Configurable Conduction Delay Circuits for High Spiking Rates

B. Belhadj*, A. Joubert*, O. Temam† and R. Heliot*†
* CEA-LETI, Minatec Campus, 17 rue des Martyrs, 38054 Grenoble, France.
† INRIA Saclay Ile-de-France, rue Jean Rostand, 91893 Orsay Cedex, France.
rodolphe.heliot@cea.fr

***Abstract*— The conduction delay in neural systems has been proven to play an important role in processing neural information. In hardware spiking neural networks (SNN), emulating conduction delays consists of intercepting and buffering spikes for a certain amount of time during their transfer. The complexity of the conduction delay implementation increases with high spiking rates; it implies (1) storing a large number of spikes into memory cells and (2) conserving the required time resolution while processing the delays. As a result, the circuit size becomes very large and difficult to integrate into large scale SNN systems. In this paper, we highlight the trade-offs of an efficient digital delay circuit design supporting high neuron firing rates. The key issue resides in conserving spikes and spike timings while limiting storage requirements. We present a digital implementation of a configurable delay circuit supporting spiking rates of up to 1Meps (Mega events per second) and a delay range going from 1µs to 50ms with a time resolution less than 5% of the configured delay time. Synthesis results show that, using the CMOS 65nm technology, the required silicon area is 1600µm^2.**

I. INTRODUCTION

Biologically speaking, the conduction delay in neural systems refers to the propagation time required for an action potential (or spike) to travel from its initiation site near the soma to the dendrite terminal of the post-synaptic cell [1][2]. Conduction delays vary greatly in the mammalian nervous system, from 100 µs to 100 ms in very long unmyelinated central axons. Formal concepts have been drawn from biological evidences highlighting the importance of the conduction delay in neural computing, e.g. the concept of "polychronization", introduced in [3].

The rising importance of conduction delays in processing neural information requires their integration in hardware platforms that model networks of spiking neurons. Hardware architectures can provide low-power, massively integrated, and high performance computing platforms for SNNs [7]. The integration of the conduction delay functionality must conserve the properties of the hardware neural systems in terms of integration and computing capabilities (see [8] and [9] for examples of delay circuit implementations).

In this work, we focus on high-performance systems implementing neuron models with high spiking rates [4][5]. The delay function implementation in such systems requires large memory capacity, in order to buffer incoming spikes accumulated over the delay period, and it may come at a steep area cost. Therefore, delay circuits should especially focus on optimizing memory utilization. We consider a digital implementation rather than an analog one, not only for scalability, inherent noise rejection, robustness to variability, and reconfigurability reasons, but also because it is well suited to the binary nature of the neural information (spike '1' or not spike '0').

We consider that each delay circuit is associated with a single spike source (e.g. a neuron) to express its latency. In this case, spikes are represented by digital pulses, which facilitate their interception and storage in memory cells. Once configured, the delay time remains the same for all the incoming spikes. The delay time configuration ranges from tens of microseconds to tens of milliseconds. The time resolution has to be adjusted according to the configured delay period. In this work, we investigate and compare three circuit implementations corresponding to different design tradeoffs in time resolution and circuit size. Section II presents the three circuits, as well as their operations. Section III compares the circuits in terms of size and temporal precision. Section IV discusses which delay circuit should be used depending on the target application.

II. CONDUCTION DELAY CIRCUITS

The main function of the delay circuit is to keep track of all the incoming spikes as well as the time at which they arrive, in order to release them after a certain delay, and with the original inter-spike intervals (ISIs) between them. We first present a counter-based delay circuit, then a register-based circuit, and finally a mixed-mode circuit, each corresponding to different design tradeoffs.

A. Counter-based delay circuit

The key part of the circuit is a counter that counts the number of incoming spikes during the configured delay time. Each new spike increments the counter by one. Whenever the delay time expires, the counter value is transmitted to the

"decounter" which starts counting down spikes and releases them one by one (see Figure 1). Although this circuit keeps track of all incoming spikes, it does not conserve the time at which they arrive. The original inter-spike intervals are not stored and cannot be guessed later on. The spike releasing time has to be approximated. In this circuit, we define a module to compute the mean ISI as a function of the number of incoming spikes. For instance, assuming that the counter records 10 spikes during the delay time, the mean ISI is the result of the division of the delay period by 10. Figure 1 depicts the circuit schematics and illustrates its operation. Black segments in Figure 1 (b) represent the real position of input spikes and the expected position of output spikes (what it should be). Red segments represent the real position of output spikes released from the counter-based circuit. The difference between pairs of black and red segments represents the error induced by the delay computation, also called *jitter*. In our case, the jitter may be positive or negative.

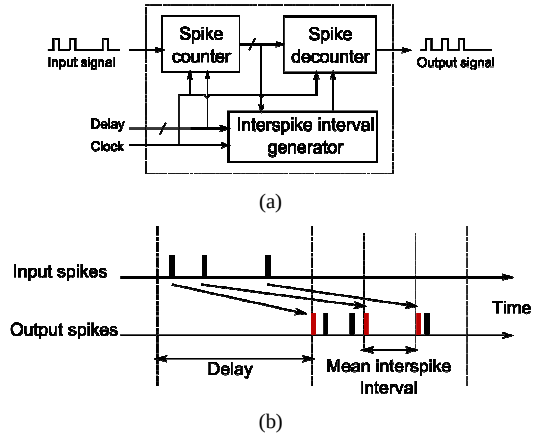

Figure 1. Counter-based delay circuit. (a) Circuit schematic. (b) Input and output spike timing illustration: black segments represent the real position of input spikes and the expected position of output spikes, while red segments represent the real position of output spikes. The distance between pairs of black and red segments is the jitter. The jitter may be positive or negative.

The counter size is a function of the maximum input firing rate (f_{max}) and the maximum delay (D_{max}). The required memory size used to store spikes is given by the following equation (expressed in bits);

$$n = 2.\log_2(D_{\max}.f_{\max} + 1) \qquad (1)$$

The jitter variability is high and may vary from 0 to D (the configured delay). The accuracy of the spike releasing time is not guaranteed and depends on the spike train characteristics. Let us define the density of an input spike train as the minimal percentage of spikes in a given time period: a density of 0% means that a delay period elapses without recording any spike, while a density of 50% means that the input spike train is, at least, half full during every delay period. The density is expressed as $f/f_{\max}$, where f is the firing rate over a given time period. The jitter J may be bounded as follows:

$$J < \left| D_{\max}(1 - density) \right| \qquad (2)$$

Thus, the jitter decreases when the density of the spike train increases, i.e., the circuit is more accurate if the spike traffic is high. This is simply due to the fact that an increasing number of spikes leads to a smaller mean ISI and, thus, reduced jitter variability. In order to illustrate this observation, Figure 2 shows software simulations of the circuit exercised with different densities of input spike trains. The delay is set to 20ms for all the simulations. Each point in the graph corresponds to the number of the released spikes that share the same jitter. The jitter variation is represented on the x axis. Simulations were run for spike train densities of 20%, 50% and 80% respectively. We can observe that, when gradually increasing the density of spike trains, the range of the jitter variation narrows. For an 80% density, the absolute value of the jitter never exceeds 20% of the delay (4ms), which may be an acceptable accuracy for certain applications.

As a conclusion, the circuit induces a jitter which decreases as the spike traffic increases, and which remains bounded to reasonably low values when the spike traffic is high. Such properties may be used for stable neural network activity and applications supporting coarse jitter variation.

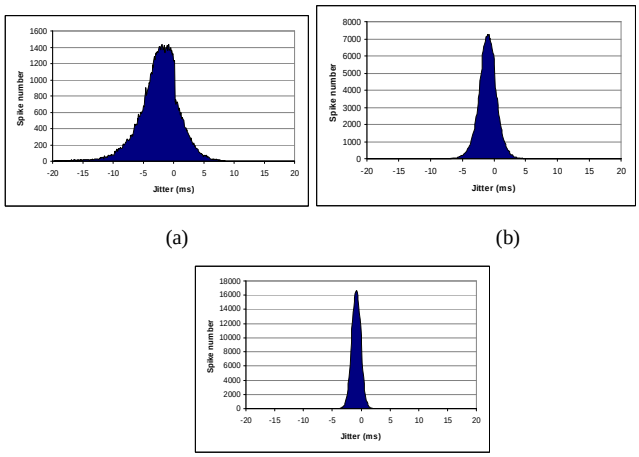

Figure 2. Jitter variation as a function of spike train density. Outgoing spikes sharing the same jitter are summed together. (a) Jitter variation for a density of 20%. (b) Jitter variation for a density of 50%. (c) Jitter variation for a density of 80%.

B. Register-based delay circuit

The circuit is based on a shift-register in which spikes are inserted and periodically shifted. The global operation is clocked by a shift period (T_{shift}) which defines, along with the register size (n), the delay duration. The delay is then configurable by fixing the shift period and it is equal to n. Whenever a spike arrives, a logical '1' is inserted in the first bit of the register. Otherwise, a logical '0' is inserted instead. The spike is then shifted within the register until it reaches the last bit, where it will be transformed again to a digital pulse; the register thus behaves like a spike FIFO (first in first out) queue.

The register size must support the highest configurable delay value and the maximum neuron firing rate. Equation (3) quantifies the register size of the delay circuit (expressed in bits),

$$n = D_{\max}.f_{\max} \qquad (3)$$

The delay is configured using T_{shift}, and only one spike can be accepted within one period. Therefore, in order to configure the maximum delay D_{max},, T_{shift} is set to $1/f_{max}$. The minimum configured delay is obtained when T_{shift} is set to $1/f_{clk}$, where f_{clk} is the clock frequency.

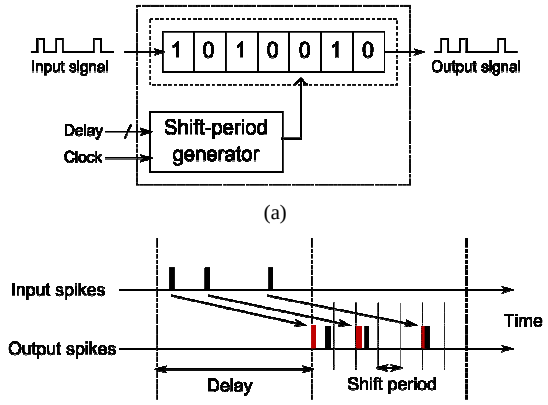

Figure 3. Register-based delay circuit. (a) Circuit scheme. (b) Input and output spike timing illustration.

Figure 3 presents the circuit schematic and illustrates its operation. In contrast to the counter-based delay circuit, the jitter is independent from the input spike train and it depends only on the shift period. The time space is slotted into several shift periods according to the delay value (e.g. 7 shift periods in Figure 3). The spike releasing time depends on the granularity of the shift period, and thus, it is related to the configured delay, which guarantees a jitter variation less or equal to the delay time step (T_{shift}):

$$J \le T_{shift} \qquad (4)$$

However, the register size increases with the maximum firing rate f_{max} and the maximum configurable delay D_{max}. The circuit becomes relatively large for high spiking rates. For instance, an application requiring a maximum firing rate of 100 KHz and a maximum delay of 50 ms needs a register of 5.10^3 bits, which induces heavy memory sizes, and thus, large circuit area.

We now investigate a mixed-mode circuit which combines the benefits of both circuits: the low hardware resource utilization of the counter-based circuit and the jitter control of the register-based circuit.

C. *Mixed counter-register delay circuit*

The circuit follows the same operation principles as the register-based circuit except that more than one spike can be accepted during the shift period (T_{shift}). A counter is therefore needed to count the number of incoming spikes within the shift period. Then, the number of spikes within the shift period, instead of a '1' or '0', is then inserted into an array of shift-registers. The array size is equal to the counter size. For instance, in Figure 5 (a), the counter size is set to 2 bits, which allows to count a maximum of 3 spikes per shift period (state '00' corresponds to "no spike"); the register array is then composed of two 3-bit deep shift-registers. Equation (5) generalizes the computation of the register array size:

$$n = \frac{D_{\max}}{T_{shift}} . \log_2 (T_{shift} \cdot f_{\max} + 1) \qquad (5)$$

Spikes are counted up and down following the same principle used in the counter-based circuit. Within a shift period, all ISIs are identical, corresponding to the mean ISI computed over the period. Similarly to equation (2), the jitter variation can be bounded by the following expression:

$$J < \left| T_{shift}(1 - density) \right| \qquad (6)$$

where *density* represents the minimal number of spikes coming during the T_{shift} period. The multi-bit shifting operation guarantees a bounded jitter which is relative to the configured delay. At the same time, it alleviates the required memory size by counting spikes instead of storing them all as binary information.

In summary, unlike for the counter-based circuit, the jitter variation can be bounded (less than the shifting period), and the memory requirement is reduced compared to the register-based circuit by counting spikes instead of storing them individually.

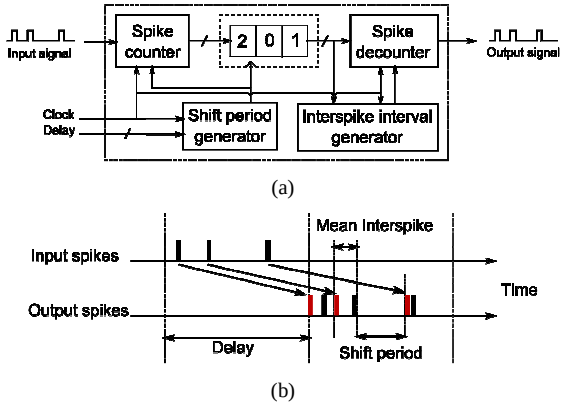

Figure 4. Mixed counter-register-based delay circuit. (a) Circuit schematic. (b) Input and output spike timing illustration.

III. CIRCUIT COMPARISON

In order to analyze the design tradeoffs, we extract the required memory sizes of the three proposed circuits as a function of the guaranteed jitter variation from equations (1)-(6). For that purpose, we set the *density* parameter to 40%, D_{max} to 50ms and f_{max} to 100KHz. Figure 5 summarizes the impact of the delay time accuracy on the circuit size. The register-based circuit can guarantee very small jitters but at the cost of large shift-register sizes. The counter-based circuit can guarantee the jitter variation to be less than 46% of the D_{max} value and uses 26 bits to compute spike delays. The mixed circuit presents the best trade-off between memory size and delay accuracy; it can guarantee the jitter variation to be less than 5% of the configured delay, and it requires only 138 bits for implementing the registers. Moreover, the number of bits decreases rapidly as the tolerated jitter increases.

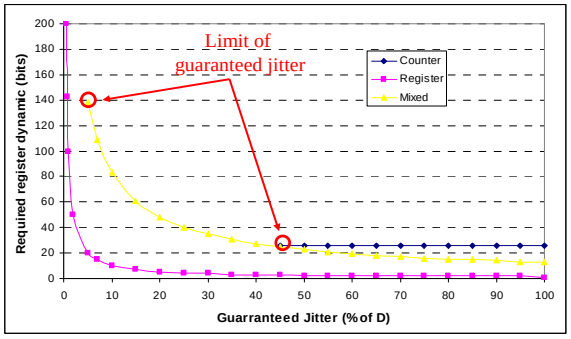

Figure 5. Comparison of the required memory size for the three circuits as a function of the guranteed jitter. The jitter is expressed as the pourcentage of the configured delay time.

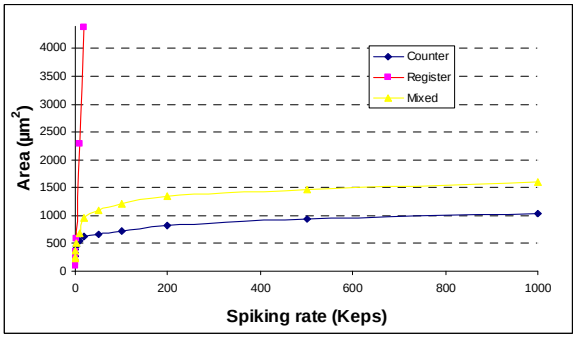

Figure 6. Synthesis results illustrating the area utilization of the three circuits as a function of f_{max} (spiking rate).

After gate-level simulation, we generated several versions of the three circuits supporting different firing rates. Each version was synthesized for 65nm technology. Figure 6 shows the results of post-synthesis area estimations. The area estimations for each circuit are provided as a function of the maximum firing rate (f_{max}). Measurements have been done using fixed values of D_{max} (50ms) and the jitter variation (5% of the configured delay time) for the register and mixed-mode circuit, except for the counter-based circuit since the jitter cannot be guaranteed. The required silicon area of the register-based circuit increases very rapidly because of the large memory requirement. It cannot support high firing rates at a reasonable area cost. The two other circuits exhibit reasonable area cost for high spiking rates. However, only the mixed-mode version can guarantee the jitter variation to be less than a given threshold without strong hypotheses on the spike trains characteristics. For 1Meps, the estimated area of the mixed counter-register circuit is 1600µm^2 against 1100µm^2 for the counter-based circuit.

IV. DISCUSSION

Through the analysis of the three proposed circuits, we can draw the following conclusions about the implementation of a delay circuit for high firing rates. The proper implementation depends on the target application requirements in terms of timing accuracy and circuit area. The first two circuits correspond to the more extreme cases, where either high temporal accuracy is needed whatever the area cost (register-based circuit), or low area cost is of prime concern whatever the temporal accuracy (counter-based circuit). The mixed counter-register circuit may be chosen for applications spanning the intermediate but more frequent situation where a certain balance must be achieved between temporal accuracy and area cost. Delays ranging from tens of microseconds to tens of milliseconds may be processed using the same circuit, which is useful for biologically realistic systems [6].

V. CONCLUSION

We have proposed three possible circuits for realizing the conduction delay functionality in hardware spiking neural network systems. The goal was to find a cost-efficient design that supports high firing rates while maintaining good temporal accuracy. We show that it can be achieved using a mixed counter-register implementation which provides a good area/accuracy tradeoff for a broad range of hardware spiking neural networks. The size of the delay circuit increases with the time granularity (temporal accuracy). As an example, we have synthesized a circuit with an area cost of only 1600 um² in CMOS 65nm and capable of processing up to 1 MEvents per second with a temporal accuracy of 5% of the configured delay time.

REFERENCES

[1] S. S-H Wang, J. R. Schulz, M. J. Burish, K. H. Harrison, P. R. Hof, L. C. Towns, M. W. Wagers and K. D. Wyatt, “Functional trade-offs in white matter axonal scaling,” J. Neurosci., vol. 28, pp. 4047-4056, 2008.

[2] H. A. Swadlow, “Impulse conduction in the mammalian brain: Physiological properties of individual axons monitored for several months,” Science, vol. 218, pp. 911-913, 1982 .

[3] E. M. Izhikevich, “Polychronization: Computation with spikes,” Neural Computation, vol. 18, pp. 245-282, 2006.

[4] J. Schemmel, D. Brüderle, K. Meier, and B. Ostendorf, “Modeling synaptic plasticity within networks of highly accelerated I&F neurons,” ISCAS conference, pp. 3367-3370, 2007.

[5] R. Héliot and O. Temam, “Robust and low-power accelerators based on spiking neurons for signal processing applications,” International Workshop on Design for Reliability (DFR), January 2011

[6] S. Saighi, T. Levi, B. Belhadj, O. Malot and J. Tomas, “Hardware system for biologically realistic, plastic and real-time spiking neural network simulations,” IJCNN conference, pp.1-7 , 2010.

[7] J. Schemmel, J. Fieres, and K. Meier, “Wafer-scale integration of analog neural networks,” IJCNN conference, pp. 431-438, 2008.

[8] M.J Pearson, A. G. Pipe, B. Mitchinson, K. Gurney, C. Melhuish, I. Gilhespy and M. Nibouche, “Implementing spiking neural networks for real-time signal-processing and control applications: a model validated FPGA approach,” IEEE transactions in neural networks, pp 1472-1478, 2007.

[9] J. L. Yang, C. W. Chao and S. Lin, “Tunable delay element for low power VLSI circuit design,” IEEE Ciruits and systems, pp. 1-4, 2006