Event management for large scale event-driven digital hardware spiking neural networks

Louis-Charles Caron*, Michiel D’Haeneb, Frédéric Mailhotc, Benjamin Schrauwen, Jean Rouata

* NECOTIS, Université de Sherbrooke, 2500 boul. de l’Université, Sherbrooke (Québec) Canada  * email: lcaron@ensta.fr
b Reservoir Lab, Ghent University, Sint Pieterenieuwstraat 41, 9000, Ghent, Belgium

event-driven approach to SNNs

Spiking neural networks (SNNs) can take many forms and involve various implementation strategies and optimizations. Two different approaches are widely used and opposed: time-driven (see Algorithm 1) and event-driven (see Algorithm 2) [1].

Algorithm 1 Time-driven implementation
1. repeat
2. move one step forward in time
3. for each neuron in the network do
4. update state
5. end for
6. until desired time is reached

Algorithm 2 Event-driven implementation
1. repeat
2. move one step forward in time
3. for each neuron involved in the event do
4. update state
5. end for
6. find next event to happen
7. until desired time is reached

Event-driven SNNs
- Time steps fit the occurrence of events
- + no event = no processing
- + temporal precision
- Neurons involved in an event are updated
- + few computations
- Inverse neuron model - better with simple SNNs
- Need to identify the next neuron to fire - event management

Digital hardware event management

Most digital hardware SNNs implement a hybrid time- and event-driven system, without any form of event management, because hardware solutions for event management scale badly with the number of events. Similar to Agis et al. [2], we strictly follow the pattern of Algorithm 2, but our Structured Heap Queue (SHQ) [3] has \(O(1)\) complexity in time.

![Diagram of the Structured Heap Queue (SHQ)](image)

Memory-optimized SHQ

Complexity in logic \(O(\log(N))\)
Complexity in memory \(O(N)\)
Complexity in time for
- Delete top element \(O(1)\)
- Delete any element \(O(1)\)
- Insert \(O(1)\)
- Read top element \(O(1)\)
- Read any element \(O(\log(N))\)

Scaling, with N the number of events

SHQ vs Agis et al. (2007)

<table>
<thead>
<tr>
<th>Method</th>
<th>Hardware</th>
<th>Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agis et al. [2]</td>
<td>Unsorted list and on-the-fly search for the next event to happen</td>
<td>10 cc with fewer than 2048 events, up to 69 cc with 16384 events</td>
</tr>
<tr>
<td>Caron et al. [3, 4]</td>
<td>Structured Heap Queue (SHQ), a variation of the pipelined heap queue [5]</td>
<td>7 clock cycles</td>
</tr>
</tbody>
</table>

Digital hardware: tailor your system to your needs by trading logic for time and memory

Structured heap queue: get a constant processing time, no matter the size of your SNN

Summary

Spiking neural network: use your own SNN
Event-driven: save resources by only computing values you are interested in

Digital hardware: FPGA implementation

Image segmentation on Xilinx XC5VSX50T FPGA using the Oscillatory Dynamic Link Matcher (ODLM) signal processing SNN [6].

- 65 536 neurons, 513 184 synapses network
- Segmentation of a 406×158-pixel image in 200ms @100MHz (takes 1.9s on an Intel Core i5 @2.4GHz, 3GB RAM)
- ~1 250 000 events / second

![Diagram of the FPGA implementation](image)

FUNCTION

<table>
<thead>
<tr>
<th>Controller</th>
<th>FFs</th>
<th>LUTs</th>
<th>BRAMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processing element</td>
<td>5033</td>
<td>635</td>
<td>0</td>
</tr>
<tr>
<td>Topology solver</td>
<td>19</td>
<td>73</td>
<td>50</td>
</tr>
<tr>
<td>Neuron memory</td>
<td>16</td>
<td>26</td>
<td>0</td>
</tr>
<tr>
<td>Weight calculator</td>
<td>1</td>
<td>1</td>
<td>44</td>
</tr>
<tr>
<td>Membrane model</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Synapse model</td>
<td>1</td>
<td>14</td>
<td>3</td>
</tr>
<tr>
<td>Inverse membrane model</td>
<td>0</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>Event queue</td>
<td>2 846</td>
<td>3 965</td>
<td>80</td>
</tr>
</tbody>
</table>

Digital hardware event management