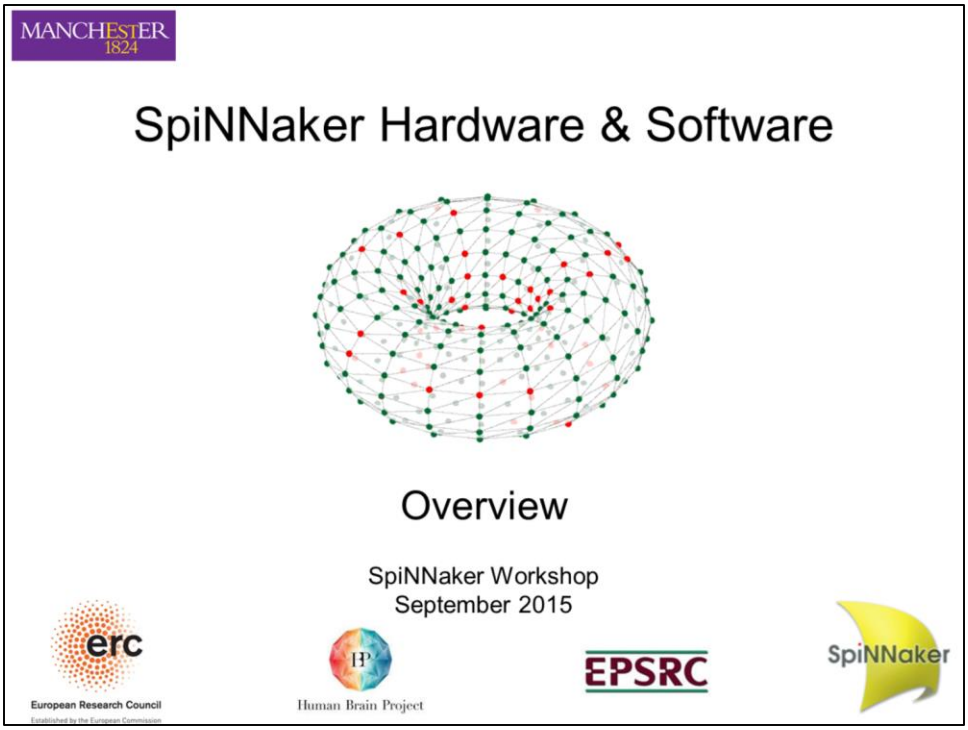


This presentation is to provide a quick overview of the hardware and software of SpiNNaker. It introduces some key concepts of the topology of a SpiNNaker machine, the unique message passing and routing functionality and the chip architecture with its restrictions and limitations.

The other function of this talk is to introduce some terminology that will be used through the workshop.

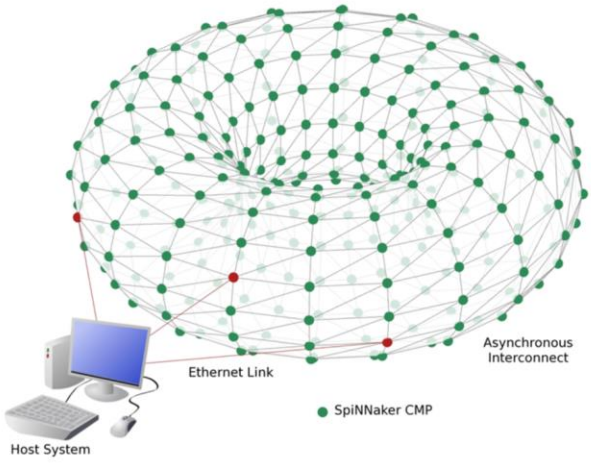
Much of the information herein will be expanded upon in later talks, so don't be too concerned with remembering everything!

MANCHESTER 1824

Contents

- What is SpiNNaker?
- SpiNNaker at different scales
- SpiNNaker architecture: chip & system
- Using SpiNNaker

SpiNNaker Project

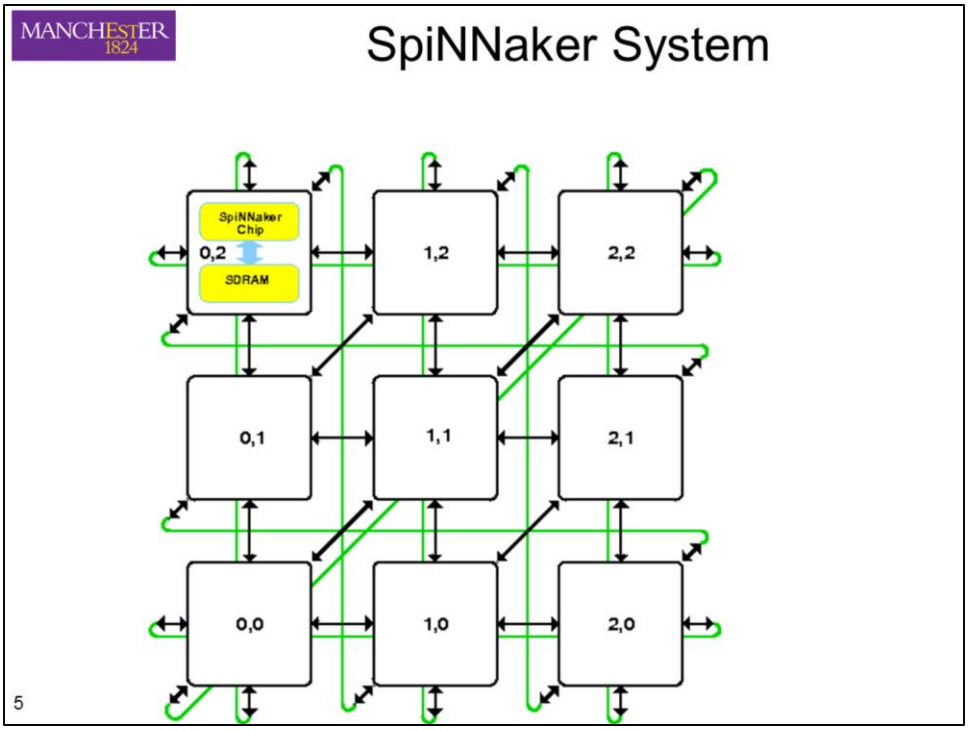


A million mobile phone processors in one computer
Able to model about 1% of the human brain…
…or 10 mice!

MANCHESTER 1824

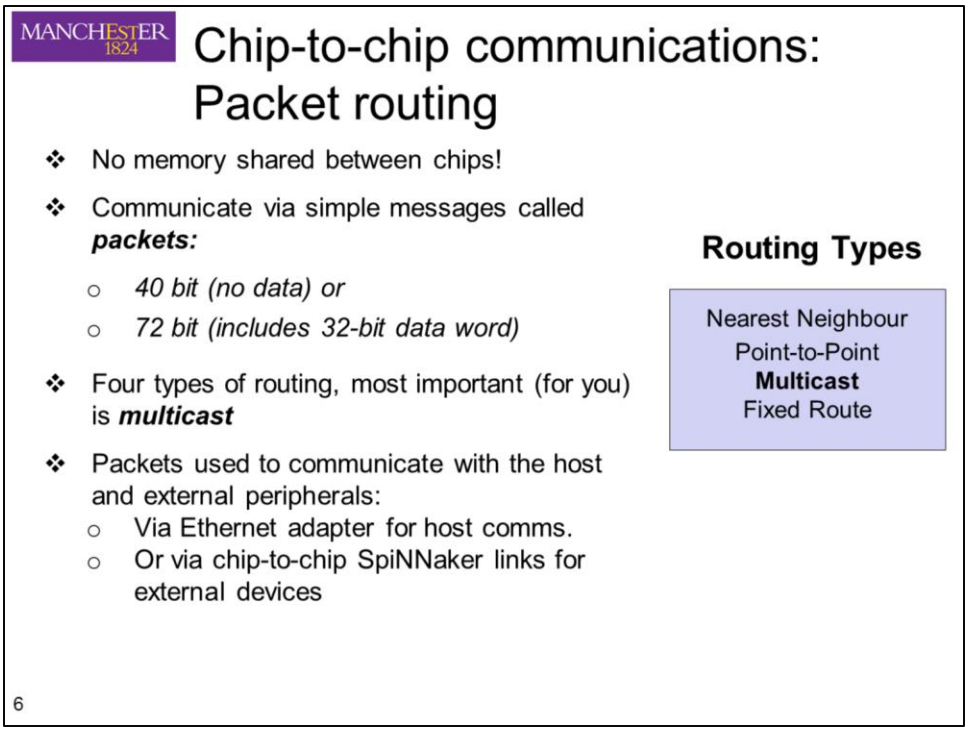
How is SpiNNaker Used?

- Some key user communities:
  - **Computational neuroscientists** to simulate large neural models and try to understand the brain
  - **Roboticists** to build advanced neural sensory and control systems
  - **Computer architects** to apply neural theories of computation to non-neural problems



SpiNNaker chips have six links: North, South, East, West, North-East and South-West. Links are bi-directional and work independently.

Topologically, an array of SpiNNaker chips forms a hexagonal grid, which can wrap around to form a cylinder or toroid. Machines can be constructed from an arbitrary sized array of chips, up to 256 x 256 in size.



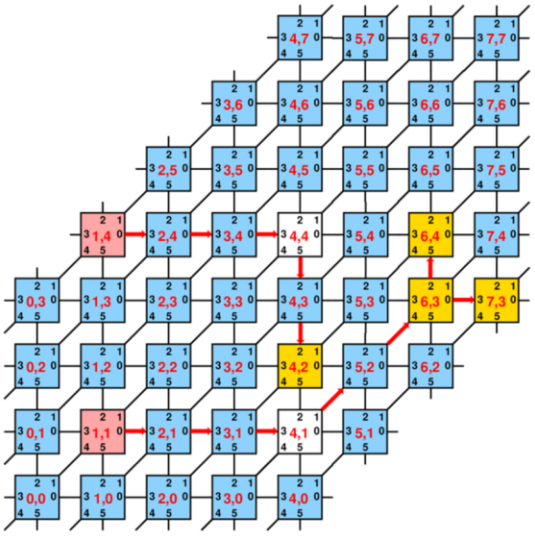
The only way for cores on two different chips to communicate is via simple messages that are passed from one chip to the other until they reach their destination. These messages are called packets. The optional dataword with each packet is called its *payload* and is always 32-bits. When the host reads or writes data to/from SpiNNaker the data is broken down into many of these packets (thought the user does not need to know this!)

The multicast routing type is the most flexible and is the one most likely to be used in applications. We can provide information about the other routing types if you are interested. They are mainly used for system functions and (in the case of the nearest neighbour type) for flood-filling the machine with common code during the initialisation phase.

MANCHESTER 1824

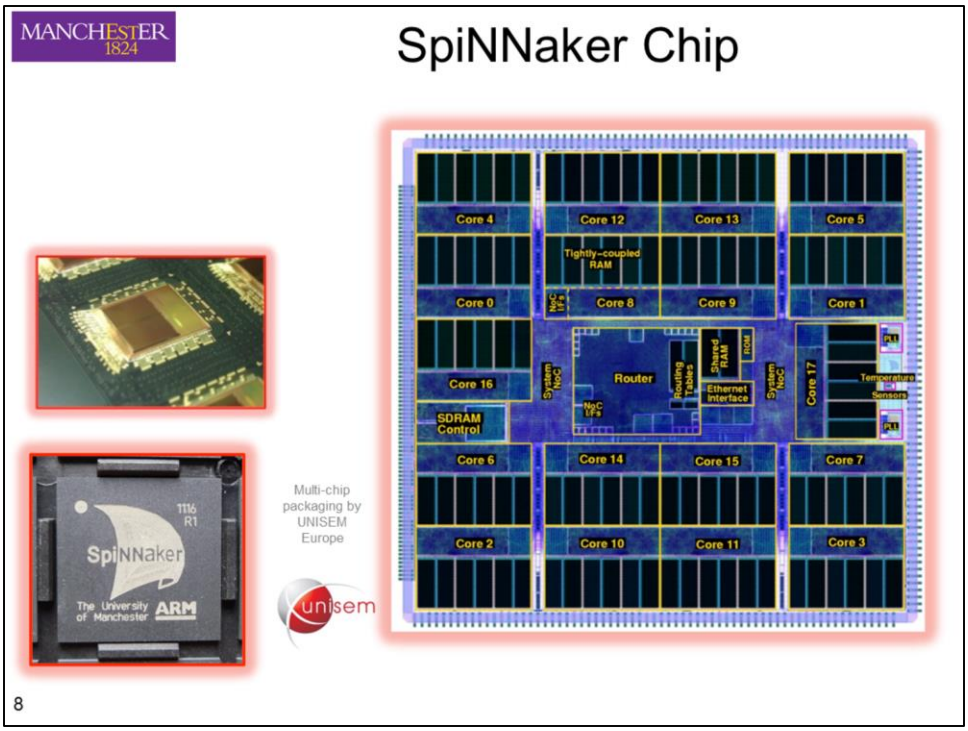
Multicast Routing

- Hardware router on each node
- Packets have a routing key
- Router has a look-up table of {*key, mask, data*} triplets
- If address matches a *key-mask* pair, the associated *data* tells router what to do with the packet

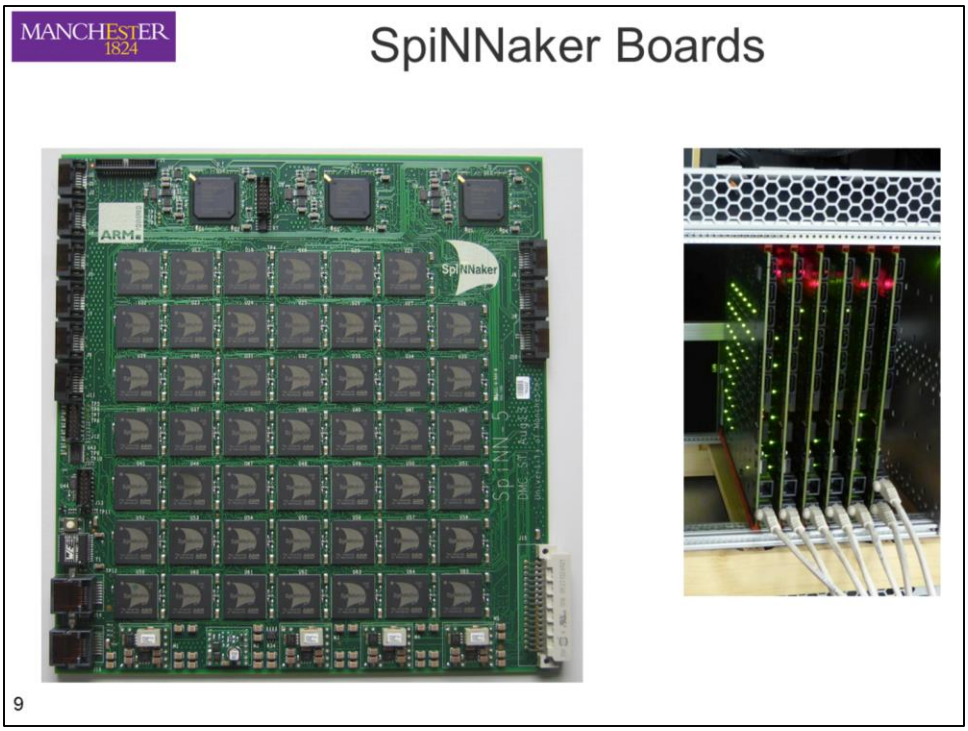


7

When a packet is generated by a core, it has a 32-bit routing key that identifies its source. The packet is given to the hardware router on its chip which decides what to do with it: to give it to one (or more!) cores on this chip or send it down one (or more!) of the six links to other chips. At each step on its journey the receiving router performs a look-up of the routing key in its 1024 entry table. The first match it finds is a ‘hit’ and it reads the associated data word in the table to see what action to take. If no match is found, the default behaviour is to send the packet out from the opposite link by which it entered.



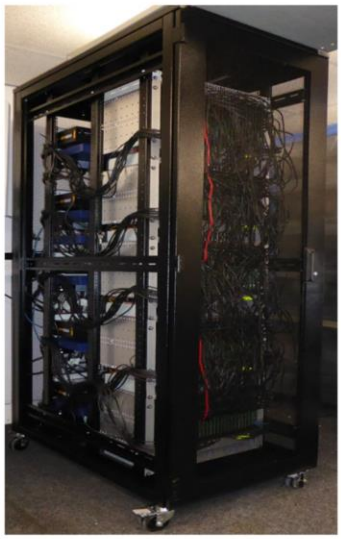
The SpiNNaker chip has 18 cores, a hardware router and an interface to 128MB of external SDRAM. All cores are identical. At boot time the first core to complete the boot process becomes the ‘monitor core’ and the next 16 become ‘application’ cores. The remaining 18th core may be non-functional as we used chips in which at least 17 cores are working (to improve the yield of useful chips!)



Each SpiNN-5 board (shown on the left) has 48 SpiNNaker chips and three FPGAs, which are used for board-to-board communications. The SpiNNaker links from each chip on the edge go via the FPGAS where they are translated into high-speed serial traffic, sent to the next board via SATA cables and then translated back into SpiNNaker link protocol. This is invisible to the packets themselves. When many bards are put together they form a *subrack* shown on the right.

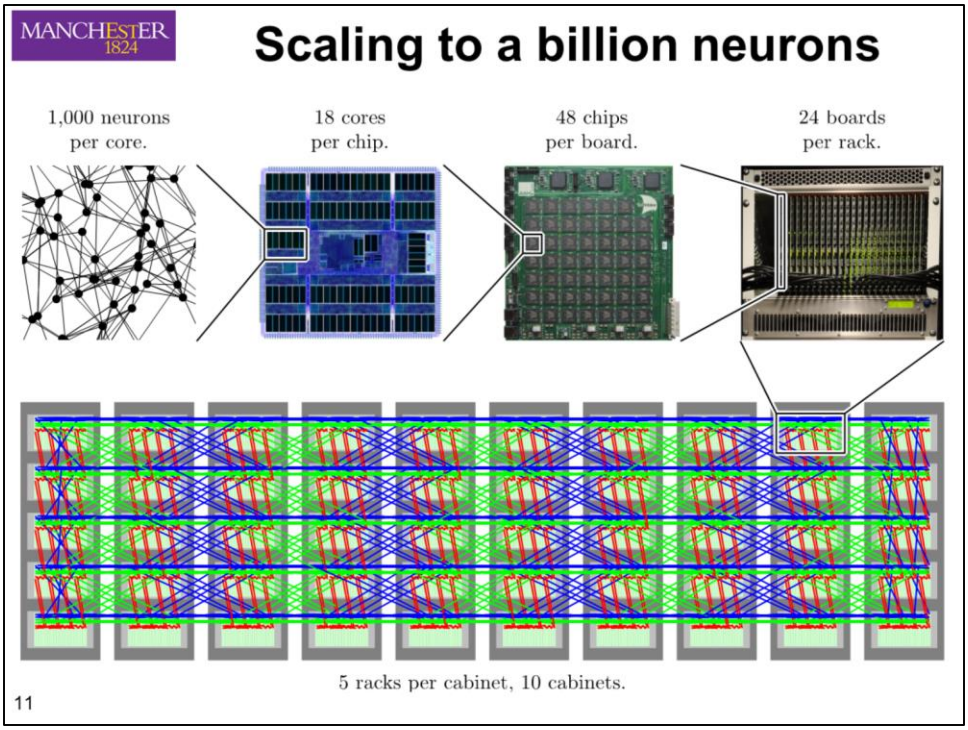
MANCHESTER 1824

SpiNNaker Machines



10

Each subrack (on the right) can hold up to 24 boards (1152 chips or 20K cores). Five subracks can be stacked to form a cabinet (on the right), containing 100K cores.



Connecting ten cabinets together into a single toroid gives us 1 million cores.

Back of the envelope calculations are that a 1 million core machine can simulation between 100 million and 1 billion simple leaky-integrate-and-fire neurons in real time.

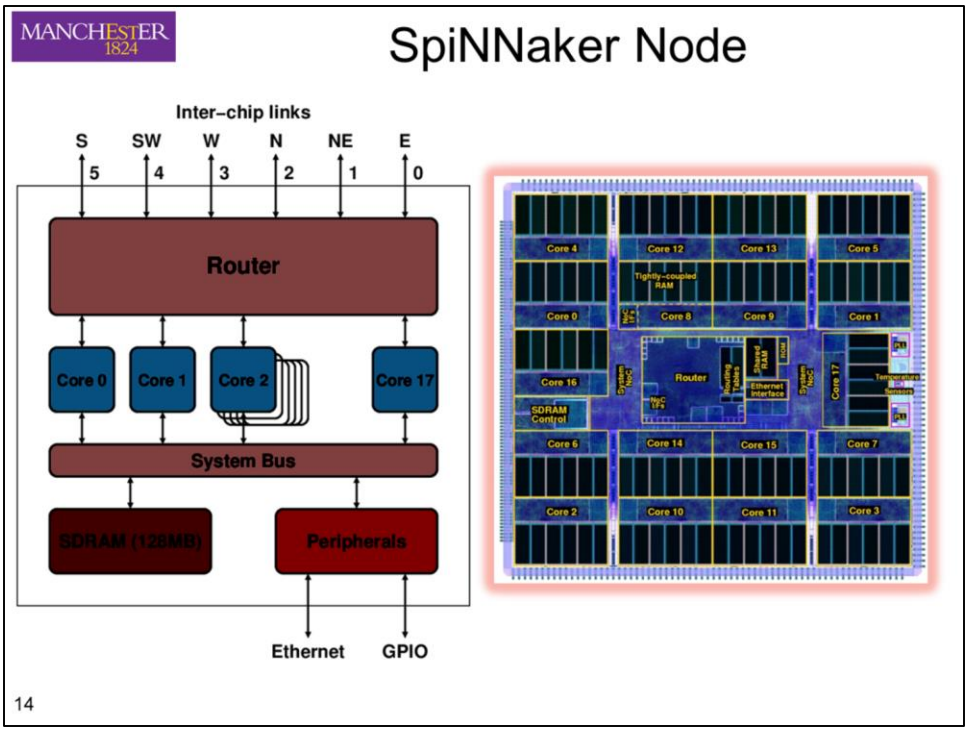
What Next for SpiNNaker?

- Five cabinet machine (500K ARM cores)
  - Will be online later this year
  - Will be available for remote access
    - Open to any research project, in principle
- SpiNNaker2 being developed within HBP
  - New systems by 2020?
- For further information contact:

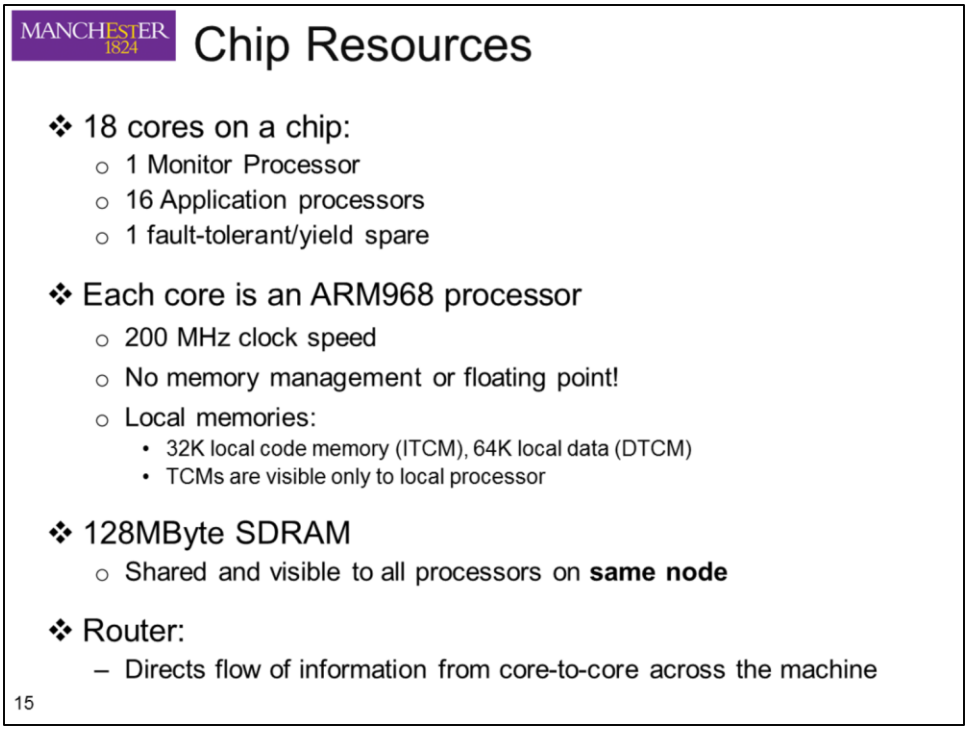
simon.davidson@manchester.ac.uk

MANCHESTER 1824

Chip Architecture



As discussed earlier, the 18 cores each share the hardware router (through which they can send packets through the links) and the shared 128MB memory. Each board can connect to a host via an Ethernet adapter.

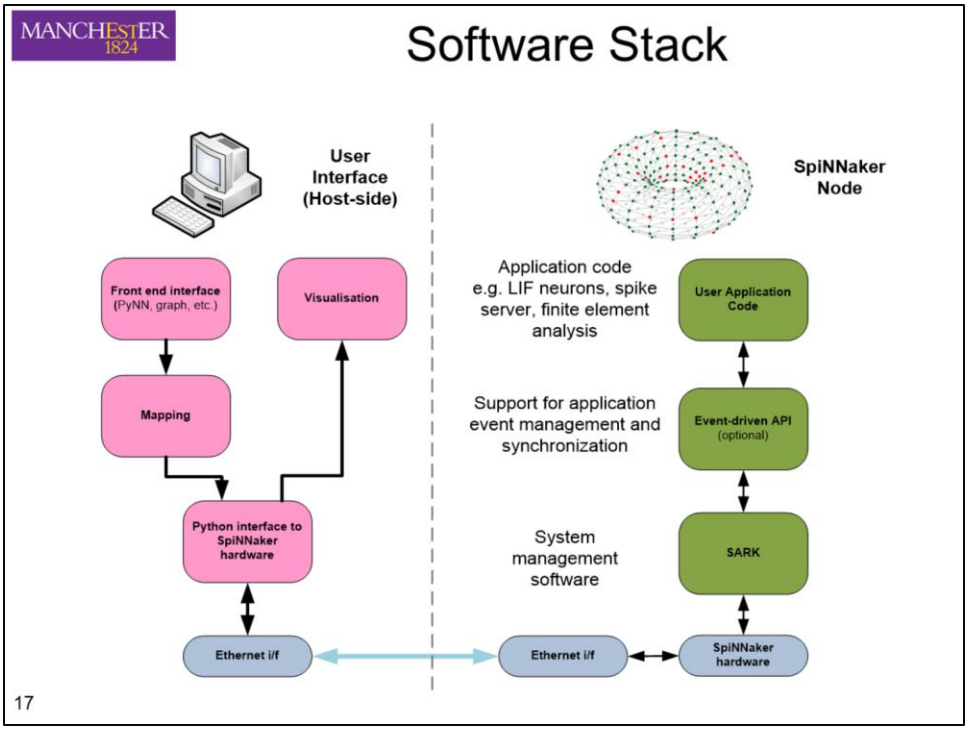


The local 64K data space can be accessed in a single cycle, whereas the shared 128MB memory takes dozens of cycles. Although the typical method of accessing this SDRAM is via DMA, the memory is mapped in the same address space as the DTCM and so it can, in principle be accessed using simple load and store instructions. There is a small, shared on-chip SRAM that has not been mentioned here. It is *mostly* used by the system for housekeeping functions and so is not generally available for applications.

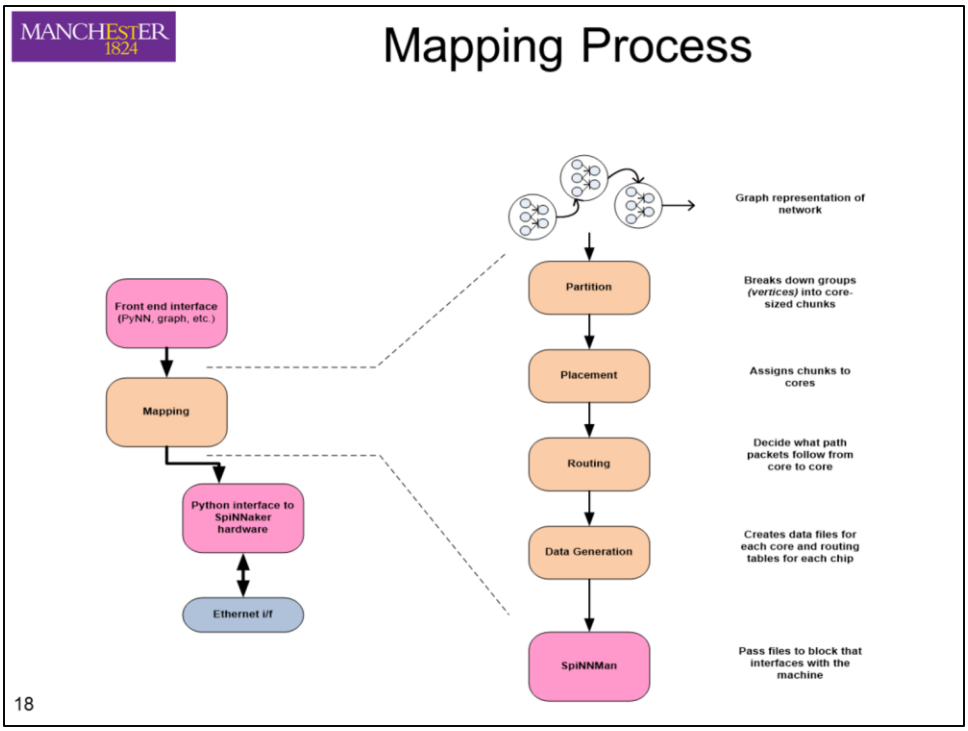
MANCHESTER 1824

Using SpiNNaker:

The Software Stack



Host-side the software is written in Python (pink boxes). On the machine, software is compile ‘c’ code (green boxes).
In our software stack, the user species their model in a domain-specific language (such as PyNN) which is translated into a graph like format in which computational elements are vertices and communication between these elements is represented by directed edges. The problem is mapped to the machine (see next slide) and the various files required for each chip and core are generated. This is loaded to SpiNNaker. A SpiNNaker application, running on many cores sits above an (optional) API and a system software layer (SARK). SARK provides essential resource management and comms functions. The API provides a framework for event-driven applications.



This slides zooms in on the mapping part of the host's activities. The user problem is translated into a graph, as described earlier. Each vertex represents some computation (e.g. a group of neurons) and this must be broken down into chunks that can be handled by a single core. This is *partitioning*. Each chunk of work is allocated to one of the cores on the target machine, in the *placement* phase. The edges of the graph, representing communication between these blocks of computation are translated into the routing of messages from one core to another. The output of this *routing* phase is a set of routing tables, one per chip, to be loaded to the machine. In the *data generation* phase the routing tables and any data required for each computation node are written.

What Files are Required for Simulation?

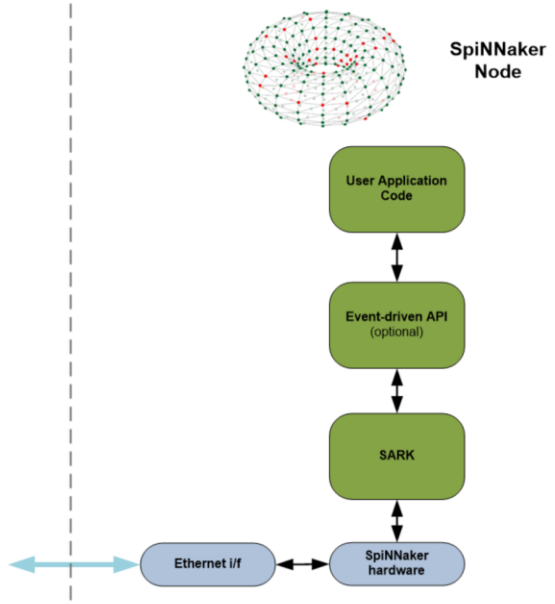
Files Required

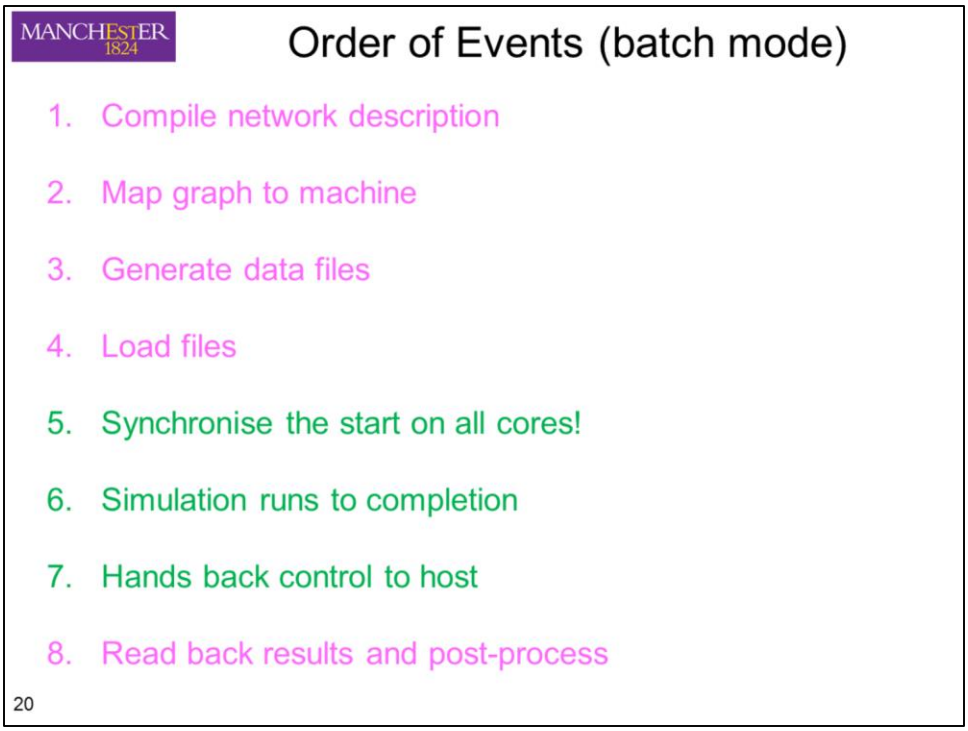
For Each Core (16 per chip):

- Application C executable (32KB)
- Application local data (64KB)
- Application shared data (8MB)

For Each Chip:

- Router table
- Any shared data tables





This describes the *batch* mode of operation, typical for running computational neuroscience models such as netorks written in PyNN. If SpiNNaker is used in a robotics environment, it may be set up to running continuously (i.e. without stopping),. It is possible to compile a network once, save all of the files and then re-load them whenever required, which is useful for robotics applications where the network does not change, but the data does.

MANCHESTER 1824

End of Overview!

- Much more detail on all of these topics
  - In the sessions to come….
- Any questions for now?
- Just one more thing to add….

Buying SpiNNaker Hardware

- 48-node board now available for sale
- Non-commercial use only
- 4-node boards can only be loaned (currently!)

- For further information contact: simon.davidson@manchester.ac.uk