MANCHESTER 1824
The University of Manchester

TECHNISCHE UNIVERSITÄT DRESDEN

Approximate Fixed-Point Elementary Function Accelerator for the SpiNNaker-2 Neuromorphic Chip

Mantas Mikaitis, PhD student @ University of Manchester, UK
mantas.mikaitis@manchester.ac.uk

25th IEEE Symposium on Computer Arithmetic
Amherst, MA, USA, June 2018

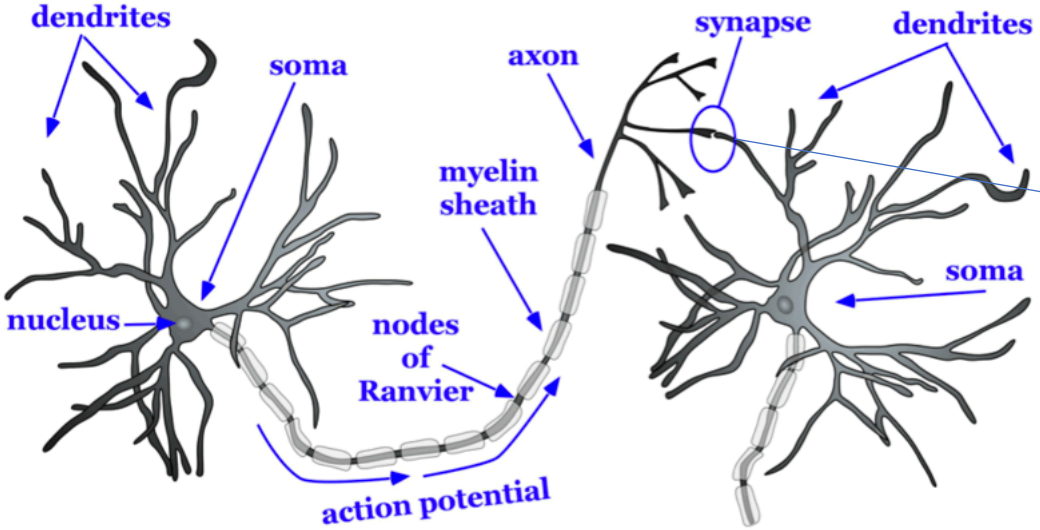
SpiNNaker2
Universal Spiking Neural Network Architecture

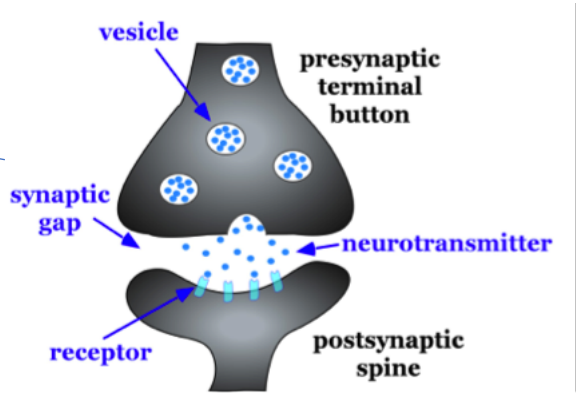
Human Brain Project

Arithmetic in neuromorphic chips

- Neuromorphic chips are designed to simulate **Spiking-Neural-Networks** – very biologically realistic models of neurons and synapses.
- The main question is: How much bits do we need for arithmetic operations in neuromorphic hardware?
- Fixed- or floating-point?
- How much bits is enough to simulate the brain? (Brain as defined by computational neuroscientists - not just application specific deep learning, machine learning etc.)
- For this work we chose: **Fixed-point, internal 39-bits with programmable approximation to 32-bit output.**

Motivation: Why accelerate exponential function?





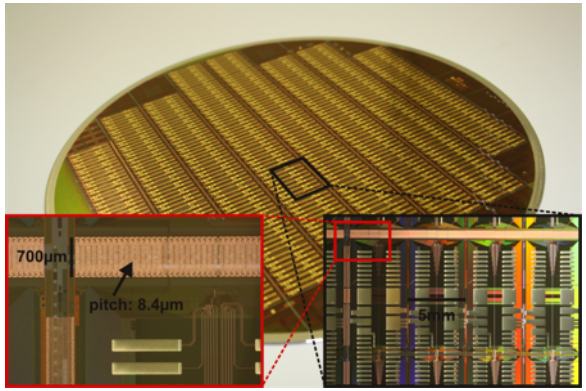
$LIF\ neuron\ membrane\ voltage\ is\ modelled\ as\colon \frac{dV}{dt} = \frac{-[u(t) - u_{rest}] + RI(t)}{\tau_m}$

Similar equations are derived for describing ion channel opening/closing, intrinsic neuron current activation/deactivation and plasticity of the synaptic gap (to change the weight in learning).

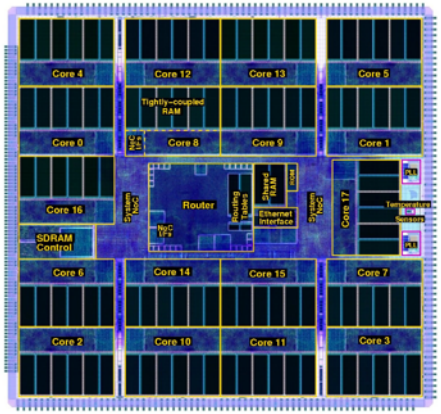
Energy/memory/delay is significant using soft-exponential (decay)!

Images: Furtak, S. (2018). Neurons. In R. Biswas-Diener & E. Diener (Eds), *Noba textbook series: Psychology*

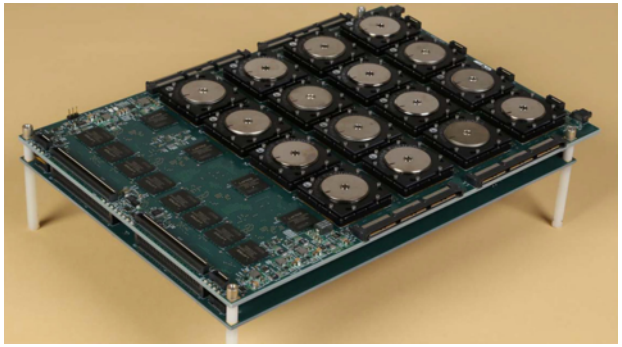
Neuromorphic chips



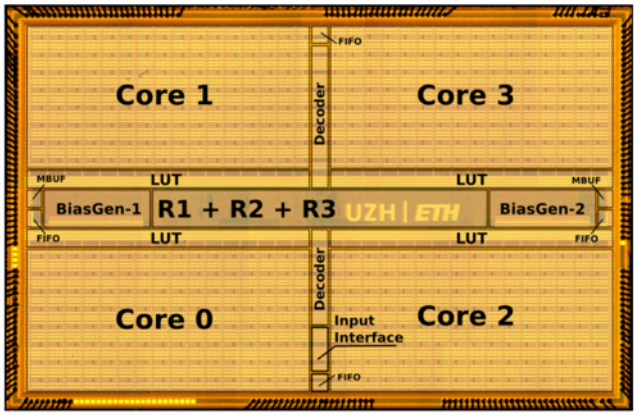
Brainscales (2011)



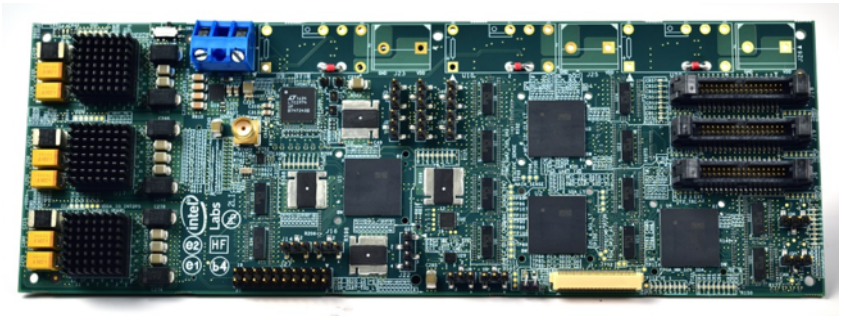
SpiNNaker (2011)



IBM TrueNorth (2014)

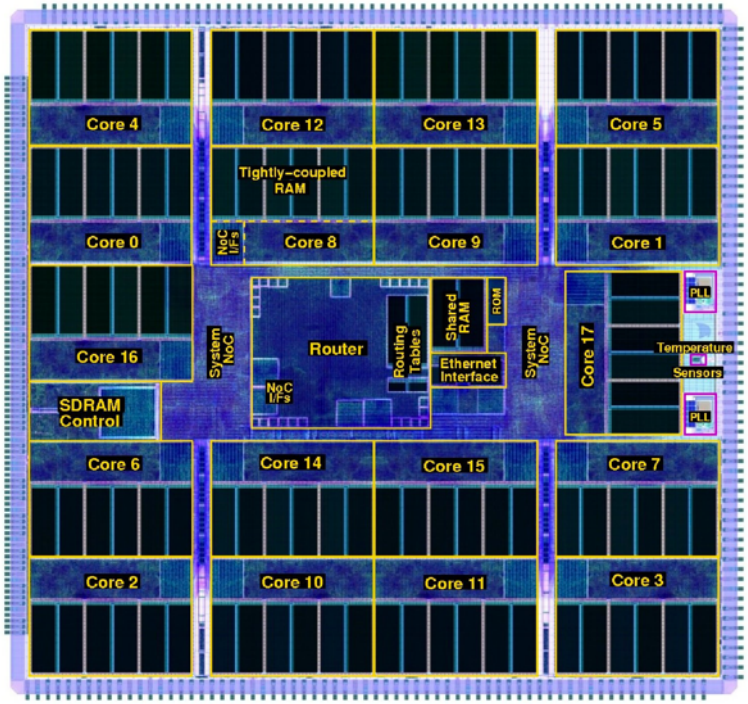


DYNAP (2018)

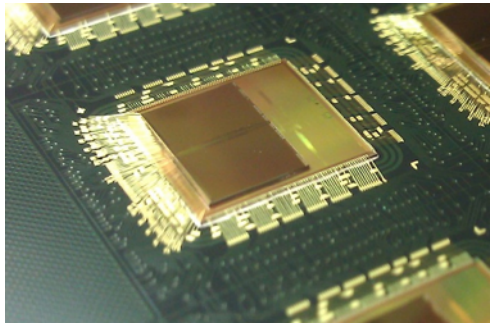


Intel Loihi (2018)

SpiNNaker (Manchester, 2011)



- 18 ARM968 cores
- 96K memory per core
- 128MB Off-chip memory
- 1W power
- Fixed-point arithmetic (GCC implementation of ISO 18037)
- 95 cycle soft-exponential



SpiNNaker-2 (Manchester, Dresden, 2020)

- 144 ARM M4F cores
- 128K memory per core (With capability to use other core's memories)
- ~2GB Off-chip memory
- Single precision floating point hardware unit
- Random Number Generators
- Machine Learning Accelerator
- 1W power (+power management based on neural network activity)
- exp and log (base e) accelerators (x144)

Image: Sebastian Höppner, SpiNNaker2, NICE2018

Most used functions in SpiNNaker

- Exponential decay e^{-x}
- Random number generation
- Reciprocal 1/x (E.g. sigmoid activation/deactivation function)
- Multiply-accumulate for ODE solvers

Proposed method for arithmetic in SpiNNaker-2:

Use fixed-point arithmetic when building accelerators – at least 4x less area/energy than floating-point*.
Use floating-point unit in ARM M4F only for accuracy sensitive models (Complex neuron ODE).
Use fixed-point arithmetic everywhere else (+ accelerators and DSP instruction set)

*Data from: “High-Performance Hardware for Machine Learning”, W. Dally, U.C. Berkeley, 2016

Well known shift-and-add algorithm for exp/log*

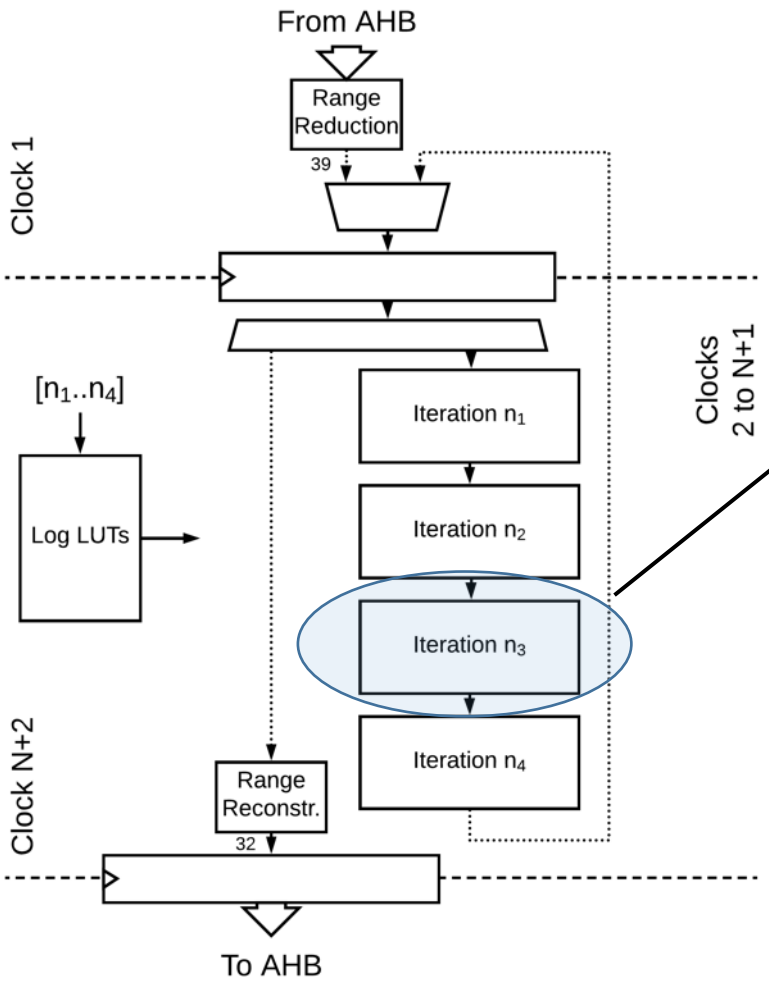
$$L_{n+1} = L_n - \ln(1 + d_n 2^{-n})$$
$$E_{n+1} = E_n + d_n E_n 2^{-n}$$

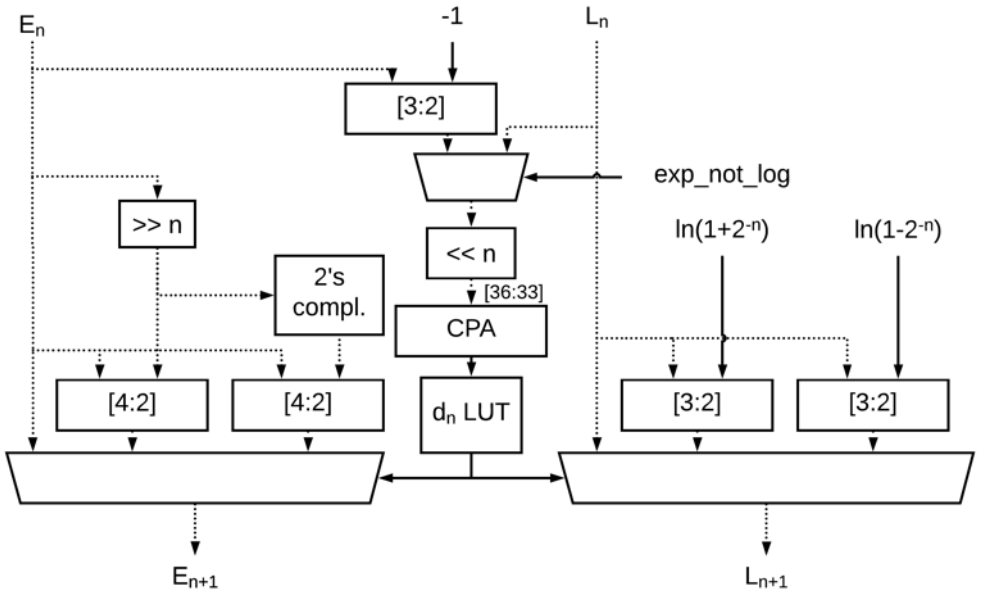
Mode	e^x	$log_e(x)$
Next iteration control	$d_n = \begin{cases} -1 \; if & 2^n L_n \le -\frac{3}{2} \\ 0 \quad if & -1 \le 2^n L_n \le -\frac{1}{2} \\ 1 \quad if & 2^n L_n \ge 0 \end{cases}$	$d_n = \begin{cases} -1 \; if & 2^n(E_n - 1) \le -1 \\ 0 \quad if & -\frac{1}{2} \le 2^n(E_n - 1) \le 0 \\ 1 \quad if & 2^n(E_n - 1) \ge \frac{1}{2} \end{cases}$

With correct initialization, n iterations produce n-1 significant bits approximation.

* Elementary Functions – Algorithms and Implementation 3rd ed., Muller, 2016

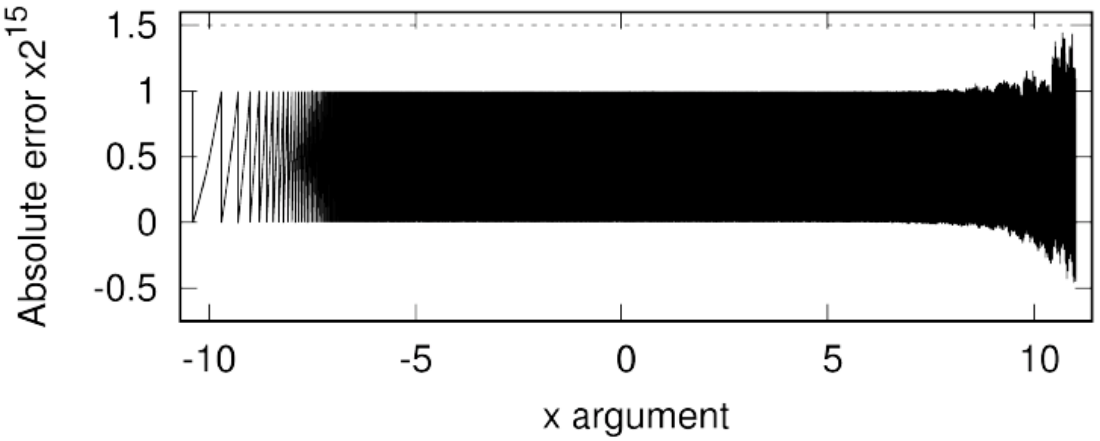
Implementation

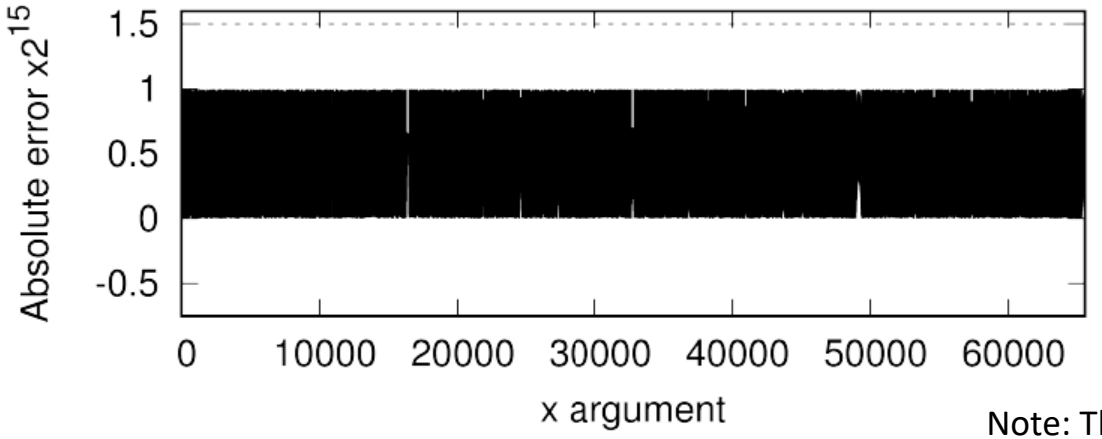




- Critical path strained in order to run more iterations per cycle.
- Iteration parallelized as much as possible – precalculate next iteration (all possible next values) while choosing d_n.

Results: Accuracy and monotonicity s16.15 format





- Full accuracy (8 loops)
- Top: exp in the domain [-10.4, 11.1]
- Bottom: log in the full domain
- Accuracy: 3 neighbouring values around C double-precision sample.

Note: This result and further are obtained by comparing to math.h double precision exp() (error = result(x) – exp(x))

Results: Accuracy and monotonicity s16.15 format

Each case has reduced accuracy by running less loops (Where each loop gives approximately 4 bits of answer).

$\epsilon = 2^{-15} = 0.000030517578125$

	e^x		$log_e(x)$	
N	Max abs.err.	Monotonic	Max abs.err.	Monotonic
8	0.00004425	Yes	0.00003082	Yes
7	0.00023559	Yes	0.00003082	Yes
6	0.00387969	Yes	0.00003082	Yes
5	0.06096649	Yes	0.00003112	Yes
4	0.99264343	Yes	0.00004089	No
3	15.3052932	No	0.00019928	No
2	241.053592	No	0.00268463	No
1	3352.69732	No	0.03837280	No

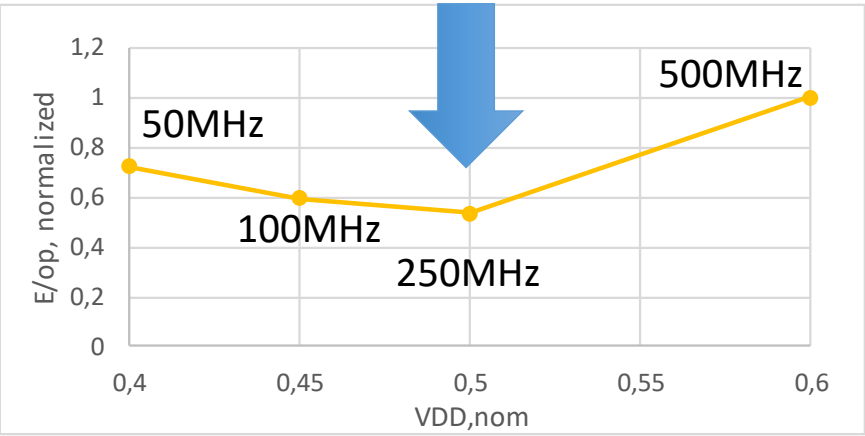
Results: Accuracy and monotonicity s0.31 format

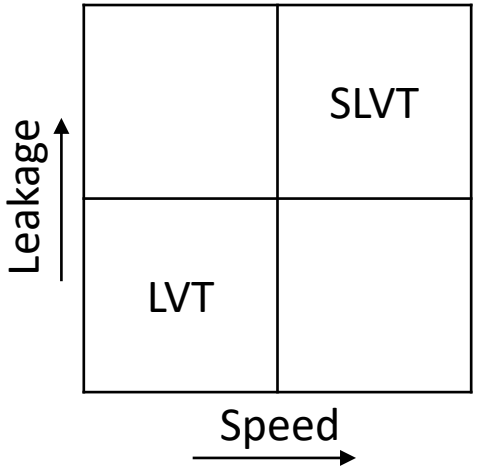
Each case has reduced accuracy by running less loops (Where each loop gives approximately 4 bits of answer).
$\epsilon = 2^{-31} = 0{:}000000000465661$

	e^x		$log_e(x)$	
N	Max abs.err.	Monotonic	Max abs.err.	Monotonic
8	0.000000000722	Yes	0.000000001387	Yes
7	0.000000003744	No	0.000000003613	Yes
6	0.000000059274	No	0.000000040312	Yes
5	0.000000945120	No	0.000000645976	No
4	0.000014910344	No	0.000010420316	No
3	0.000236990545	No	0.000170091129	No
2	0.003536022179	No	0.002655907041	No
1	0.045793333569	No	0.038344341439	No

Technology and Implementation Strategy

- GLOBALFOUNDRIES 22FDX (FDSOI) technology [1]
- Adaptive body biasing (ABB) solution and foundation IP by Dresden Spinoff Racyics [2] → Enables operation down to 0.40V (0.36V wc)
- Forward Body Bias Scheme with Low-VT (LVT) and Super-low-VT (SLVT) flavors.
- Power performance area (PPA) studies for neuromorphic application scenarios





Target implementation point for maximum energy efficiency at nominally **0.50V** and **250MHz** (worst case 0.45V and 0C)

Source: Sebastian Höppner, SpiNNaker2, NICE2018

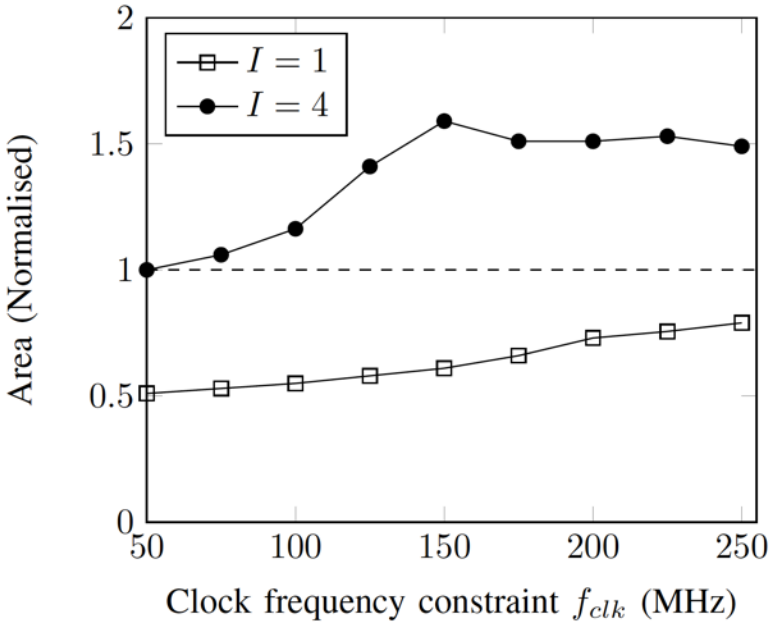
Results: Synthesis and timing analysis

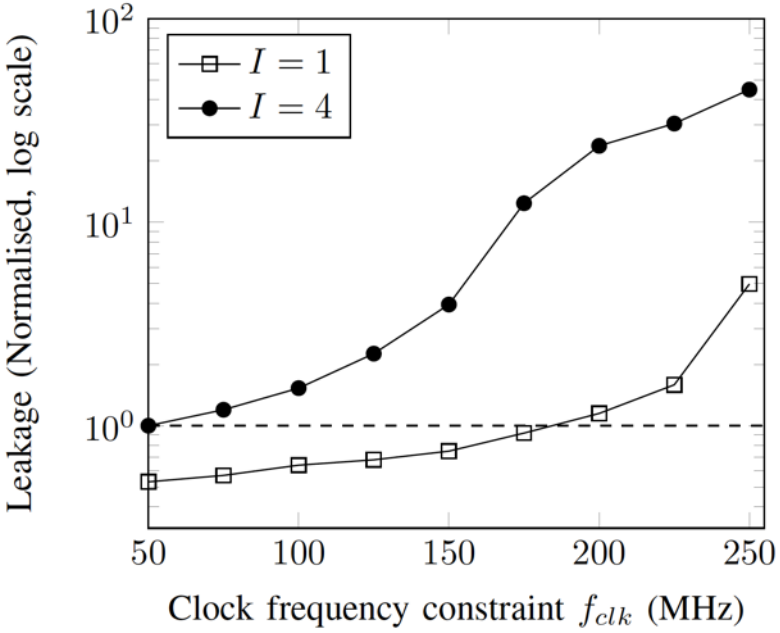
6 accelerator versions are covered with varying number of iteration hardware units instantiated, denoted by variable I.

Iterations per cycle, I	Area (μm^2)	SLVT cells	Timing met	Max latency (cycles/op)
1	6108	7.3%	Y	34
2	8755	3.1%	Y	18
3	10361	21.2%	Y	13
4	11524	36%	Y	10
6	17368	59.6%	Y	8
8	21893	68.7%	N	6

Results: Synthesis and timing analysis

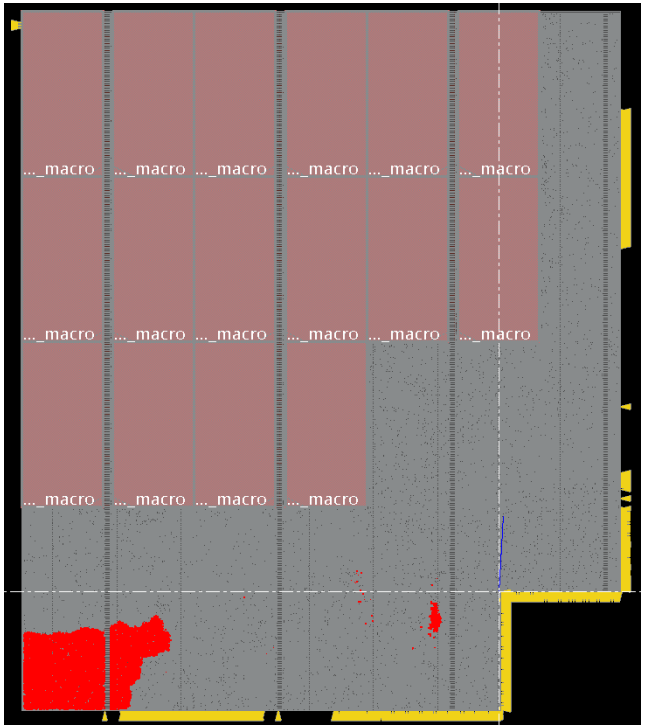
- Same conditions as before, but now varying the clock frequency constraint.
- Area and leakage is measured for two units with I=1 and I=4.





Results: Place and route

- Full processing element is shown after P&R with the accelerator in red.
- The power consumption of the circuit is analysed in a typical process condition at worst case power conditions of 0.5V at 85C.
- Software testcases on a netlist of the PE show 0.16-0.39nJ/exp energy (depending on the level of approximation)
- 56x-325x lower EDP than SpiNNaker software exp.



	Exp accelerator	Software exp
Throughput	20.8-50M exp/s	2.6M exp/s
Latency	5-12 cycles/exp*	95 cycles/exp
Energy per exp	0.16 nJ/exp- 0.39 nJ/exp	2.74 nJ/exp
Total area	5928 μm^2	-

Conclusion

- Accelerator with almost full accuracy in fixed-point s16.15 and s0.31 formats was presented.
- Approximation control for experimenting with accuracy was explored.
- The prototype chip is currently in manufacturing. The chips will arrive in the lab later in 2018.
- Iterative algorithms cause challenges for tighter timing constraints due to very sequential nature.
- We have discussed how to parallelize a single iteration module, but leakage is still a problem if more than 2 iterations are placed in a clock cycle.
- Unit with 4 iterations is quite a good design point for power, area and ops/s.

Further work

Exponential/logarithm unit:

- Floating-point conversion
- Rounding; higher radix shift-and-add; programmable fixed-point format.
- We have another exponential function design using LUTs and polynomial approximation – comparison of two approaches in 22nm.
- How to parallelize shift-add algorithms further?

Other neuromorphic arithmetic for saving energy/memory:

- Stochastic rounding (allows smaller precision arithmetic without loss of accuracy in some applications)
- Approximate arithmetic with errors in the circuit (leverage error tolerance of neuromorphic applications)

Extra references

Images of the chips:

- https://newsroom.intel.com/editorials/intel-creates-neuromorphic-research-community/
- http://www.artificialbrains.com/brainscales
- https://en.wikipedia.org/wiki/TrueNorth
- https://ai-ctx.com/products/dynap/
- http://niceworkshop.org/wp-content/uploads/2018/05/2-27-SHoppner-SpiNNaker2.pdf

Implementation technology:

[1] R. Carter et al., "22nm FDSOI technology for emerging mobile, Internet-of-Things, and RF applications," 2016 IEEE International Electron Devices Meeting (IEDM), San Francisco, CA, 2016, pp. 2.2.1-2.2.4. doi: 10.1109/IEDM.2016.7838029

[2] www.makeChip.design

Acknowledgements

Co-authors: Dave Lester, Delong Shang, Steve Furber, Gengting Liu, Jim Garside, Stefan Szholze, Sebastian Höppner, Andreas Dixius

Also thanks for the useful discussions: Felix Neumärker, Johannes Partzsch, Dongwei Hu, Michael Hopkins.

And many thanks to the reviewers of ARITH25 for their feedback.