

## Chaos-Based Random Number Generators—Part II: Practical Realization

Toni Stojanovski, Johnny Pihl, and Ljupčo Kocarev

**Abstract**—This paper and its companion (Part I) are devoted to the analysis of the application of a chaotic piecewise-linear one-dimensional (PL1D) map as Random Number Generator (RNG). In Part I, we have mathematically analyzed the information generation process of a class of PL1D maps. In this paper, we find optimum parameters that give an RNG with lowest redundancy and maximum margin against parasitic attractors. Further, the map is implemented in a 0.8  $\mu\text{m}$  standard CMOS process utilizing switched current techniques. Post-layout circuit simulations of the RNG indicate no periodic attractors over variations in temperature, power supply and process conditions, and maximum redundancy of 0.4%. We estimate that the output bit rate of our RNG is 1 Mbit/s, which is substantially higher than the output bit rate of RNGs available on the market.

**Index Terms**—Chaos, CMOS, random number generator.

### I. INTRODUCTION

To construct a random number generator (RNG) based on chaos, in this paper and its companion [1] we exploit the double nature of chaos, deterministic in microscopic space, and random in macroscopic space. In Part I, we have shown that our chaos-based RNG is mathematically proven to act as an information source and is not prone to silent breakdowns. In this paper, we address the questions of practical realization of our RNG. In particular, we find optimum parameters of RNG and show how it can be efficiently implemented on-chip. Here is the layout of our paper. Section II shows the way a RNG based on a chaotic map can be optimized. Section III benefits from the results presented in Section II, and gives the design of a chaotic RNG based implemented by virtue of switched current circuits. Concluding remarks are given in Section IV.

### II. OPTIMIZATION OF CHAOS-BASED RNGS

In Part I, we have analyzed the information generation process of the following PL1D maps:

$$x_{n+1} = f(x_n) = \begin{cases} q_1 + k_1 x_n, & \text{for } x_n < 0 \\ -1 + k_2 x_n, & \text{for } x_n \geq 0 \end{cases} \quad (1)$$

in the region  $\mathcal{P} = \{(k_1, k_2, q_1) \mid 1 < k_1 < 2, k_1 \leq k_2 < 2, k_1 - 1 < q_1 < 1/(k_2 - 1)\}$  of the (three-dimensional) 3-D parameter space  $k_1 \times k_2 \times q_1$ . In this section, we optimize our chaos-based RNG.

#### A. Redundancy Reduction Techniques

As shown in Part I, larger  $k_1$  and  $k_2$  mean smaller redundancy and a better RNG, but they also mean a higher risk of appearance of periodic attractors and of breakdown of the RNG. Therefore,  $k_1$  and  $k_2$  must

Manuscript received January 7, 2000; revised September 17, 2000. This work was supported in part by the Army Research Office under Grant DAAG55-98-1-0269, in part by the Department of Electronics, under Grant DE-FG03-95ER14516, and in part, by the National Science Foundation under Grant NCR-9612250. This paper was recommended by Associate Editor C. K. Tse.

T. Stojanovski is with the Expert Information Services, Melbourne 3027, Australia (e-mail: tonis@expert.com.au).

J. Pihl is with NTNU, Department of Physical Electronics, N-7034 Trondheim, Norway (e-mail: Johnny.Pihl@fysel.ntnu.no).

L. Kocarev is with the Institute for Nonlinear Science, University of California, San Diego, La Jolla, CA 92093-0402 USA (e-mail: kocarev@heisenberg.ucsd.edu).

Publisher Item Identifier S 1057-7122(01)01400-3.

be small enough to assure chaotic behavior of (1) across all temperature and power supply fluctuations. Increased redundancy for smaller  $k_1$  and  $k_2$  must be lowered via processing the output bits. Redundancy in an information source can be due to two sources, difference in the probabilities of the two binary symbols, and memory of an information source. A good redundancy reduction technique must affect both sources of randomness. The two simplest redundancy reduction techniques, which can be on-chip implemented with a very simple circuitry, are bit skipping [2], [3] and bit counting.

In bit skipping only every  $p$ -th bit from the original binary sequence is used. For example, if the original sequence is  $X_0, X_1, X_2, \dots$ , then bit skipping where only every  $p$ -th bit is used will produce the sequence  $X_0, X_p, X_{2p}, \dots$ . Skipping bits reduces only the redundancy due to the memory of an information source, but it does not reduce the difference in the probabilities of the two binary symbols. When  $p \rightarrow \infty$ , the redundancy tends to  $1 - H_B(P\{X = 0\})$  where  $P\{X = 0\}$  is probability of binary symbol 0.

In bit counting, bits from the original binary sequence are grouped in blocks of  $p$  bits and summed up modulo 2 to produce an output bit. For example, if the original sequence is  $X_0, X_1, X_2, \dots$ , then bit counting with blocks of  $p$  bits will produce the sequence  $Y_0, Y_p, Y_{2p}, \dots$ , where  $Y_{ip} = X_{ip} \oplus X_{ip+1} \oplus \dots \oplus X_{ip+p-1}$ , and  $\oplus$  denotes summation modulo 2. For a given  $p$ , a lower limit of redundancy is  $1 - H_B(P\{Y = 0\})$ , where  $P\{Y = 0\}$  is probability of binary symbol 0 in the new sequence  $Y_0, Y_1, Y_2, \dots$ , and  $\lim_{p \rightarrow \infty} P\{Y = 0\} = 0.5$ . Bit counting is equivalent to following redundancy reduction technique. From the original sequence  $X_0, X_1, X_2, \dots$  produce a new sequence  $Z_0, Z_1, Z_2, \dots$  via  $Z_0 = X_0, Z_i = Z_{i-1} \oplus X_i$  for  $i > 0$ , and then apply bit skipping thus yielding the sequence  $Z_p, Z_{2p}, Z_{3p}, \dots$ . Bits from sequences  $Y_0, Y_p, Y_{2p}, \dots$  and  $Z_p, Z_{2p}, Z_{3p}, \dots$  are related via the deterministic transformation  $Z_p = Y_0$  and  $Z_{(i+1)p} = Z_{ip} \oplus Y_{ip}$  for  $i > 0$ , and therefore their entropies and redundancies are identical. Which redundancy reduction technique is preferred between these two, depends on the ease of practical implementation. They can both be implemented with a one-stage binary counter, the only difference being that the binary counter is set to 0 at the start of every block of  $p$  bits. Contrary to bit skipping, bit counting affects both sources of redundancy. This is the reason behind the superiority of bit counting with respect to bit skipping, and this will be pointed out throughout following sections. We show that bit counting is superior to bit skipping in the sense that it is more robust to the inevitable fluctuations of the parameter values from the nominal ones, and provides lower redundancy.

Both bit counting and bit skipping reduce the output bit generation rate  $p$  times, and one must compromise between reduction in the redundancy and reduction in the bit generation rate. Therefore we give results only for moderate values of  $p \leq 6$ . Even for  $p \leq 6$  redundancies are very small, and further reduction in the bit generation rate by choosing larger  $p$  can not be justified.

#### B. Redundancy Reduction in 1-D Regions

Figs. 1 and 2 show redundancies for bit skipping and bit counting in the 1-D regions  $\mathcal{P}_1$  and  $\mathcal{P}_2$ , which are defined as  $\mathcal{P}_1 = \{(k_1, k_2, q_1) \mid 1 < k_2 = k_1 < 2, q_1 = 1\}$  and  $\mathcal{P}_2 = \{(k_1, k_2, q_1) \mid 1 < k_1 = k_2 < 2, q_1 = k_1 - 1\}$ . Common value of  $k_1$  and  $k_2$  is denoted  $k$  in both  $\mathcal{P}_1$  and  $\mathcal{P}_2$ . For  $\mathcal{P}_1$ , where both binary symbols are equiprobable, bit skipping and bit counting are with comparable performances. For  $\mathcal{P}_2$  bit counting is substantially superior to bit skipping. In  $\mathcal{P}_2$ , binary symbols are not equiprobable, and bit skipping offers no cure against such a source of redundancy. As illustrated in Fig. 2(b), redundancies for bit skipping go very quickly

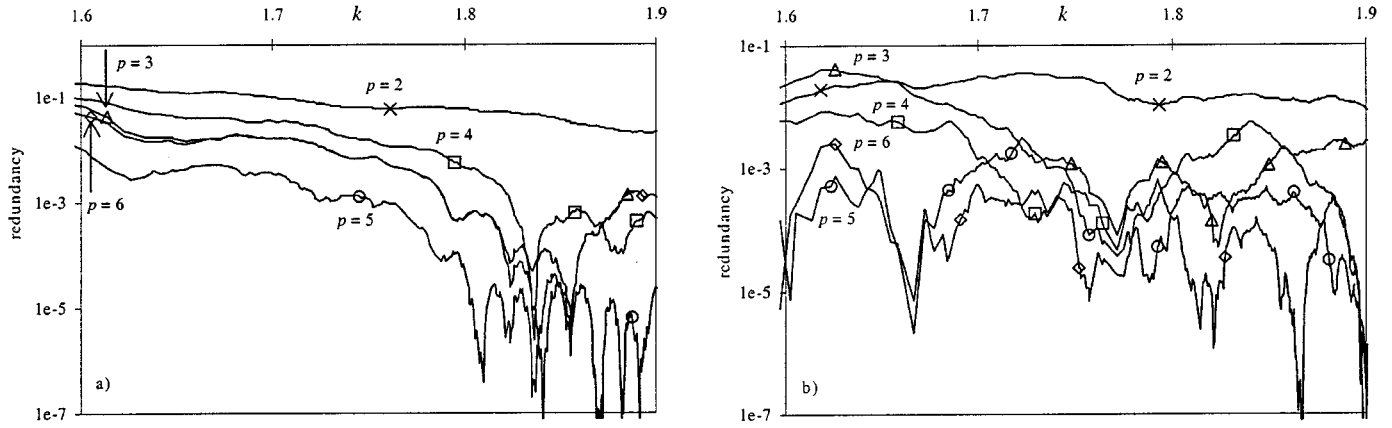


Fig. 1. Redundancy in parameter region  $\mathcal{P}_1$  for: (a) bit counting and (b) bit skipping redundancy reduction techniques.

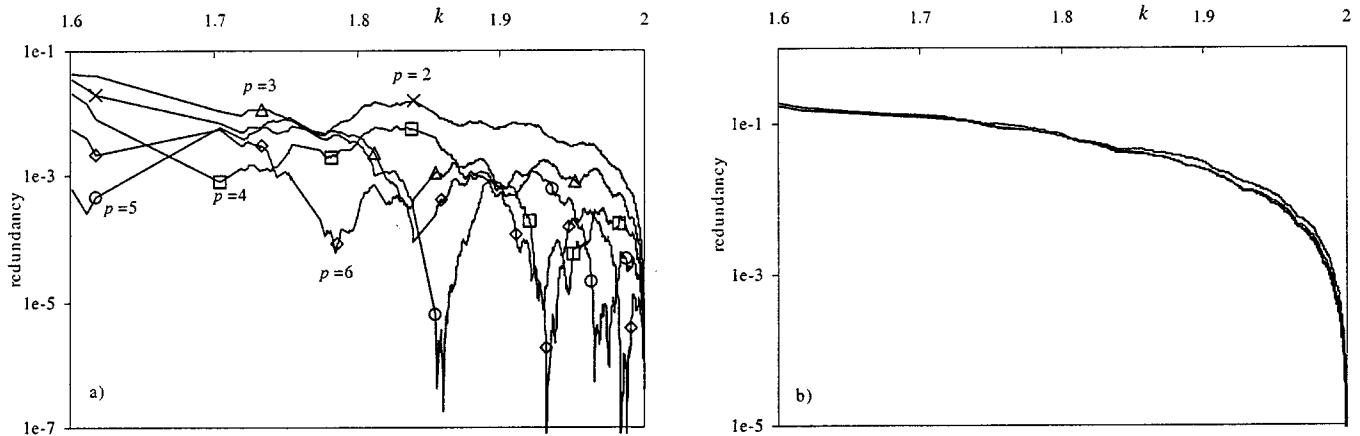


Fig. 2. Redundancy in parameter region  $\mathcal{P}_2$  for: (a) bit counting and (b) bit skipping redundancy reduction techniques.

down to the lower limit  $1 - H_B(P\{X = 0\})$ , which is determined by the differences in probabilities of 0s and 1s.

There are local minima of the redundancy both in  $\mathcal{P}_1$  and  $\mathcal{P}_2$ , that is, larger  $p$  and/or higher slope  $k$  do not always mean smaller redundancy, and smaller redundancy does not necessarily mean lower margins  $L$ . We observed local minima of the redundancy also in the other 1-D regions which we examined. For bit counting in  $\mathcal{P}_1$ ,  $p = 5$  is better than  $p = 6$ , and  $p = 3$  is better than  $p = 4$  for moderate values of  $k < 1.85$ . This is a consequence of the fact that  $P\{Y = 0\} = P\{Y = 1\}$  only for odd  $p$ , while  $P\{Y = 0\} \neq P\{Y = 1\}$  for even  $p$  contributes to higher redundancy. Figures also show that width of local minima decrease for higher  $p$ . Thus, a local minima for higher  $p$  can be difficult to achieve with real circuits where parameter fluctuations are inevitable. This is an additional reason against usage of higher  $p$ .

On the basis of previous results and analysis carried out also in several other 1-D regions of the parameter space, we conjecture that the conclusions on the existence of local minima of the redundancy, on the narrower width of local minima for higher  $p$ , and on the superiority of bit counting to bit skipping in sense of assuring less redundancy are valid also in the 3-D parameter space  $\mathcal{P}$ . The existence of local minima creates possibility for optimization of the choice of parameters, that is, smaller  $k_1$ ,  $k_2$  and  $p$  can be chosen to provide smaller redundancy and higher margin  $L$ .

### C. Optimum Choice of Parameters

What is the optimum choice of parameters? How can we exploit results from this subsection to find the optimum choice? When one uses

map (1) as an RNG, then one wants to be secure against appearance of parasitic attractors. When designing an RNG from the circuit implementation, one can compute the fluctuations in  $I_+$  and  $I_-$  due to temperature, power supply, and fabrication fluctuations, and then one can specify a minimum required margin  $L_{\min}$  against appearance of parasitic attractors. The next requirement is that the bit generation rate is higher than a certain value  $v_b$ , which for a given clock frequency  $v_c$  transforms into a requirement that  $p \leq p_{\max} = \lceil v_c/v_b \rceil$ , where  $\lceil x \rceil$  denotes the largest integer smaller than or equal to  $x$ . For given  $L_{\min}$  and  $p_{\max}$ , we define the optimum parameters as the set of parameters  $(k_1, k_2, q_1, p)_{\text{opt}}$  which minimizes redundancy  $\rho$  amongst all sets of parameters  $(k_1, k_2, q_1, p)$  which satisfy  $(k_1, k_2, q_1) \in \mathcal{P}$ ,  $p \leq p_{\max}$ ,  $L \geq L_{\min}$ , that is,

$$(k_1, k_2, q_1, p)_{\text{opt}} = \min_{(k_1, k_2, q_1) \in \mathcal{P}, p \leq p_{\max}, L \geq L_{\min}} \rho. \quad (2)$$

Brute-force searching of optimum parameters in  $\mathcal{P}$  is a formidable task, and the following discussion relieves it. We show that optimization in the 1-D region  $\mathcal{P}_1$  provides results that are almost as good as those obtained by optimization in the 3-D region  $\mathcal{P}$ .

Instead of searching  $\mathcal{P}$ , we restrict our attention to a 3-D region  $\mathcal{P}_3 = \{(k_1, k_2, q_1) \mid (k_1 \in (1.6, 1.9), k_2 \in (k_1, 1.9), q_1 \in (0.9, 1.1))\}$ . Very small slopes give small  $h^{\beta}$  entropy, while very large ones provide small margin  $L$ . For larger  $|q_1 - 1|$ , the map becomes increasingly asymmetrical and the difference in probabilities of 0s and 1s increases.

It is possible that a local minimum of a redundancy curve in  $\mathcal{P}_1$  (see Fig. 1) is not a local minimum in  $\mathcal{P}$ . Then, a set of parameters lying very close to  $\mathcal{P}_1$  is a local minimum in  $\mathcal{P}$ . This was the motivation to examine 1% and 2% neighborhoods of  $\mathcal{P}_1$ ,  $\mathcal{P}_{1,1\%} = \{(k_1, k_2, q_1) \mid k_1 \leq$

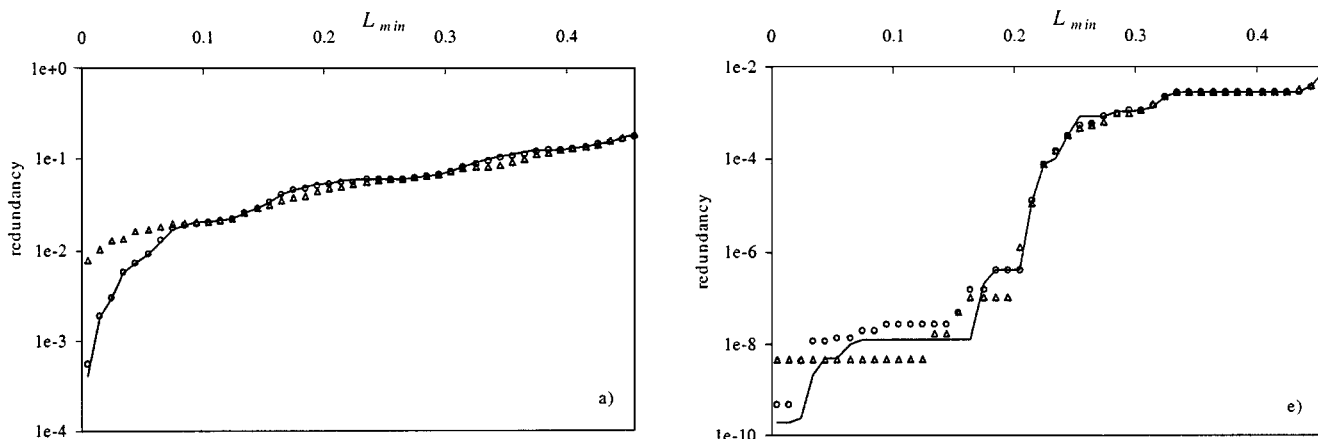


Fig. 3. Optimum redundancy as a function of the minimum margin  $L_{\min}$  in regions  $\mathcal{P}_1$  (solid line),  $\mathcal{P}_3$  (triangles), and  $\mathcal{P}_{1,1\%} \cup \mathcal{P}_{1,2\%}$  (circles) for bit counting and : (a)  $p_{\max} = 2$  and (b)  $p_{\max} = 6$ .

$k_2 \leq 1.01k_1, 0.99 \leq q_1 \leq 1.01$  and  $\mathcal{P}_{1,2\%}$  defined in a similar way to  $\mathcal{P}_{1,1\%}$ .

We divided  $\mathcal{P}_3$  into  $115\,351 = 61 \times 61 \times 31$  equal cubes. For a point  $(k_1, k_2, q_1)$  lying in  $\mathcal{P}_1$  we define its 1% neighborhood as  $\{(k_1, k_2, q_1) \mid k_1 \leq k_2 \leq 1.01k_1, 0.99 \leq q_1 \leq 1.01\}$ . We divided the 1% neighborhoods of 80 points  $k_1 = 1.6, 1.605, 1.61, \dots, 1.995$  into 288 equal cubes thus producing a total of 23 040 cubes in  $\mathcal{P}_{1,1\%}$ . An analogue procedure was repeated for  $\mathcal{P}_{1,2\%}$ . For each of the  $115\,351 + 23\,040 + 23\,040$  cubes we found an inner point for which  $\beta$  is a Markov partition of order  $r \leq 12$  (12 is chosen because the execution time and the memory requirements of the computer program that computes redundancies grow exponentially with  $r$ ). Then for each of these inner points we computed redundancies of bit skipping and bit counting for  $2 \leq p \leq 6$ .

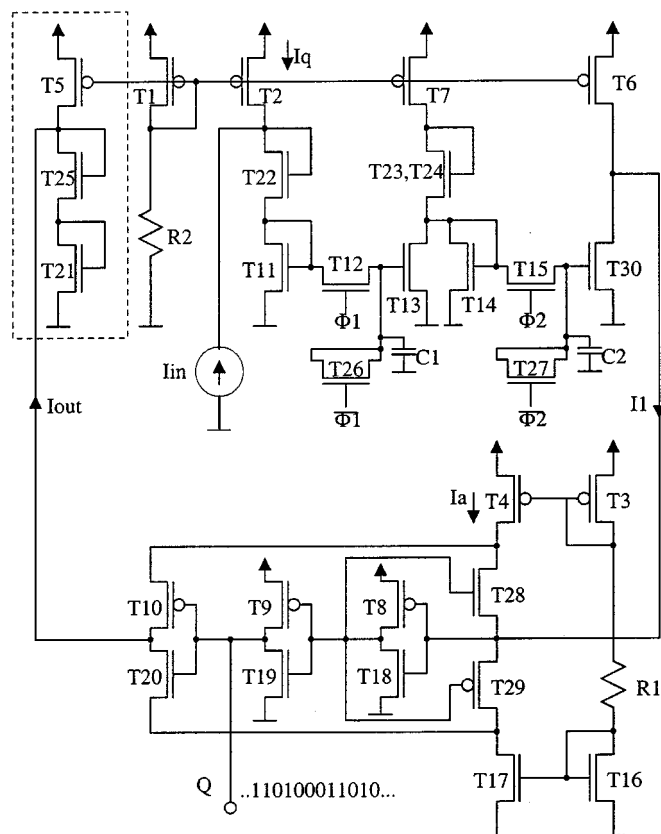
This way, we examined behavior of redundancy in the 1% neighborhood of  $\mathcal{P}_1, \mathcal{P}_{1,1\%} = \{(k_1, k_2, q_1) \mid k_1 \leq k_2 \leq 1.01k_1, 0.99 \leq q_1 \leq 1.01\}$ . We repeated this procedure for a 2% neighborhood  $\mathcal{P}_1$ .

On the basis of previous computations, we performed optimization (2) in  $\mathcal{P}_1, \mathcal{P}_3$ , and  $\mathcal{P}_{1,1\%} \cup \mathcal{P}_{1,2\%}$  for  $0 < L_{\min} < 0.45$  and  $2 \leq p_{\max} \leq 6$ . We present results only for  $p_{\max} = 2$  and  $p_{\max} = 6$ , see Fig. 3. For large  $L_{\min}$  (larger than approximately 0.15), optimization in  $\mathcal{P}_1$  provides very close results to optimization in  $\mathcal{P}_3$  and  $\mathcal{P}_{1,1\%} \cup \mathcal{P}_{1,2\%}$ , while for smaller  $L_{\min}$  optimizing in  $\mathcal{P}_1$  is superior to optimizing in  $\mathcal{P}_3$  and  $\mathcal{P}_{1,1\%} \cup \mathcal{P}_{1,2\%}$ . The latter superiority is due to the small width of minima for  $k_1, k_2 \approx 2$ , and the chance is small that these minima are hit by the finite number of points analyzed in  $\mathcal{P}_3, \mathcal{P}_{1,1\%}$ , and  $\mathcal{P}_{1,2\%}$ . A twice finer grid of points in  $\mathcal{P}_3$  and  $\mathcal{P}_{1,1\%}$  would increase the number of points 8 times. In contrast, in  $\mathcal{P}_{1-D}$  we analyzed only 574 points for which  $\beta$  is a Markov partition with order  $r < 12$ . Motivated by (i) the substantially smaller computational complexity of the analysis and optimization in  $\mathcal{P}_1$  than in  $\mathcal{P}_3, \mathcal{P}_{1,1\%}$ , and  $\mathcal{P}_{1,2\%}$ ; and (ii) the comparable or superior quality of the optimization in  $\mathcal{P}_1$  compared to the optimization in  $\mathcal{P}_3, \mathcal{P}_{1,1\%}$ , and  $\mathcal{P}_{1,2\%}$  we suggest only an optimization in  $\mathcal{P}_1$  is done when bit counting is employed.

### III. PRACTICAL REALISATION

#### A. Circuit Design

Parameter variations due to implementation imprecision and external influences (temperature, power supply etc.) need to be estimated. As they are slowly varying compared to the iteration speed of the map, their temporal changes can be neglected and it can be approximated that the parameters are constant in time, though mismatched from the nominal ones.



T1..T5:  $10\mu$  T6:  $20\mu$  T7:  $19.4\mu$  T8..T10:  $13\mu$  T11..T21:  $5\mu$   
T22..T25:  $1.3\mu/2\mu$  T26:  $3.3\mu$  T27:  $3\mu$  T28:  $4\mu$  T29:  $8\mu$   
T30:  $9.9\mu$  C1..C2:  $1\text{pF}$

Fig. 4. Transistor diagram for VLSI implementation in switched current technique, of the PLID map. It shows the open-loop setup when the design is simulated after layout in a  $0.8\mu\text{m}$  CMOS process. During normal operation  $I_{\text{out}}$  is closed to  $I_{\text{in}}$ , and a nonoverlapping two-phase clock is applied to  $\phi_1$  and  $\phi_2$ .

We now look at how the chaotic map

$$x'_{n+1} = \begin{cases} q'_1 + k'_1(x'_n - T_L), & \text{for } x'_n < T_L \\ q'_2 + k'_2(x'_n - T_L), & \text{for } x'_n \geq T_L \end{cases} \quad (3)$$

may be implemented in VLSI technology. Fig. 4 shows a VLSI implementation of map (3), in a standard  $0.8\mu\text{m}$  CMOS process [5]. The implementation is a switched-current based circuit based on [4].

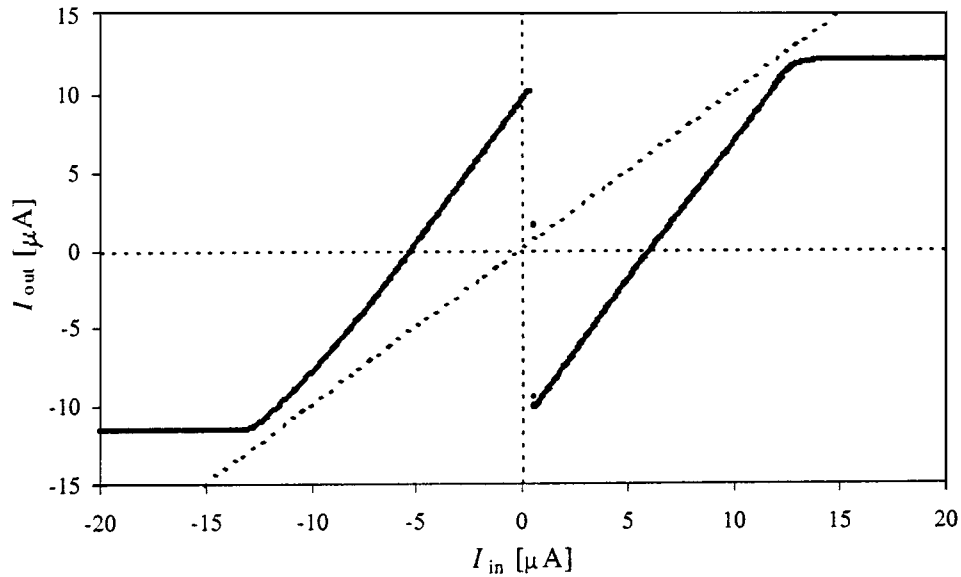


Fig. 5. Map function obtained by open-loop SPICE simulation utilising Level 15 MOS models under typical mean process conditions. No parasitic stable points exist for temperatures between  $-25^{\circ}\text{C}$  and  $+75^{\circ}\text{C}$  and  $\pm 10\%$  power supply fluctuation.

The upper half of the circuit performs the slope multiplication and storage operation, and the lower half performs the nonlinear discrimination function [4]. The discriminator operates in one of two modes: For  $I_1 > 0$  T18, T19 are on and T20, T29 conduct so that through T17  $I_{\text{out}} = I_1 - I_a$ . For  $I_1 < 0$  T8, T9 are on and T10, T28 conduct so that through T4  $I_{\text{out}} = I_1 + I_a$ . The figure shows the setup for open-loop simulation where the output current is terminated in a stage equivalent to the input. During closed-loop operation, the input and output are connected and a two-phase nonoverlapping clock is applied to  $\Phi_1$  and  $\Phi_2$ . The circuit was designed for a nominal threshold value of  $T_L = 0$ , and a nominal slope of 1.82. This corresponds to one of the minimums of the redundancy curve for  $p = 5$ . The slope is small enough to provide a good margin  $L = 19.6\%$ .

#### B. Circuit Analysis

With the circuit extracted from layout, the design was simulated open-loop in SPICE across 4.5...5.5 V power supply range and temperatures  $-25^{\circ}\text{C}$  to  $+75^{\circ}\text{C}$ , at typical mean process conditions. The proprietary charge based transistor model from Austria Mikro Systems (AMS) (Level 15) was used [5].  $I_q$  was selected to  $16\ \mu\text{A}$  and  $I_a$  to  $12\ \mu\text{A}$ . For each pair of temperature and power supply, redundancies for bit counting and bit skipping with  $p = 2, \dots, 6$  were computed. Maximum redundancy over all temperatures and power supplies is minimum for  $p = 5$  with a value of 0.4%. No parasitic attractors were detected. This was found to be true also for different process corners except for worst case speed process parameters. A different setting of  $I_q$  and  $I_a$ , yielding no parasitic attractors could, however, be found also for this process corner. The map, obtained from circuit simulation after layout is shown in Fig. 5 ( $27^{\circ}\text{C}$ ,  $+5\text{V}$ ).

The slope is 1.82 at 5 V and  $27^{\circ}\text{C}$ . Simulations also indicated a maximum clock feed-through in  $I_{\text{out}}$  of  $0.3\ \mu\text{A}$  across its entire range. On a step input from  $-9\ \mu\text{A}$  to  $+9\ \mu\text{A}$  in  $I_{\text{in}}$ ,  $I_{\text{out}}$  settles to within  $0.1\ \mu\text{A}$  in 140 ns. Maximum operating clock frequency is estimated to 5 MHz, which together with bit counting with  $p = 5$  yields a total RNG

bit rate of 1 Mbit/s. This is substantially higher than the output bit rate of the RNGs available on the market: from 7600 bits/s to 76 000 bits/s. Furthermore, our RNG requires no software postprocessing.

#### IV. CONCLUSION

We have addressed the practical issues of our chaos-based RNG. Performances of simple redundancy reduction techniques such as bit counting and bit skipping, which due to their simplicity can be implemented on-chip, are analyzed. Bit counting is superior to bit skipping as it yields smaller redundancy, provides higher robustness of redundancy to fluctuations in parameters, and optimization of parameters can be done in a much simpler mode. Post-layout circuit simulations of the RNG indicate maximum redundancy of 0.4% for bit counting with  $p = 5$ , and no parasitic attractors over variations in temperature, power supply and process conditions. Maximum operating clock frequency is estimated to 5 MHz, which together with bit counting with  $p = 5$  yields a total RNG bit rate of 1 Mbit/s. This is substantially higher than the output bit rate of the RNGs available on the market.

#### REFERENCES

- [1] T. Stojanovski and L. Kocarev, "Chaos based random number generators Part I: Analysis," *IEEE Trans. Circuits Syst. I*, vol. 48, pp. 281–288, Mar. 2001.
- [2] S. Espejo-Meana, A. Rodriguez-Vazquez, J. L. Huertas, and J. M. Quintana, "Application of chaotic switched-capacitor circuits for random-number generation," in *Proc. Eur. Conf. Circuit Theory Design 1989*, 1989, pp. 440–444.
- [3] G. M. Bernstein and M. A. Lieberman, "Secure random number generation using chaotic circuits," *IEEE Trans. Circuits Syst.*, vol. 37, pp. 1157–1164, Dec., 1990.
- [4] M. Delgado-Restituto, F. Medeiro, and A. Rodriguez-Vazquez, "Non-linear switched-current CMOS IC for random signal generation," *Electron. Lett.*, vol. 29, no. 25, pp. 2190–2191, 1993.
- [5] *0.8 μm CMOS process*, Austria Mikro Systems, Unterpremstatten, Austria, 1995.