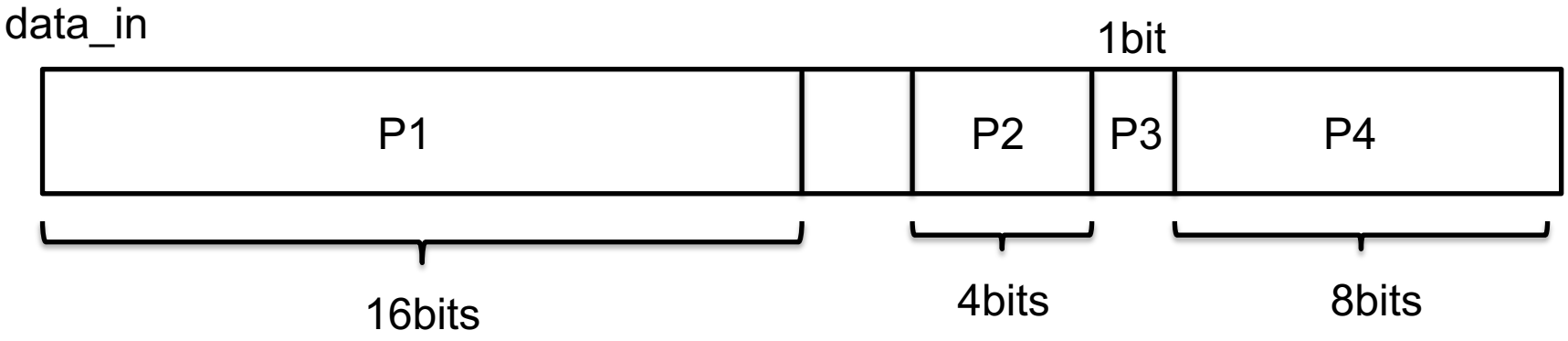
MANCHESTER 1824
The University of Manchester

What ARM core spends time on when receiving a spike

- **Copying** neuron spike history from SDRAM to internal memory.
- **Copying** neuron parameters from SDRAM to internal memory.
- **(For each neuron (~255))** Separating packed data into different parameters using **bit-masking and shifting instructions**.
- Apply spike using **mathematical** model specified.
- **Copying** changed parameters back into SDRAM.

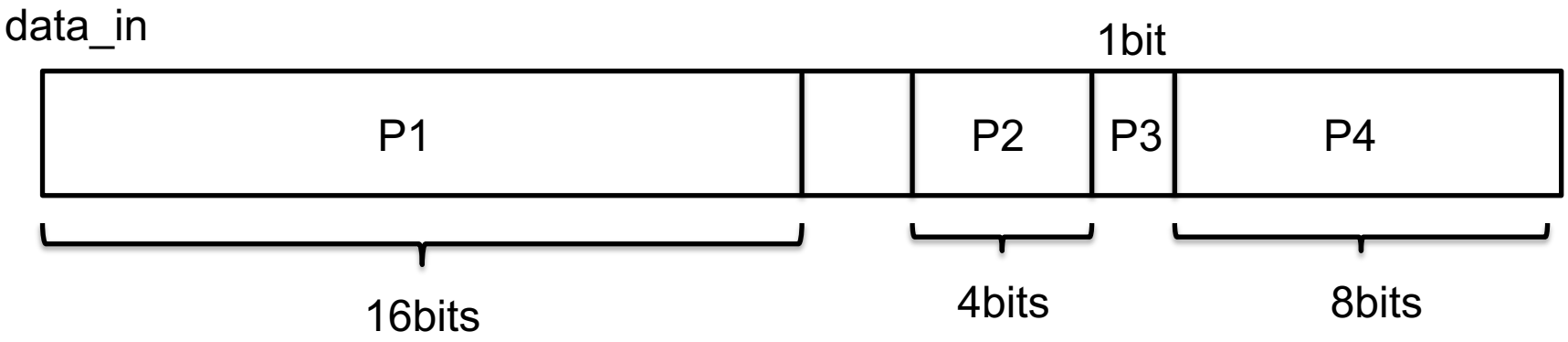
MANCHESTER 1824
The University of Manchester

Running example – bit shifting and masking



P1 = (data_in &0xFFFF0000) >> 16
P2 = (data_in & 0x00001E00) >> 9
P3 = (data_in & 0x00000100) >> 8
P4 = (data_in & 0x000000FF)

Running example – bit shifting and masking

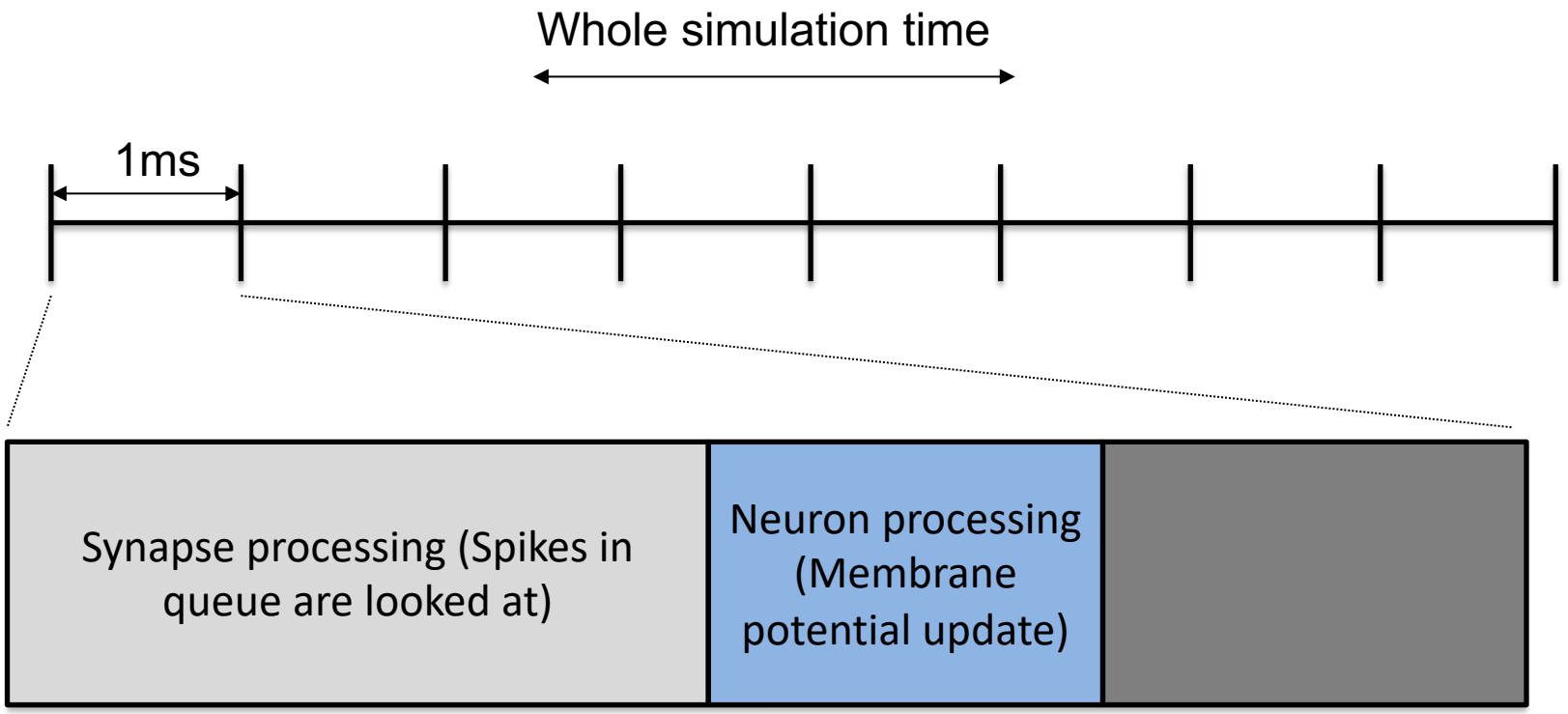


P1 = (data_in &0xFFFF0000) >> 16
P2 = (data_in & 0x00001E00) >> 9
P3 = (data_in & 0x00000100) >> 8
P4 = (data_in & 0x000000FF)

This is done hundreds of times per 1ms.

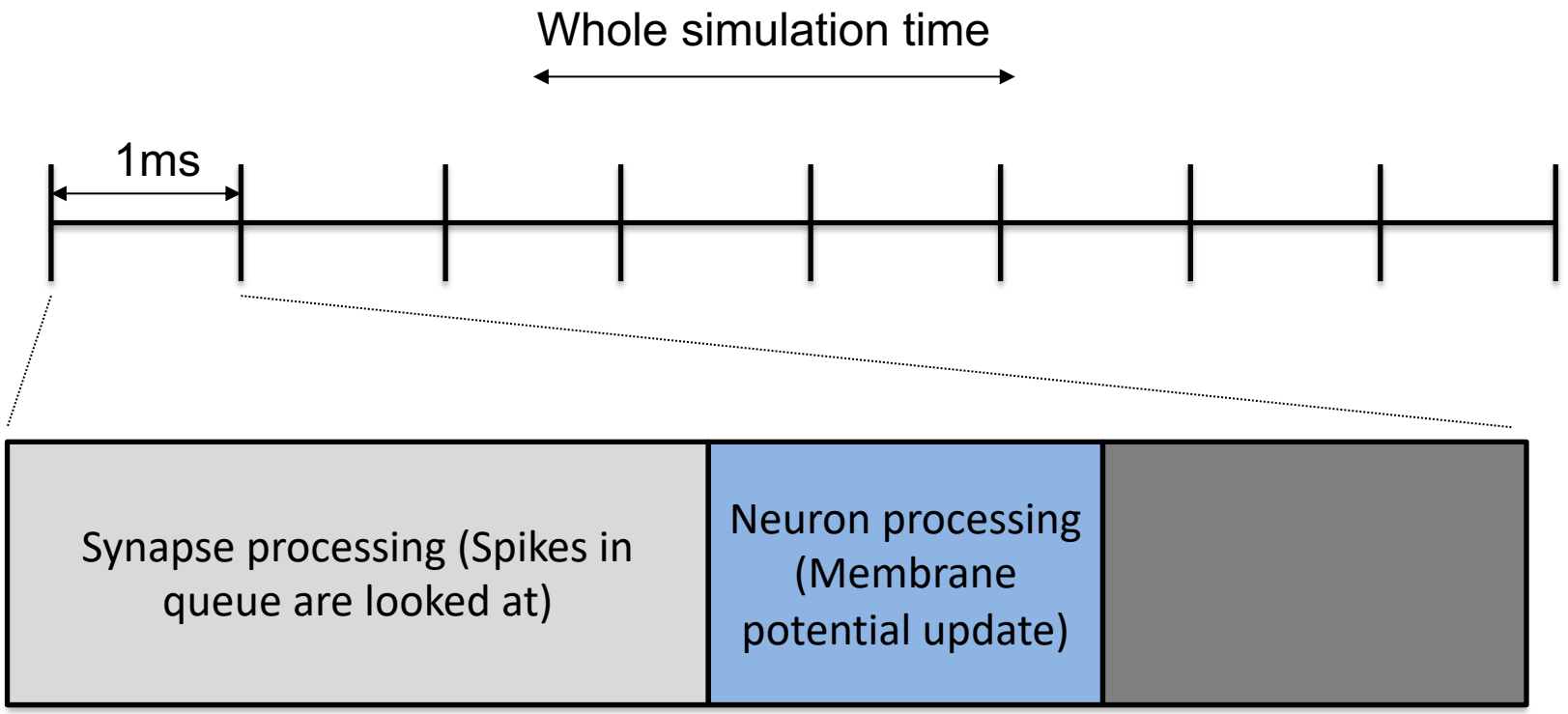
MANCHESTER 1824
The University of Manchester

Real-Time simulation constraint



MANCHESTER 1824
The University of Manchester

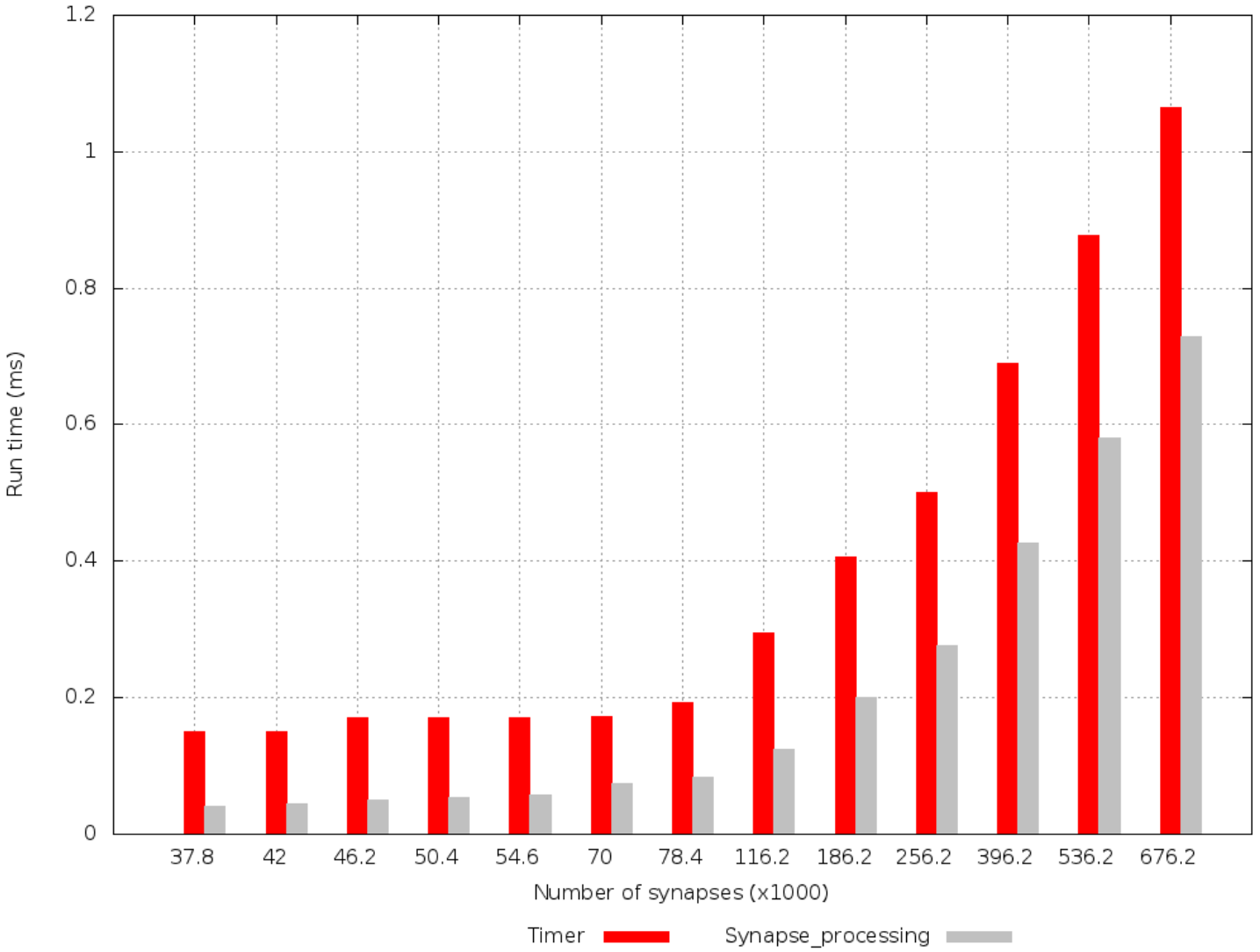
Real-Time simulation constraint



Important rule: Run-time of synapse processing and neuron processing functions combined, cannot overrun 1ms.

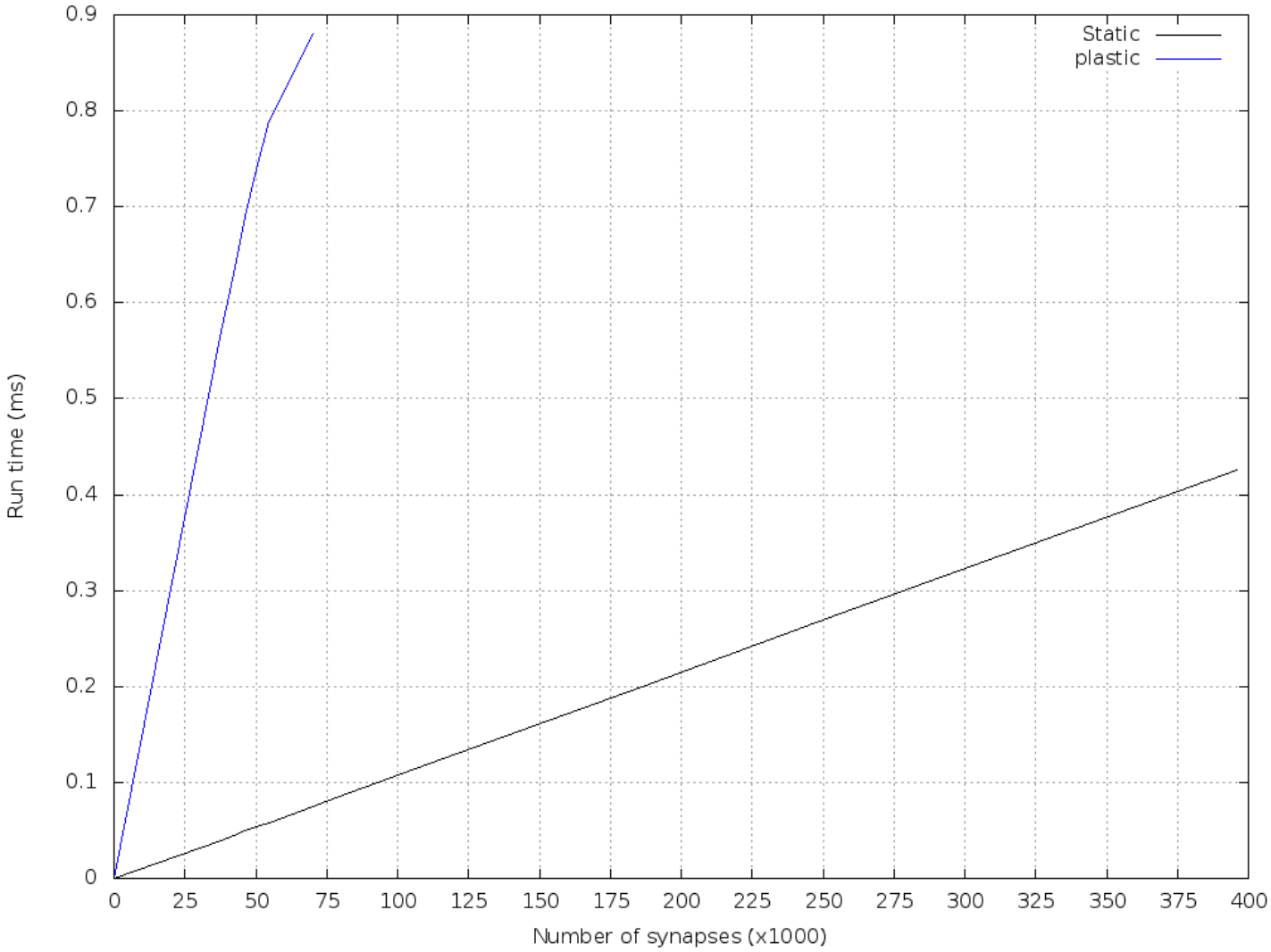
MANCHESTER 1824
The University of Manchester

Real-Time simulation constraint



MANCHESTER 1824
The University of Manchester

Real-Time simulation constraint



MANCHESTER 1824
The University of Manchester

What ARM core spends time on when receiving a spike

- **Copying** neuron spike history from SDRAM to internal memory.
- **Copying** neuron parameters from SDRAM to internal memory.
- **(For each neuron (~255))** Separating packet data into different parameters using **bit-masking and shifting instructions**.
- Apply spike using **mathematical** model specified.
- **Copying** changed parameters back into SDRAM.

MANCHESTER 1824
The University of Manchester

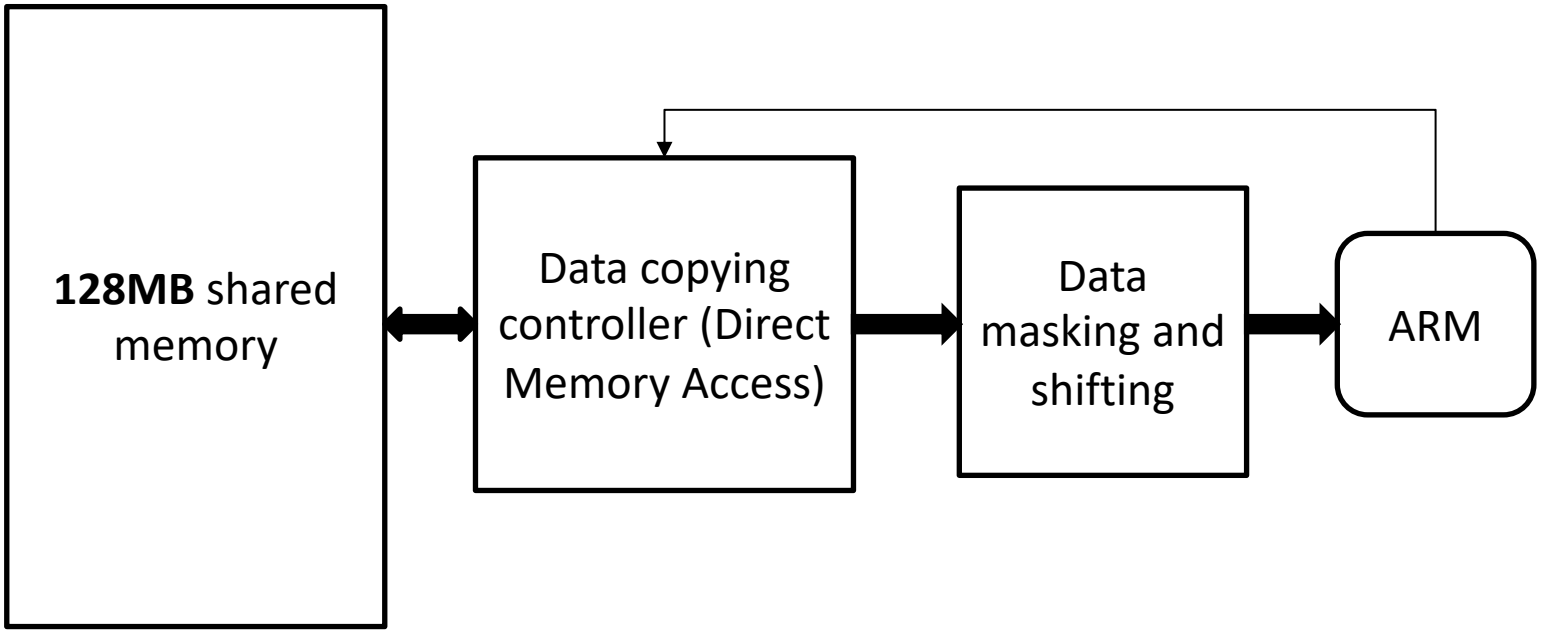
Methods – Hardware accelerators

- **Copying** neuron spike history from SDRAM to internal memory.
- **Copying** neuron parameters from SDRAM to internal memory.
- **(For each neuron (~255))** Separating packet data into different parameters using **bit-masking and shifting instructions**.
- Apply spike using **mathematical** model specified.
- **Copying** changed parameters back into SDRAM.

MANCHESTER 1824
The University of Manchester

Methods – Hardware accelerators

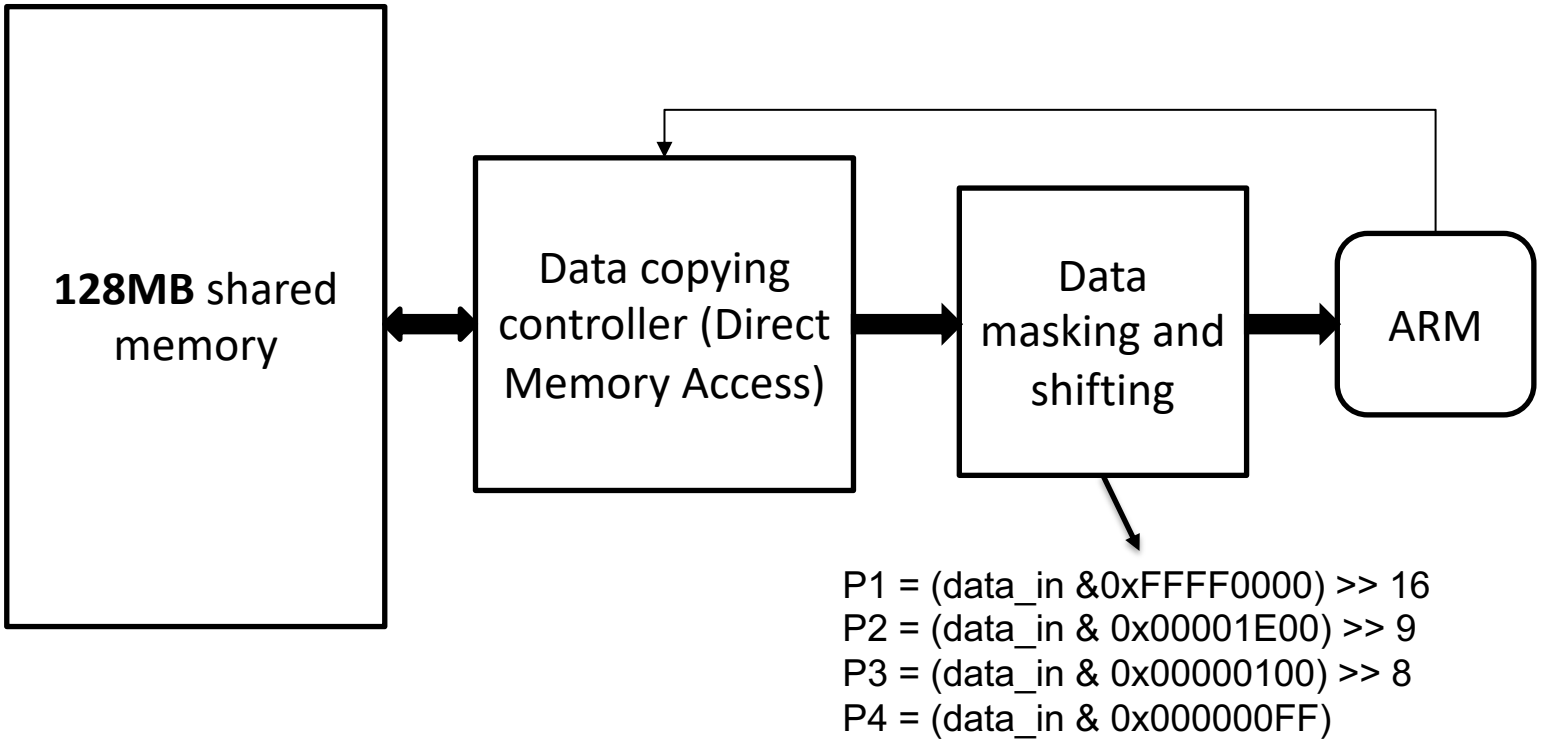
- Create a hardware module that will do bit-masking and shifting of data before it is written to ARM core.



MANCHESTER 1824
The University of Manchester

Methods – Hardware accelerators

- Create a hardware module that will do bit-masking and shifting of data before it is written to ARM core.



MANCHESTER 1824
The University of Manchester

Methods – Hardware accelerators

- Many neural algorithms require EXP, LOG and Random number generators.

MANCHESTER 1824
The University of Manchester

Methods – Hardware accelerators

- Many neural algorithms require EXP, LOG and Random number generators.
- Currently all this is done **in software**.

MANCHESTER 1824
The University of Manchester

Methods – Hardware accelerators

- Many neural algorithms require EXP, LOG and Random number generators.
- Currently all this is done **in software**.
- Create a hardware module that will do EXP and LOG functions **without multiplication** (Using shifting and adding instead).

MANCHESTER 1824
The University of Manchester

Methods – Hardware accelerators

- Many neural algorithms require EXP, LOG and Random number generators.
- Currently all this is done **in software**.
- Create a hardware module that will do EXP and LOG functions **without multiplication** (Using shifting and adding instead).
- More possibilities exist, depending on specific problems of **types of simulations**.

MANCHESTER 1824
The University of Manchester

Methods – Hardware accelerators

- Many neural algorithms require EXP, LOG and Random number generators.
- Currently all this is done **in software**.
- Create a hardware module that will do EXP and LOG functions **without multiplication** (Using shifting and adding instead).
- More possibilities exist, depending on specific problems of **types of simulations**.
- The main rule is that hardware is fixed, so we have to **identify fixed parts of current and future simulation types**.

MANCHESTER 1824
The University of Manchester

Summary

- Synapse processing on SpiNNaker poses challenges for real-time simulations
- Software optimizations are not enough
- Proposal is to introduce hardware accelerators to speed up the process