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

Louis-Charles Caron[a]*, Michiel D'Haene[b], Frédéric Mailhot[a], Benjamin Schrauwen[b], Jean Rouat[a]
[a] NECOTIS, Université de Sherbrooke, 2500 boul. de l'Université, Sherbrooke (Québec) Canada * email: lcaron@ensta.fr
[b] Reservoir Lab, Ghent University, Sint Pietersnieuwstraat 41, 9000, Ghent, Belgium

UNIVERSITÉ DE SHERBROOKE

UNIVERSITEIT GENT

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

```
repeat
    move one step forward in time
    for each neuron in the network do
        update state
    end for
until desired time is reached
```

Algorithm 2 Event-driven implementation

```
repeat
    move one event forward in time
    for each neuron involved in the event do
        update state
    end for
    find next event to happen
until desired time is reached
```

Event-driven SNNs

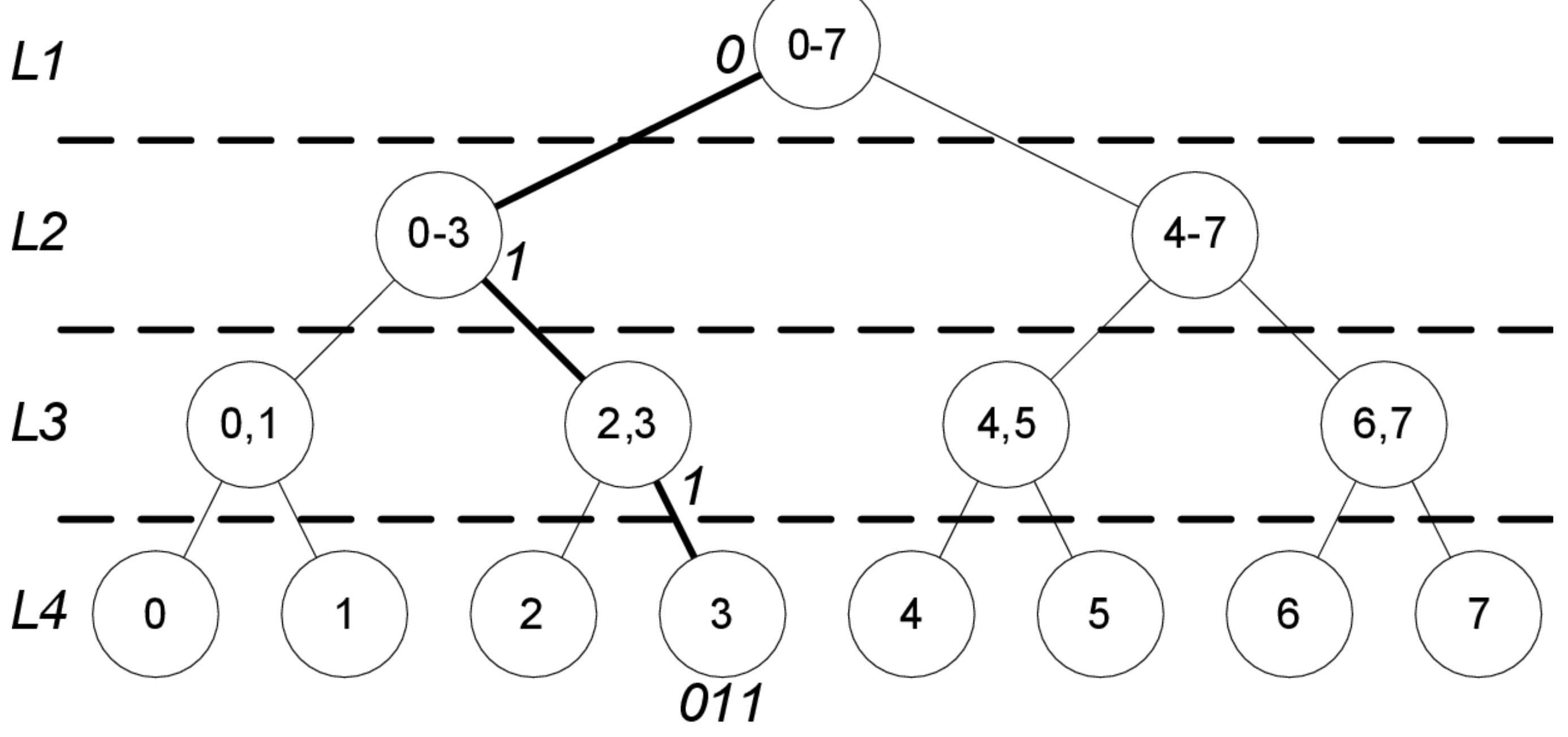
- ➢ Time steps fit the occurrence of events
  - + no event = no processing
  - + temporal precision
- ➢ Neurons involved in an event are updated
  - + **few computations**
- ➢ Inverse neuron model
  - - better with simple SNNs
- ➢ Need to identify the next neuron to fire
  - - **event management**

Digital hardware event management

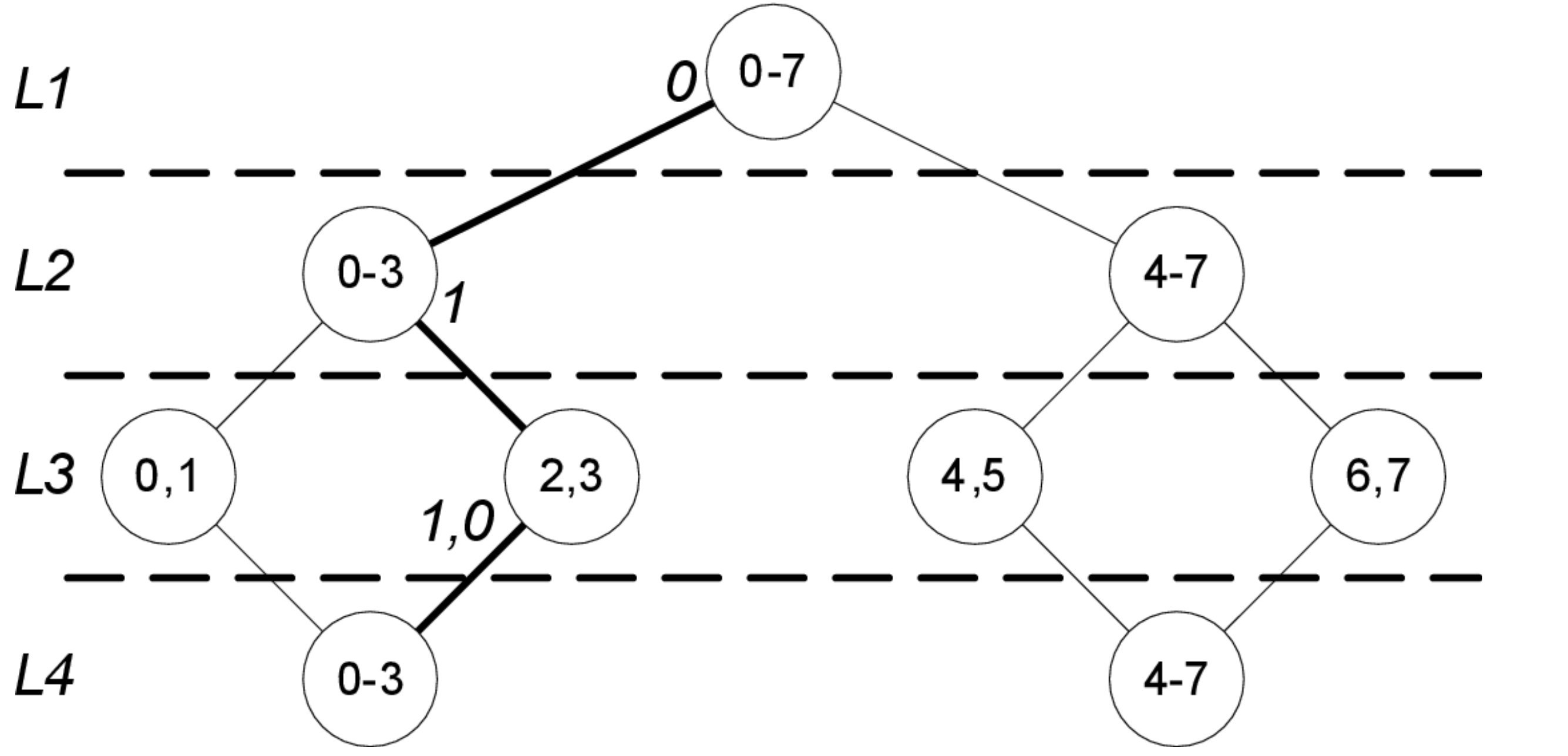
Most **digital hardware SNNs** implement a hybrid time- and event-driven system, without any form of event management, because hardware **solutions for event management** scale badly with the number of events. Similar to Agís et al. [2], we strictly follow the pattern of Algorithm 2, but our **Structured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

	Method	Hardware		Performance
		Memory	2-input comparators	Clock cycles (cc)
Agís et al. [2]	Unsorted list and on-the-fly search for the next event to happen	Memory array, with 256 read ports and **1 write port** **33% less memory usage**	Pipelined comparator tree, 255 comparators	10 cc with fewer than 2048 events, up to 69 cc with 16384 events
Caron et al. [3, 4]	**Structured Heap Queue** (SHQ), a variation of the pipelined heap queue [5]	**Binary memory tree with 2 read and 1 write ports per level** (**28 read ports** and 14 write ports with 16384 events)	**1 comparator per level** (**13 comparators** with 16384 events)	**7 clock cycles**

The Structured Heap Queue (SHQ)



The memory-optimized SHQ



Scaling, with N the number of events

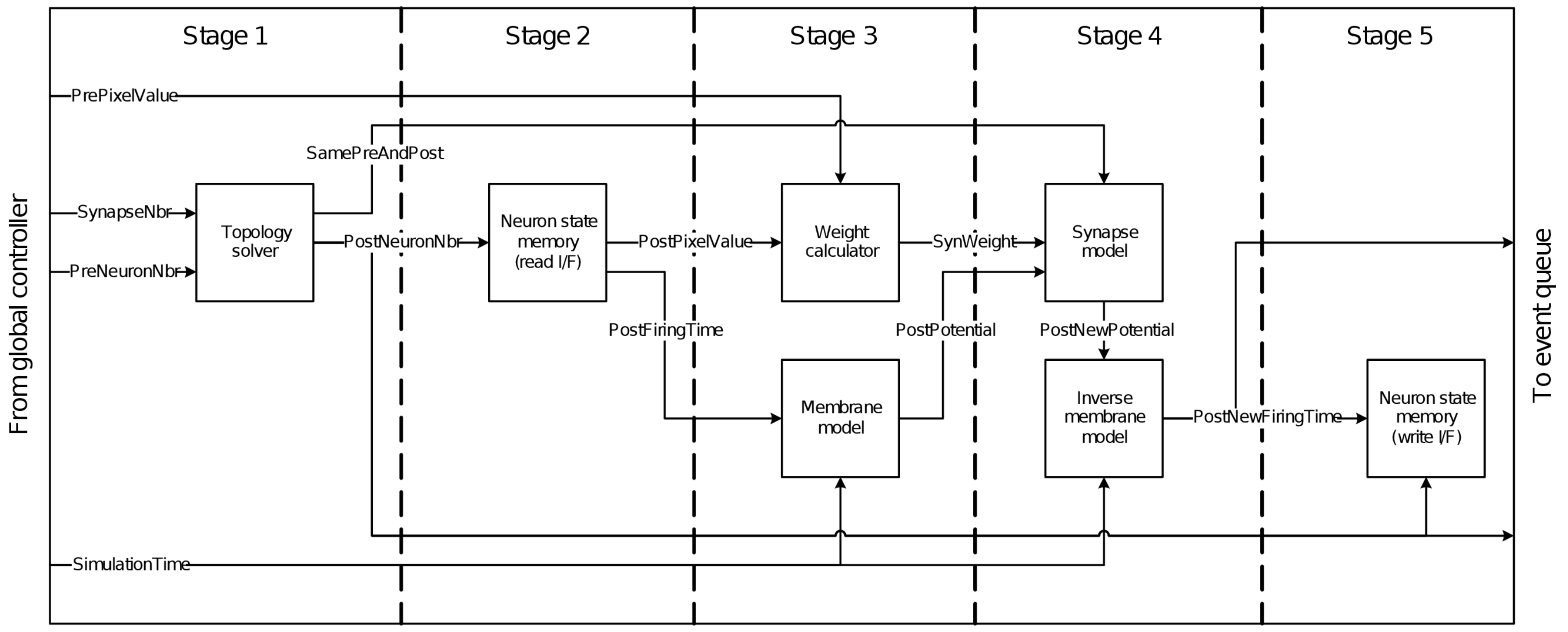
	SHQ	Agís et al. (2007)
Complexity in logic	$O(\log(N))$	$O(N)$*
Complexity in memory	$O(N)$	$O(N)$
Complexity in time for		
Delete top element	$O(1)$	$O(\log(N))$*
Delete any element	$O(1)$	$O(1)$
Insert	$O(1)$	$O(1)$
Read top element	$O(1)$	$O(\log(N))$*
Read any element	$O(\log(N))$	$O(1)$

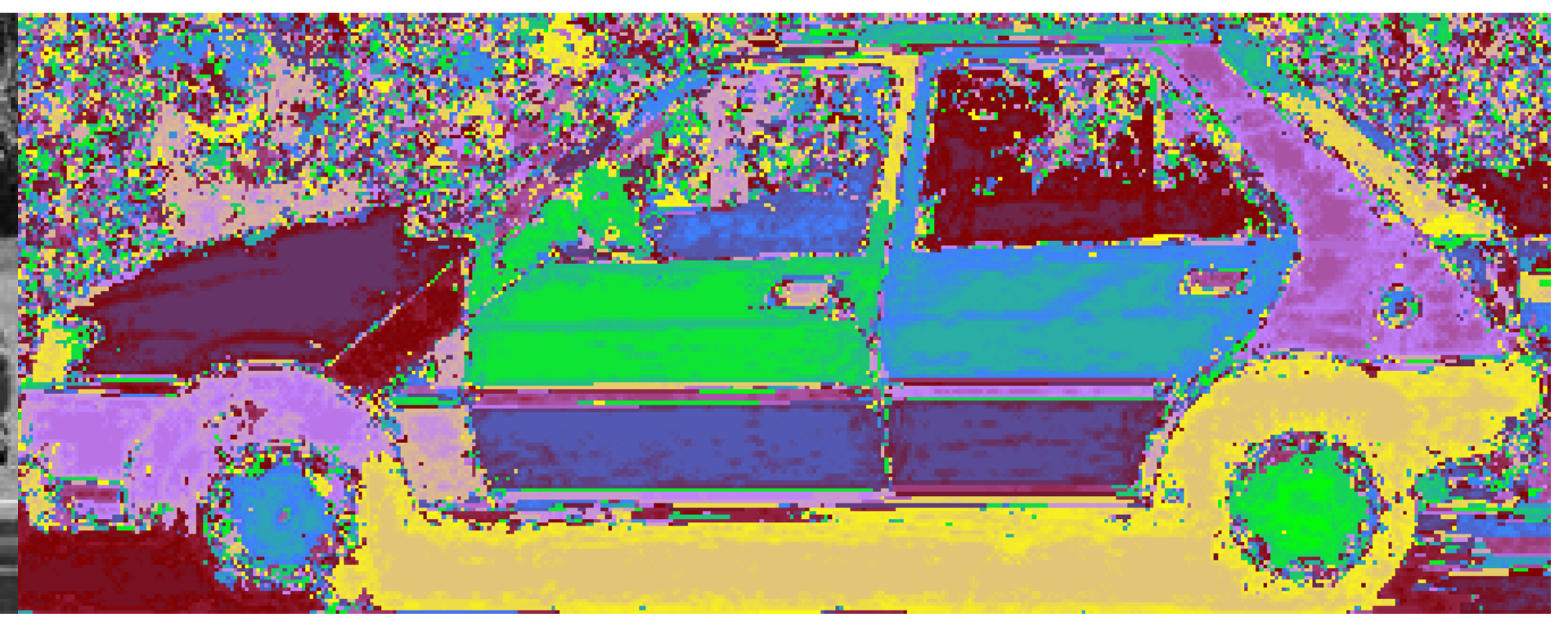
*In (Agís et al., 2007), logic can be traded for time, and conversely, time for logic.

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**





FUNCTION	FFs	LUTs	BRAMs
Controller	503	635	0
Processing element	19	73	50
Topology solver	16	26	0
Neuron memory	1	1	44
Weight calculator	0	8	0
Membrane model	1	14	3
Synapse model	0	10	0
Inverse membrane model	1	14	3
Event queue	2 846	3 965	80

Summary

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

[1] D'Haene, M. (2010). Ph.D. thesis.
[2] Agís, R., Ros, E., Díaz, J., Carrillo, R., & Ortigosa, E. M. (2007). International Journal of Electronics, 94 , 469–480.
[3] Rouat, J., Bergeron, J., Caron, L. C., de Ladurantaye, V., & Mailhot, F. (2012). WIPO Patent No. 2012167359. Geneva, Switzerland: World Intellectual Property Organization.
[4] Caron, L. C., D'Haene, M., Mailhot, F., Schrauwen, B., & Rouat, J. (2013). Neural Networks, ISSN 0893-6080 10.1016/j.neunet.2013.02.005.
[5] Ioannou, A., & Katevenis, M. G. H. (2007). IEEE/ACM Transactions on Networking, 15 , 450–461.
[6] Pichevar, R., Rouat, J., & Tai, L. (2006). Neurocomputing, 69 , 1837–1849.

Fonds de recherche
Nature et
technologies
Québec

CRSNG
NSERC