Introduction to Stochastic Computing using a Remote Lab with Reconfigurable Logic

http://dx.doi.org/10.3991/ijoe.v12i04.5091

Jorge Lobo
University of Coimbra, Coimbra, Portugal

***Abstract*—This short paper introduces the basic concepts of Stochastic Computing (SC), and presents additions to a remote lab with reconfigurable logic to allow testing SC circuits. Recently, SC has been revisited and evaluated as a possible way of performing approximate probabilistic computations for artificial perception systems. New modules allow the generation of pseudo-random numbers, given a seed key and using linear feedback shift registers, but also having true random number generation using ring oscillators and embedded PLLs. Stochastic computing allows a tradeoff between resource usage and precision, allowing very simple circuits to perform computations, at the expense of a longer integration time to have reasonable results. We provide the basic stochastic computing modules, so that any user can use them to build a stochastic computing circuit and go beyond software simulations, providing a remote hardware device to test real circuits at high clock speeds.**

***Index Terms*—remote lab, reconfigurable logic, stochastic computing.**

I. INTRODUCTION

Stochastic Computing (SC), has been proposed [1, 2] as an alternative number representation scheme which provides higher tolerance to errors and more compact operators than conventional representation schemes, e.g. fixed-point and floating point binary representations.

Recently is has been revisited and evaluated as a possible way of performing approximate computations in probabilistic computations for artificial perception systems. Essentially, stochastic, or telegraphic, signals are defined as generated by a memoryless continuous-time stochastic process producing two distinct values. A stochastic stream, according to [3], is defined as a sequence of stochastic signals over time, where its value is defined as the number of ones over the total number of bits.

Recently, it has gained the attention of the scientific community for its low implementation cost, fast computations and robustness of approximated streams [4]. This demo showcases the basic principles of Stochastic Computing and allows the design and test of circuits using the remote FPGA. This enables to go beyond simple simulations, and having real circuits working at high clock speeds and embedded real random number generators.

The main limitations of SC are that a linear increase in the precision of stochastic computations requires an exponential increase in the length of the bit-stream, sensitivity to temporal correlations, and limited dynamic range of the representation. Working with very long bit steams can be time consuming when performing simulations, and dedicated hardware with the remote lab is a clear advantage.

The remote FGPA laboratory [5] is online at http://lsd.deec.uc.pt and will be used for the interactive demonstration. To have access to the hardware a simple registration is required to have a user account. Figure 1 shows the remote interaction with the DE2 FPGA board.

The remote lab connects an ALTERA DE2 FPGA board over the Internet, allowing it to be used as if the user is in loco, programming, testing and looking at it. This board has many switches, keys, LEDs, LCDs and 7-segment displays, and we need to look at it and use the switches and keys, remotely. This was done with a webpage interface, with PHP dynamic functionalities that enabled the remote use of the DE2 boards, and a webcam for visual feedback. In [6] the system is described in more detail, and presented in the context of a digital design course.

II. STOCHASTIC COMPUTING

Figure 1 shows a stochastic computing circuit realizing the arithmetic function z = X1.X2.X4 + X3.(1 − X4).

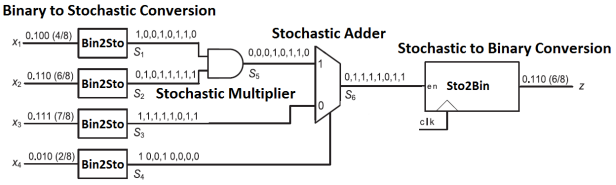

Figure 1. Example of a stochastic computing circuit [4]

The following subsections present the basic units required to implement stochastic computing circuits. More details on these and other stochastic units can be found in [4].

A. Pseudo-Random Number Generation

To generate the stochastic signals we need a source of random numbers. The pseudo-random number generator is a circuit that generates sequences of bits with properties approximate of sequences of random bits, and it is at the heart of the generation of the stochastic variables [7]. In this work a Linear Feedback Shift Register, LFSR, was used to generate the pseudo-random bit-streams because of its circuit simplicity, reproducibility and speed. An example of a LFSR is presented in figure 2.

It is made of an N-bit shift-register, and a combinatorial circuit that produces the input signal out of specific bits, or taps, in the shift-register. A table with the appropriate taps to use for LFSR of different sizes is presented in [8].

B. Binary to Stochastic Conversion

Assuming we start with a binary number, a circuit is needed to convert a binary constant into a stochastic bit-stream. Figure 3 shows the architecture of the block to do this conversion. It is made of a PRNG and a ”less than’” comparator. In terms of operation, the PRNG, generates a

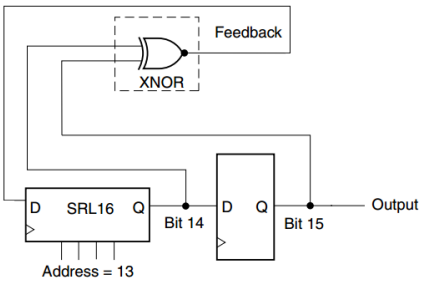

Figure 2. Block diagram of the pseudo-random generator [8].

value from a uniform distribution, which covers all 2^m possible values for an input of the comparator.

The other input of the comparator is connected to the input of the value that is to be converted to stochastic computing. If the pseudo-random value is less than the value being compared, the comparator produces a 1, 0 otherwise. Thus, the portion of the time ON is related to the value being converted.

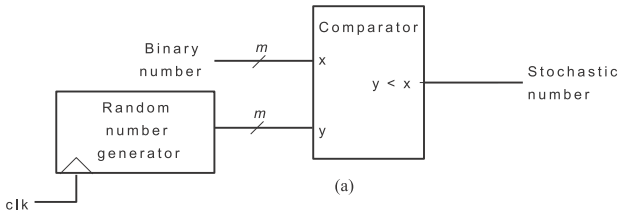

Figure 3. Binary to stochastic conversion [8].

C. Stochastic to Binary Conversion

Retrieving information from the stochastic system, into a known and compact binary format, is achieved by counting the number of ones present in the given stochastic sequence.

Moreover, to limit this process in time, there are two counters, one to count the number of ones and the other to count the number of ones and zeros. Thus, the output of the stochastic to binary converter p, which encodes the probability of the signal being 1, is defined by the ratio between the two in equation (1).

$$p = \frac{number\ of\ ones\ in\ the\ sequence}{total\ number\ of\ bits\ in\ the\ sequence} \quad (1)$$

Figure 4 presents the architecture of this unit with the counter for the number of 1s in the stochastic stream. Another counter is required to count the duration of the stochastic stream in parallel (temporal window).

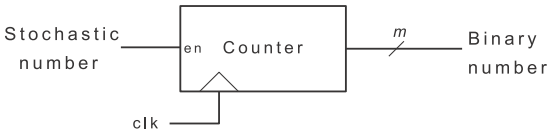

Figure 4. Stochastic to binary conversion [4].

D. Multiplication

Assuming two independent stochastic signals i and j, associated with the real numbers c_i, c_j (both $\in[0,1]$) by using a bitwise logic AND gate we can generate a third stochastic signal k having the following property:

$$c_k = c_i . c_j \quad (2)$$

This is similar to the multiplication between two probabilities. Given two probabilities $P(x)$ and $P(y)$, assuming that x and y are independent, $P(x=a\ and\ y=b)$ translates into $P(x=a) \times P(x=b)$. Figure 5 shows the block diagram of the circuit to compute multiplications for two stochastic signals using a logic AND gate.

Figure 5. Stochastic multiplier.

E. Addition

Again, assuming n independent stochastic signals, an addition is achieved by connecting the stochastic signals to a n-input multiplexer and a n-modular counter which selects an input of the multiplexer at each time a clock toggles its status.

The stochastic signal obtained at the output of the multiplexer has the following property:

$$s = \frac{1}{n}\sum_{i=1,..,n} c_i \quad (3)$$

In fact, the computation offered by the multiplexer is the average of n stochastic signals, using the same weights, since $1/n$ is constant for all the values of n. On the other hand, this scaling of the signal allows keeping the representation at the output within [0,1]. Figure 6 shows the architecture for the stochastic adder using a 2-input multiplexer and a clock signal to toggle between these two inputs. For multiplexers with $n>2$ inputs, a cyclic counter is connected to the input selection port. When cascading stochastic streams should be used with equal probability to select the mux.

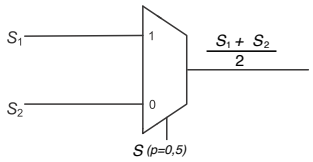

Figure 6. Stochastic adder.

III. REMOTE LAB FOR STOCHASTIC COMPUTING

The remote lab connects an ALTERA DE2 FPGA board over the Internet, allowing it to be used as if the user is in loco, programming, testing and looking at it [5]. Figure 7 illustrates the basic concept of this remote lab. A web server that is accessible through the Internet is connected via USB to the FPGA JTAG port on the board, and a webcam for visual feedback connects to the server, conveying real-time results on the 16x2 LCD and the 7-segments displays [5,6]. The board switches and keys are virtualized with a webpage interface, with PHP dynamic functionalities that enabled the control of the remote hardware via the JTAG port.

The user steps to use this online platform for SC are:

- Access the online remote lab (http://lsd.deec.uc.pt)
- Register
- Download the provided stochastic modules
- Build stochastic circuit
- Test the stochastic circuit following the instructions given in [5,6]

Figure 7. Concept of the reconfigurable logic remote laboratory board: virtualised board keys and switches, webcam to observe LEDs, 7 segment displays and LCD display [5,6].

A. Building and Testing a Circuit

To build a stochastic circuit, the users need to first download the stochastic modules. The provided modules for SC are:

- Binary to Stochastic Converter
- Random Number Generators (RNG)
  - Linear Feedback Shift Register
  - Ring Oscillators based RNG
- Comparator
- Stochastic to Binary Converter
- Stochastic Multiplier and Stochastic Adder

They correspond to the concept figures presented above, and can be seen in the following example.

The stochastic bit-stream generator is composed of a RNG and a Comparator. For the RNG two modules are provided: A Linear Feedback Shift Register (LFSR), and a hardware Random Number Generator. The LFSR has the advantage of its circuit simplicity, reproducibility and speed, but requires a seed. As alternative, we provide a high speed on-chip solution for generating random numbers using a RNG based on jitter sampling on ring oscillators requiring no seed [9]. The user can use it to replace the LFSRs or it can also be used to seed the LFSRs.

Figure 8 presents an example of a stochastic computing circuit using the provided modules, which implements the arithmetic function

$$z = X_3 \cdot \left(X_1 \cdot X_2 + X_4 + 1 \right) . \quad (4)$$

IV. CONCLUSIONS

We have presented a brief introduction to Stochastic Computing, and provided modules on a remote lab to experiment with this type of circuits. The remote laboratory provides a reconfigurable logic device (an Altera DE2 board hosting a Cyclone IV FPGA) to test real circuits. Besides the hardware device, the remote lab also provides basic modules for Stochastic Computing and depending on user's interests and needs, complex stochastic computing circuits can be tested.

ACKNOWLEDGMENT

The author would like to acknowledge the valuable contribution of José Domingos Alves in preparing the modules for the remote lab and presenting the demo at Exp.at'2015.

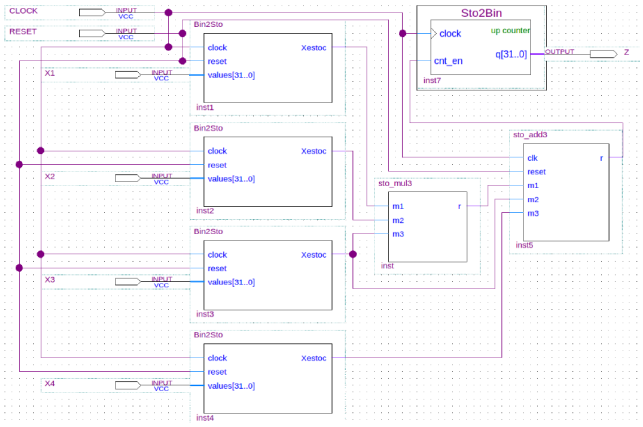

Figure 8. Stochastic computing circuit built with the provided modules implementing function in equation (4).

REFERENCES

[1] B. R. Gaines, "Techniques of identification with the stochastic computer," in *Proc. International Federation of Automatic Control Symposium on Identification*, Progue, 1967.

[2] J. Von Neumann, "Probabilistic logics and synthesis of reliable organisms from unreliable components," in *Automata Studies (C. Shannon and J. McCarthy, eds.*) vol. 34, pp. 43–98, Princeton University Press, 1956.

[3] Gaines, B. R. 1969. "Stochastic computing systems". *Adv. Information Systems Science* vol 2, 37–172, 1969. http://dx.doi.org/10.1007/978-1-4899-5841-9_2

[4] A. Alaghi and J. P. Hayes, "Survey of stochastic computing, " *ACM Trans. Embedded Computing Systems*, vol. 12, pp. 92:1–92:19, May 2013.

[5] J. Lobo, "Interactive demonstration of a remote reconfigurable logic laboratory for basic digital design," *International Journal of Online Engineering (iJOE)*, vol. 8, pp. 19–20, 2012. http://dx.doi.org/10.3991/ijoe.v8is1.1886

[6] J. Lobo, "A Remote Reconfigurable Logic Laboratory for Basic Digital Design", in *Proceedings of the 1st Experiment@ International Conference - Remote & Virtual Labs - exp.at'11*, November 2011.

[7] E. Barker and J. Kelsey, "Recommendation for random number generation using deterministic random bit generators," *National Institute of Standards and Technology*, 2012. http://dx.doi.org/10.6028/NIST.SP.800-90a

[8] P. Alfke, "Efficient shift registers, LFSR counters, and long pseudo-random sequence generators," *Xilinx Application Note XAPP 052*, July 1996.

[9] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Transactions on Computers*, 2007. http://dx.doi.org/10.1109/TC.2007.250627

[10] B. R. Gaines, "Stochastic computing," in Proceedings of the April Spring Joint Computer Conference, AFIPS '67, New York, NY, USA: ACM, pp. 149-156, 1967. http://dx.doi.org/10.1145/1465482.1465505

AUTHOR

Jorge Lobo has a Ph.D. degree from the University of Coimbra. He currently works as Assistant Professor with tenure at the Electrical and Computer Engineering Department, University of Coimbra, Coimbra, Portugal. He is responsible for courses on Digital Design, Microprocessors and Computer Architecture. His current research is carried out at the Institute of Systems and Robotics, University of Coimbra, working in the field of computer vision, sensor fusion, and mobile robotics. Current research interests focus on inertial sensor data integration in computer vision systems, Bayesian models for multimodal

perception of 3D structure and motion, and real-time performance using GPUs and reconfigurable hardware. He has participated in several national and European projects, most recently in BACS - Bayesian Approach to Cognitive Systems, HANDLE - Developmental pathway towards autonomy and dexterity in robot in-hand manipulation, and currently in BAMBI - Bottom-up Approaches to Machines dedicated to Bayesian Inference (FET - FP7-ICT-2013-C, 618024). E-mail: jlobo@ isr.uc.pt

The work leading to these results was partially supported by the European Commission collaborative FET project BAMBI - Bottom-up Approaches to Machines dedicated to Bayesian Inference - FP7-ICT-2013-C, project number 618024 (www.bambi-fet.eu), and by the Institute of Systems and Robotics at Coimbra University.

Submitted, 09 March 2015. Published as resubmitted by the authors on 09 April 2015.