FPGA Implementation and Evaluation of Discrete-time Chaotic Generators Circuits

Pascal Giard[†‡], Georges Kaddoum[†], François Gagnon[†] and Claude Thibeault[†]

[†]LaCIME, Department of Electrical Engineering, École de technologie supérieure,
1100 Notre-Dame O., Montréal, Canada, H2R 2J7
Emails: {pascal.giard,georges.kaddoum}@lacime.etsmtl.ca, {francois.gagnon,claude.thibeault}@etsmtl.ca

[‡]IML, Department of Electrical and Computer Engineering, McGill University,
3480 University St., Montreal, Canada, H3A 0E9
Email: pascal.giard@mail.mcgill.ca

***Abstract*—In this paper, implementation of discrete-time chaotic generators widely used in digital communications is studied and evaluated. The study focuses on power consumption, resource usage, and maximum execution frequency of implementations for two common Field Programmable Gate Arrays (FPGAs). While the Bernoulli map ranks first in all three aspects, results show significant ranking differences among the other chaotic generators. Results were obtained by first implementing the chaotic generators in a high level register to transistor level description language and then using tools from FPGA manufacturers to obtain the resource usage as well as estimate the other desired characteristics.**

I. Introduction

The chaotic synchronization demonstrated in [1] increases the interest to implement chaotic signals in communication systems. The aim of this interest returns mainly to the advantages provided by chaotic signals, such as robustness in multipath environments and resistance to jamming [2]. In addition, chaotic signals are non-periodic, broadband, and difficult to predict and to reconstruct. These are properties which coincide with the requirements for signals used in communication systems, particularly in spread-spectrum and secure communication applications [3], [4].

In the literature a large number of papers exists on chaotic spreading sequences design [5], optimization [6] and modeling [7], [8]. Another approach to select a good chaotic sequence for spread spectrum based on the bit energy distribution is studied in [9]. From these previous works, a first selection of chaotic sequence generators based on the performance properties can be done. However, to go further into the design of a chaotic communication system, a chaotic generator must be carefully chosen from that first selection to take into account power consumption and simplicity of integration. These criteria together with the performance properties must not be neglected. This later problem is discussed in the paper in order to give a tool for engineers to select the optimal chaotic generator based on design constraints.

Many papers study the design and implementation of a chaotic generator on a Field Programmable Gate Array (FPGA) [10]–[13]. To our best knowledge, there is no paper which helps us select the optimal chaotic generator based on engineering priorities for a given FPGA.

In this paper, we first give an overview of chaotic generators widely used in communication systems. We then briefly present implementation details of the generators before describing how we calculated resource usage and estimated maximum execution speed as well as dynamic power consumption. Obtained characteristics for various generators are then compared on two common FPGAs. Tables summarizing the results and comparing the properties of different generators are offered to facilitate the choice of an adequate chaotic generator based on engineering priorities. We terminate this paper with some concluding remarks.

II. Chaotic generators

A. Overview

The choice of discrete-time chaotic generators returns to their wide use in chaos-based communication systems [2]. The chosen discrete-time chaotic generators are:

1) Bernoulli map (B):

$$\mathbf{x}[n+1] = \begin{cases} B\mathbf{x}[n] - A & \text{if } \mathbf{x}[n] \geq 0 \\ B\mathbf{x}[n] + A & \text{if } \mathbf{x}[n] < 0 \end{cases} \quad (1)$$

where A and B are constants. In this paper $A = 0.5$ and $B = 1.75$.

2) Chebychev map (Ch):

$$\mathbf{x}[n+1] = 1 - 2\mathbf{x}^2[n] \quad (2)$$

3) Tent map (T):

$$\mathbf{x}[n+1] = A - B|\mathbf{x}[n]| \quad (3)$$

where A and B are constants and $|\cdot|$ is the absolute value operator. A and B are chosen to be 0.5 and 1.99 respectively.

[†] This work has been supported in part by Ultra Electronics TCS and the Natural Science and Engineering Council of Canada as part of the 'Wireless Emergency and Tactical Communication Chair' at École de technologie supérieure.

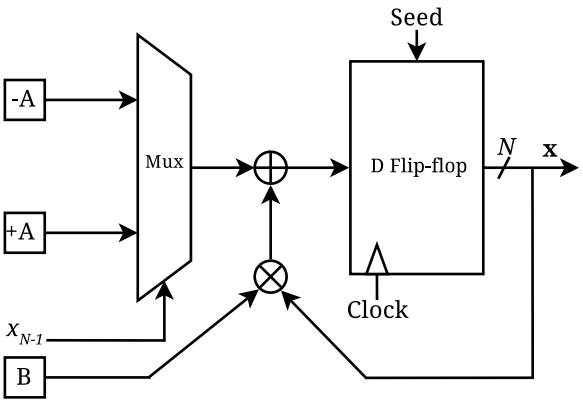

Fig. 1. Bernoulli map block diagram

4) Cubic map (Cu):

$$\mathbf{x}[n+1] = 4\mathbf{x}^3[n] - 3\mathbf{x}[n] \tag{4}$$

B. Implementation

Chaotic generators were implemented using a high level register to transistor level description language: VHDL. Generators are synchronous, have an asynchronous reset and use a fixed point number representation. Generators have been implemented for three word lengths: 16, 24 and 32 bits. As a reference point, a Gold number generator with a register size $R = 6$ has also been implemented.

Gold number generators are a particular class of Pseudo Noise (PN) generators widely used in spread-spectrum applications and communication systems in general. They are typically implemented with Fibonacci linear feedback shift registers, by modulo-2 addition of sequences of the same maximal length, and thus have the property of being periodic number generators. As chaos generators do not have that property, they are expected to replace Gold number generators, especially for secure communications.

Block diagrams of the discrete-time chaotic generators are shown in Figures 1 to 4 where N is the word length and x_{N-1} is the most significant bit of the signed vector $\mathbf{x}$. On these figures, left arrows "$\leftarrow$" are binary left shift operators, and the symbols "$\oplus$" and "$\otimes$" designate signed arithmetic additions and multiplications respectively. Seeds were chosen in order to obtain functions with a chaotic behavior. Functionality of our implementations was verified against software reference models written in C++.

Given that the FPGA circuit has a finite resolution, the chaotic codes generated will eventually become periodic. As a result, the period of the chaotic samples as well as their variation depending on the initial conditions are of interest. A periodicity test based on initial conditions was studied in [14] where a set of initial conditions are chosen to give the longest sequences. In our paper we do not study the periodicity of the chaotic generators, only implementation aspects are studied and analyzed.

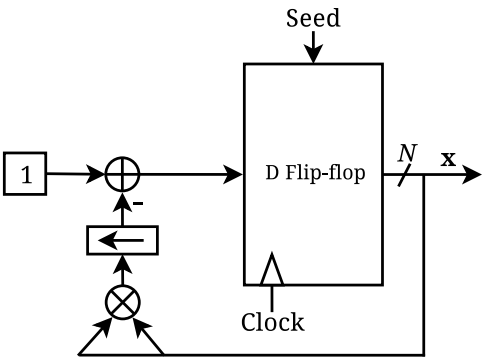

Fig. 2. Chebychev map block diagram

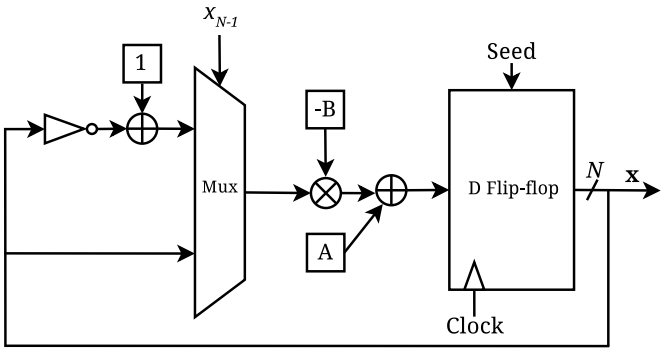

Fig. 3. Tent map block diagram

C. Obtaining characteristics

Generators were implemented and evaluated for two common FPGAs using tools provided by the manufacturers as shown in Table I. These tools include resource usage calculation in addition to maximum clock frequency and dynamic power consumption estimation. Targeted FPGAs include the Xilinx Virtex 6 (XC6VLX240T-3 FFG1156) and the Altera Cyclone III (EP3C25F324C8). The former has been chosen to represent high speed, high capacity FPGAs while the latter priories low power at the cost of reduced maximum clock frequency. The Xilinx synthesizer was set to optimize for power.

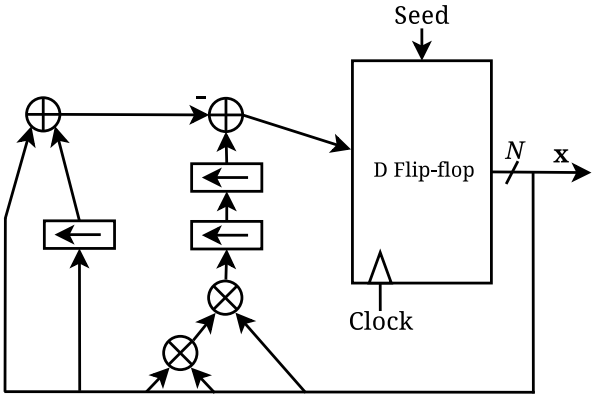

Fig. 4. Cubic map block diagram

TABLE I
TOOLS USED TO OBTAIN CHARACTERISTICS

	Resources/Frequency	Power	Version
Xilinx	PlanAhead	XPower Analyzer	12.4
Altera	Quartus II	PowerPlay	10.1

TABLE II
ESTIMATED DYNAMIC POWER CONSUMPTION (MW) FOR CIRCUITS WITH DIFFERENT WORD LENGTHS (BITS), USING DSP BLOCKS OR NOT

Xilinx Virtex 6	WL	B	Ch	T	Cu
DSP blocks	16	2.31	3.2	5.11	7.76
	24	2.35	3.7	4.02	18.22
	32	2.49	4.22	5.5	24.02
No DSP blocks	16	4.57	5.63	9.42	16.81
	24	4.61	5.7	13.86	34.7
	32	7.53	15.39	20.46	62.32
Altera Cyclone III	**WL**	**B**	**Ch**	**T**	**Cu**
DSP blocks	16	0.12	0.38	0.28	0.84
	24	0.16	0.89	0.74	2.04
	32	0.2	1.4	1.12	5.65
No DSP blocks	16	0.12	0.5	0.28	1.25
	24	0.16	1.05	1.12	3.1
	32	0.2	1.79	2.06	5.65

The Altera synthesizer was set to use a balanced optimization technique and to put extra effort on power optimization. All characteristics were obtained from the implemented and routed designs with and without the use of digital signal processing (DSP) blocks. In the case of the Altera Cyclone III, DSP blocks actually are embedded multiplier 9bit elements. From the designs, only multipliers were offloaded to DSP blocks. Furthermore, power estimation was obtained for the targeted frequency of 100 MHz for the Xilinx FPGA and 48 MHz for the Altera FPGA.

III. RESULTS AND DISCUSSION

a) Power consumption: Table II shows the dynamic power consumption for the chaotic generators implemented on both the Xilinx Virtex 6 and Altera Cyclone III FPGAs, with and without the use of DSP blocks. As expected, on both FPGAs, the dynamic power consumption is reduced when DSP blocks are actually used.

Interestingly, in terms of dynamic power consumption, the Chebychev and Tent chaotic generators do not always rank in the same order on the Altera and Xilinx FPGAs. While on Xilinx they always rank as second and third, on Altera they mostly rank as third and second respectively.

b) Resource usage: In terms of FPGA resource usage, the amount of resources required by the implemented generators show different ranking on both FPGAs for different word lengths. Whether DSP blocks can be used or not also influences the ranking with the exception of the Cubic generator that is always last. As shown in Table III, when DSP blocks are allowed on the Xilinx FPGA, the Bernoulli and Chebychev generators rank first by requiring the same resources at all word lengths while Tent ranks third. However, when DSP blocks are not used, the Bernoulli generator clearly requires fewer resources followed by Chebychev and Tent (at a word length of 24 bits, the Tent generator requires less resources than Chebychev).

TABLE III
RESOURCE USAGE, AMOUNT OF LOOKUP TABLES (XILINX) OR LOGIC ELEMENTS (ALTERA) AND DSP BLOCKS FOR CIRCUITS WITH DIFFERENT WORD LENGTHS (BITS), USING DSP BLOCKS OR NOT (DSP BLOCKS ARE SHOWN IN BRACKETS)

Xilinx Virtex 6	WL	B	Ch	T	Cu
DSP blocks	16	9 (1)	9 (1)	41 (1)	32 (3)
	24	13 (2)	13 (2)	61 (2)	75 (8)
	32	17 (4)	17 (4)	81 (4)	115 (10)
No DSP blocks	16	48	224	236	799
	24	70	444	404	1710
	32	92	730	823	2945
Altera Cyclone III	**WL**	**B**	**Ch**	**T**	**Cu**
DSP blocks	16	37 (0)	17 (2)	177 (0)	69 (6)
	24	54 (0)	57 (7)	144 (7)	257 (19)
	32	69 (0)	89 (8)	186 (8)	387 (24)
No DSP blocks	16	37	341	177	1057
	24	54	819	807	2477
	32	69	1402	1403	4321

Also shown in Table III, on the Altera FPGA, the Bernoulli generator still ranks first. Otherwise, results are similar to those for the Xilinx FPGA with the exception of the Tent generator using fewer resources than the Chebychev generator when DSP blocks are disabled. Interestingly, the Altera Quartus II synthesizer was able to infer a sum of shifted vectors from the constant multiplication in the Bernoulli generator, eliminating an expensive multiplier. It was also the case for the 16 bit Tent generator.

c) Clock frequency: Finally, as shown in Table IV, for the Altera Cyclone III, the Bernoulli generator is also the fastest with a maximum clock frequency ranging from 8% to 155% higher than its closest competitor, the Chebychev generator. On the Xilinx Virtex 6, results are similar with the Bernoulli map implementation having a higher maximum clock frequency than the Chebychev generator ranging from 0% to 124%.

Notice that the Cubic chaotic generator design was not always able to meet the minimum targeted clock frequency of 100 MHz for the Virtex 6 and 48 MHz for the Cyclone III. Although not shown here, the targeted frequency could be met by modifying the design into a pipelined architecture with an initial latency of 3 clock cycles at the expense of greater resource usage.

All in all, on both FPGAs, the Bernoulli chaotic generator is clearly the most energy efficient among the compared generators with its low resource usage, high maximum execution frequency and low dynamic power consumption. However, by visual inspection of equation (3), the generator based on the Tent map was expected to come second on all aspects and it is not the case.

d) Gold number generator: As a reference, on the Xilinx Virtex 6, a Gold number generator with $R = 6$ requires 6 lookup tables, can be executed at the theoretical maximum clock frequency of 800 MHz and consumes an estimated 2.22 mW of dynamic power. On the Altera Cyclone III, the same Gold number generator occupies 12 logic elements, can be

TABLE IV
MAXIMUM EXECUTION FREQUENCY (MHZ) FOR CIRCUITS WITH DIFFERENT WORD LENGTHS (BITS), USING DSP BLOCKS OR NOT

Xilinx Virtex 6	**WL**	**B**	**Ch**	**T**	**Cu**
DSP blocks	16	256.2	244.5	180.9	115.1
	24	194.5	186.9	144.7	81.6
	32	128.0	128.0	103.7	72.5
No DSP blocks	16	333.2	169.0	140.0	103.3
	24	302.6	144.0	116.2	93.9
	32	265.9	118.7	111.8	76.0
Altera Cyclone III	**WL**	**B**	**Ch**	**T**	**Cu**
DSP blocks	16	185.9	171.5	92.5	65.0
	24	179.5	94.6	65.8	41.6
	32	154.3	85.6	59.2	38.0
No DSP blocks	16	185.9	96.1	92.5	41.5
	24	179.5	70.3	57.1	31.2
	32	154.3	64.1	50.9	28.9

executed with a maximal clock frequency of 621.5 MHz and consumes an estimated 40μW.

IV. CONCLUSION

This paper first studies the implementation of four discrete-time chaotic generators, widely used in digital communications, for two common FPGAs. Then the resource usage and estimated maximum execution speed as well as dynamic power consumption are evaluated and compared. Results show that the Bernoulli map is estimated to be the most energy efficient chaotic generator as it requires fewer resources, can be clocked at higher frequencies and consumes less dynamic power than the other chaotic generators. Depending on the priorities of the designer, the order in which the other chaotic generators rank varies.

REFERENCES

[1] L. M. Pecora and T. L. Carroll, "Synchronization in Chaotic Systems," *Phys. Rev. Lett.*, vol. 64, pp. 821–824, Feb 1990.

[2] F. C. M. Lau and C. K. Tse, *Chaos-Based Digital Communication Systems: Operating Principles, Analysis Methods, and Performance Evaluation*. Springer Verlag, 2003.

[3] M. Itoh, "Spread Spectrum Communication via Chaos," *International Journal of Bifurcation and Chaos*, vol. 9, pp. 155–214, 1999.

[4] M. Xu and H. Leung, "A Novel High Data Rate Modulation Scheme Based on Chaotic Signal Separation," *Communications, IEEE Transactions on*, vol. 58, no. 10, pp. 2855–2860, 2010.

[5] C. C. Chen, K. Yao, K. Umeno, and E. Biglieri, "Design of Spread-Spectrum Sequences using Chaotic Dynamical Systems and Ergodic Theory," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 48, pp. 1110–1114, 2001.

[6] R. Rovatti, G. Setti, and G. Mazzini, "Toward Sequence Optimization for Chaos-Based Asynchronous DS-CDMA Systems," in *Global Telecommunications Conference (GLOBECOM)*, vol. 4, Sydney, Australia, 1998, pp. 2174–2179.

[7] G. Mazzini, G. Setti, and R. Rovatti, "Chaotic Complex Spreading Sequences for Ssynchronous DS-CDMA part I: System modeling and results," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 44, pp. 937–1947, 1997.

[8] R. Rovatti, G. Setti, and G. Mazzini, "Chaotic Complex Spreading Sequences for Asynchronous DS-CDMA part II: Some theoretical performance bounds," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 45, pp. 496–504, 1998.

[9] G. Kaddoum, P. Chargé, D. Roviras, and D. Fournier-Prunaret, "A Methodology for Bit Error Rate Prediction in Chaos-Based Communication Systems," *Circuits, Systems, and Signal Processing*, vol. 28, no. 6, pp. 925–944, 2009.

[10] L. Cong and W. Xiaofu, "Design and Realization of an FPGA-Based Generator for Chaotic Frequency Hopping Sequences," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 48, no. 5, pp. 521–532, 2001.

[11] M. Azzaz, C. Tanougast, S. Sadoudi, and A. Dandache, "Real-time FPGA Implementation of Lorenz's Chaotic Generator for Ciphering Telecommunications," in *Circuits and Systems and TAISA Conference (NEWCAS-TAISA), Joint IEEE North-East Workshop on*, 2009, pp. 1–4.

[12] D. Majumdar, R. Moritz, H. Leung, and B. Maundy, "An Enhanced Data Rate Chaos-based Multilevel Transceiver Design Exploiting Ergodicity," in *Military Communications Conference, (MILCOM)*, Nov 2010, pp. 1256 –1261.

[13] P. Dabal and R. Pelka, "A Chaos-Based Pseudo-Random Bit Generator Implemented in FPGA Device," in *Design and Diagnostics of Electronic Circuits Systems (DDECS), IEEE 14th International Symposium on*, April 2011, pp. 151 –154.

[14] L. Simic and S. Berber, "Performance Analysis of a Chaos-Based Multi-User Communication System Implemented in DSP Technology," in *International Conference on Wireless Broadband and Ultra Wideband Communication*, 2006.