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
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: \[
\frac{dV}{dt} = \frac{-[u(t) - u_{rest}] + RI(t)}{\tau_m}
\]

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)!

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

Neuron models of size equivalent to 1% of human brain
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)
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)

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

\[ L_{n+1} = L_n - \ln(1 + d_n 2^{-n}) \]

\[ E_{n+1} = E_n + d_n E_n 2^{-n} \]

<table>
<thead>
<tr>
<th>Mode</th>
<th>( e^x )</th>
<th>( \log_e(x) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Next iteration control</td>
<td>( d_n = \begin{cases} -1 &amp; \text{if } 2^n L_n \leq -\frac{3}{2} \ 0 &amp; \text{if } -1 \leq 2^n L_n \leq -\frac{1}{2} \ 1 &amp; \text{if } 2^n L_n \geq 0 \end{cases} )</td>
<td>( d_n = \begin{cases} -1 &amp; \text{if } 2^n (E_n - 1) \leq -1 \ 0 &amp; \text{if } -\frac{1}{2} \leq 2^n (E_n - 1) \leq 0 \ 1 &amp; \text{if } 2^n (E_n - 1) \geq \frac{1}{2} \end{cases} )</td>
</tr>
</tbody>
</table>

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

* Elementary Functions – Algorithms and Implementation 3\textsuperscript{rd} 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).

\[ \varepsilon = 2^{-15} = 0.000030517578125 \]

<table>
<thead>
<tr>
<th>N</th>
<th>Max abs.err.</th>
<th>Monotonic</th>
<th>Max abs.err.</th>
<th>Monotonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.00004425</td>
<td>Yes</td>
<td>0.00003082</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>0.00023559</td>
<td>Yes</td>
<td>0.00003082</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>0.00387969</td>
<td>Yes</td>
<td>0.00003082</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>0.06096649</td>
<td>Yes</td>
<td>0.00003112</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>0.99264343</td>
<td>Yes</td>
<td>0.00004089</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>15.3052932</td>
<td>No</td>
<td>0.00019928</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>241.053592</td>
<td>No</td>
<td>0.00268463</td>
<td>No</td>
</tr>
<tr>
<td>1</td>
<td>3352.69732</td>
<td>No</td>
<td>0.03837280</td>
<td>No</td>
</tr>
</tbody>
</table>
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).
\[ \epsilon = 2^{-31} = 0.000000000465661 \]

<table>
<thead>
<tr>
<th>N</th>
<th>Max abs.err.</th>
<th>Monotonic</th>
<th>Max abs.err.</th>
<th>Monotonic</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>0.0000000000722</td>
<td>Yes</td>
<td>0.000000001387</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>0.0000000003744</td>
<td>No</td>
<td>0.000000003613</td>
<td>Yes</td>
</tr>
<tr>
<td>6</td>
<td>0.0000000059274</td>
<td>No</td>
<td>0.0000000040312</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>0.000000945120</td>
<td>No</td>
<td>0.000000645976</td>
<td>No</td>
</tr>
<tr>
<td>4</td>
<td>0.00014910344</td>
<td>No</td>
<td>0.00010420316</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>0.000236990545</td>
<td>No</td>
<td>0.00170091129</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>0.003536022179</td>
<td>No</td>
<td>0.002655907041</td>
<td>No</td>
</tr>
<tr>
<td>1</td>
<td>0.045793333569</td>
<td>No</td>
<td>0.038344341439</td>
<td>No</td>
</tr>
</tbody>
</table>
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 0°C)

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 $I$.

<table>
<thead>
<tr>
<th>Iterations per cycle, $I$</th>
<th>Area ($\mu m^2$)</th>
<th>SLVT cells</th>
<th>Timing met</th>
<th>Max latency (cycles/op)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6108</td>
<td>7.3%</td>
<td>Y</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>8755</td>
<td>3.1%</td>
<td>Y</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>10361</td>
<td>21.2%</td>
<td>Y</td>
<td>13</td>
</tr>
<tr>
<td>4</td>
<td>11524</td>
<td>36%</td>
<td>Y</td>
<td>10</td>
</tr>
<tr>
<td>6</td>
<td>17368</td>
<td>59.6%</td>
<td>Y</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>21893</td>
<td>68.7%</td>
<td>N</td>
<td>6</td>
</tr>
</tbody>
</table>
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 $I=4$. 

![Graph showing area and leakage vs. clock frequency constraint for $I=1$ 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.

<table>
<thead>
<tr>
<th></th>
<th>Exp accelerator</th>
<th>Software exp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput</td>
<td>20.8-50M exp/s</td>
<td>2.6M exp/s</td>
</tr>
<tr>
<td>Latency</td>
<td>5-12 cycles/exp*</td>
<td>95 cycles/exp</td>
</tr>
<tr>
<td>Energy per exp</td>
<td>0.16 nJ/exp</td>
<td>2.74 nJ/exp</td>
</tr>
<tr>
<td></td>
<td>0.39 nJ/exp</td>
<td></td>
</tr>
<tr>
<td>Total area</td>
<td>5928 µm²</td>
<td></td>
</tr>
</tbody>
</table>
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:
- http://www.artificialbrains.com/brainscales
- https://ai-ctx.com/products/dynap/

Implementation technology:


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.