

# Conclusions and next steps



Nick Brown  
EPCC University  
of Edinburgh



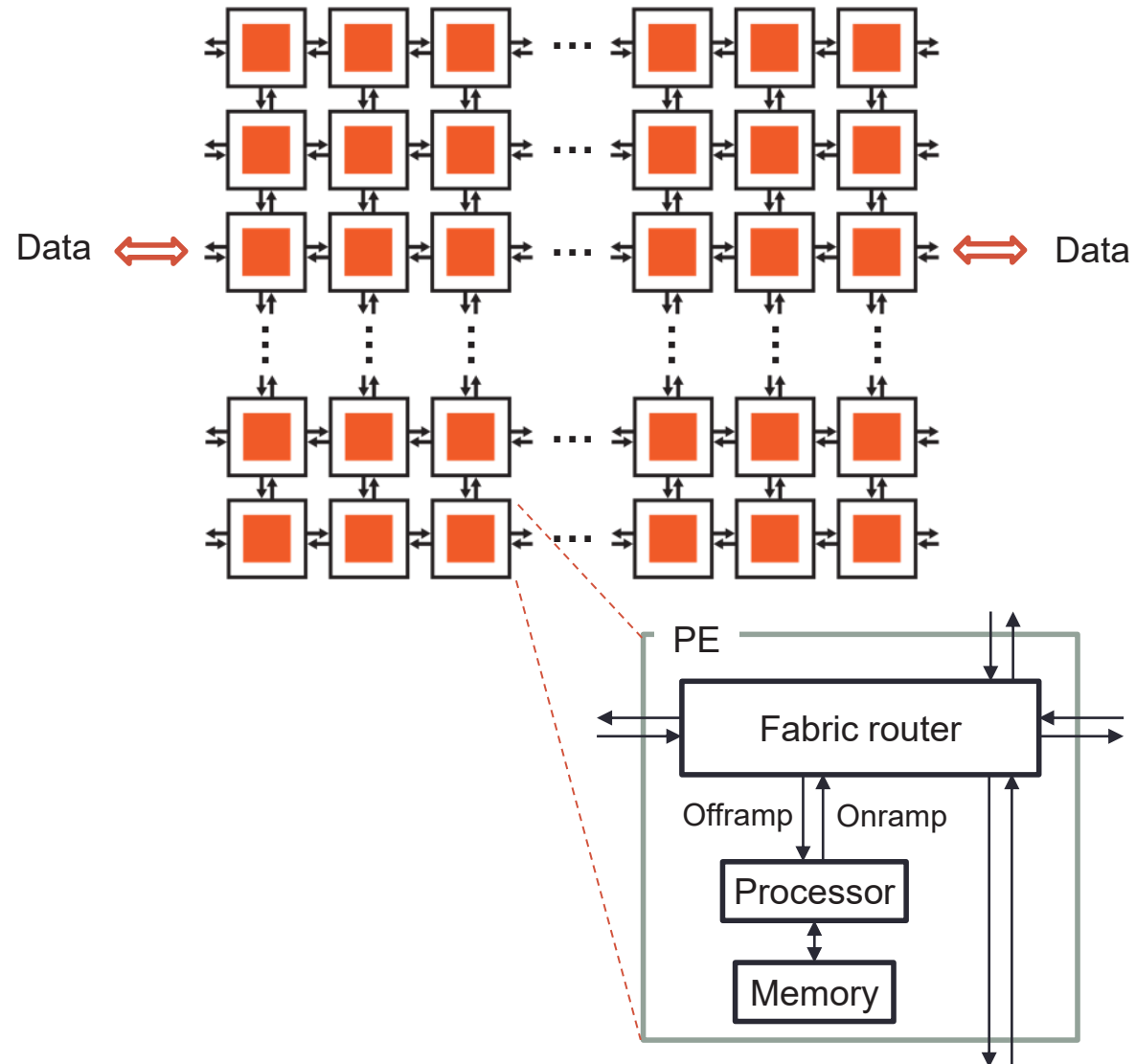
Joseph Lee  
EPCC University  
of Edinburgh



Leighton Wilson  
Cerebras Systems

# We hope you have enjoyed the tutorial....

- The Cerebras CS-2 is a powerful system with significant potential for HPC workloads
- We have seen that it is programmed using CSL, which is generally fairly similar to other languages but with some specifics for the architecture
- We have explored three hands-on activities and run these on a real CS-2 machine
  - And leveraged the simulator during development, which is standard practice on the CS-2



# Taking things further

- We have covered the basics here which should be enough for you to understand programming of the machine for multi-PE codes
  - But by necessity we couldn't fit everything in, and there are other aspects you can now explore
- More advanced topics
  - **Sentinels**
    - *Associates a task ID with routable colors for more routing flexibility*
  - **Switches**
    - *Runtime control of routes*
  - **Filters**
    - *Selectively consume wavelets*
  - **FIFOs**
    - *Hardware queues for buffering*
    - *Connecting FIFOs to host-device communication for streaming*
  - **More advanced task handling**
    - *For scheduling and managing tasks*
  - **Storage classes**
    - *Making variables visible outside the PE*
  - **Modules**
    - *For improved structuring of CSL codes*
  - **Map builtin**
    - *For performing custom operations on DSDs*

# Cerebras SDK libraries

- We explored the communications library, but there are others which you might find helpful
  - Similarly to the communications library, these provide standard API function calls that you can leverage from your code
- **Complex**
    - *For working with complex numbers*
  - **Debug**
    - *A tracing library for debugging CSL codes*
  - **Layout**
    - *Access to information about where the PE is located on the WSE*
  - **Malloc**
    - *Allocation for memory on the WSE*
  - **Math**
    - *Standard math functions on the WSE*
  - **Random**
    - *Random number generation*
  - **Tile\_config**
    - *Hardware configuration of the WSE*
  - **Time**
    - *Timestamping on the WSE*

# Example CSL programs are also available

**Repository:** <https://github.com/Cerebras/csl-examples>

- Introductory Tutorials
- GEMV
- GEMM
- Cholesky Decomposition
- 1D and 2D FFT
- 7-Point Stencil SpMV
- Power Method
- Conjugate Gradient
- Preconditioned Conjugate Gradient
- Finite Difference Stencil Computations
- Mandelbrot Set Generator
- Shift-Add Multiplication
- Hypersparse SpMV
- Histogram Computation

# Materials and the CS-2 community

- All materials for this tutorial are open source and can be found at
  - <https://github.com/EPCed/cs2-sdk-training>
- There is lots of documentation on the Cerebras website on CSL and the SDK
  - <https://sdk.cerebras.net/csl>
  - <https://github.com/Cerebras/csl-examples>
- There is a CS-2 developer community that you can join
  - Roughly monthly meetings
  - Forums: [discourse.cerebras.net](https://discourse.cerebras.net)
- The SDK is free to download, meaning you can have a copy of the compiler and simulator on your own local machine



# Going forwards...

- We are very happy to chat with you about your own codes on the CS-2 and offer any advice that we can here
- The visitor accounts will stay open on the EPCC CS-2, so you can continue to experiment with the codes and run on the machine
  - And remember you can install the SDK locally too
- There are also cloud CS-2s hosted by Cerebras

