TOWARDS A UNIQUE FPGA-BASED IDENTIFICATION CIRCUIT USING PROCESS VARIATIONS

H. Yu, P.H.W. Leong *

Dept. Computer Science and Engineering
The Chinese University of Hong Kong
email: {hlyu,phwl}@cse.cuhk.edu.hk

H. Hinkelmann, L. Moller, M. Glesner

Institute of Microelectronic Systems
Technische Universitat Darmstadt (TUD)
email: hinkelmann@mes.tu-darmstadt.de
{moller,glesner}@mes.tu-darmstadt.de

P. Zipf

Digital Technology Lab
University of Kassel
email: zipf@uni-kassel.de

ABSTRACT

A compact chip identification (ID) circuit with improved reliability is presented. Ring oscillators are used to measure the spatial process variation and the ID is based on their relative speeds. A novel averaging and postprocessing scheme is employed to accurately determine the faster of two similar-frequency ring oscillators in the presence of noise. Using this scheme, the average number of unstable bits i.e. bits which can change in value between readings, measured on an FPGA is shown to be reduced from 5.3% to 0.9% at $20°C$. Within the range $20 - 60°C$, the percentage of unstable bits is within 2.8%. An analysis of the effectiveness of the scheme and the distribution of the errors is given over different temperature ranges and FPGA chips.

1. INTRODUCTION

Generating a unique identification number (ID) for a production chip is often desirable in embedded systems. Examples of applications where IDs can be used include: generating keys for a public-key cryptosystem; for identification of users or products in smartcards, keyless entry systems and RFID devices; to identify nodes on networks, and for digital rights management.

Techniques used to provide an on-chip ID include writing a unique number to a non-volatile memory such as a ROM or FLASH, post-fabrication modification of the chip using lasers or fuses, and creating a signature from variations in mismatch on the chip. For example, the Dallas Semiconductor iButton employs laser etching to provide a unique, unalterable 64-bit key. Members of the iButton family all have the ID but can also include non-volatile memory, a real-time clock, data logger, monetary devices with SHA1 challenge response, password protected memory and temperature/humidity loggers.

The traditional way of storing ID information in an FPGA is in volatile or non-volatile memory. For example, in Xilinx Virtex devices, a bitstream can be encrypted using a secret key and the same key stored on the FPGA device. When the bitstream is downloaded, a hardware decryption core decrypts the bitstream using the key. Thus the bitstream can only operate with devices programmed with the secret key. This key is stored in RAM and it is not possible to read back the value from the device. A battery is also needed to preserve its contents after powerdown and RAM is used to increase its resistance to tampering over flash or antifuse technologies which can be attacked by thermal imaging and other techniques [1]. Unfortunately, the Xilinx bitstream ID is not available for other applications as one of its features is that it cannot be read back. In this paper we present a technique to produce IDs with improved repeatability over previously reported ring oscillator(RO) based schemes.

It is possible to generate a chip ID from intra-chip process variations and previous work in this area is reviewed in the next section. IDs with approximately 5% unstable bits, i.e. bits which change in value for different readings, at room temperature were reported [2, 3, 4]. Errors occur because analogue measures of process variation are thresholded and converted to a binary number. As the measure-

*The authors gratefully acknowledge support from the Research Grants Council of the Hong Kong Special Administrative Region, China (Earmarked Grant CUHK413707) and the Germany/Hong Kong Joint Research Scheme (grant G_HK007/07).

ments involve thermal noise, consecutive readings can be different and so there is always a finite probability of it changing between readings. Previous work assumed that the code is recorded either in a memory or database. Subsequent readings are compared with this stored value using a metric such as Hamming distance and a match assumed if the distance is sufficiently small.

In many applications, a unique ID is required. Whether or not a practical unique ID with no unstable bits can be generated from process variation alone remains an open question. In this paper, we present a scheme for generating chip IDs with higher repeatability than previously reported designs. The contributions of this work include:

- a novel circuit which uses an adaptive averaging technique to generate a chip ID with a lower number of unstable bits than previously reported
- an analysis of the effect of temperature variation on chip ID generation
- a statistical analysis of the error properties of the chip ID scheme from a statistically significant number of FPGA chips.

The rest of this paper is organised as follows. Section 2 describes background to this work. The design of the circuits used to generate the ID are described in Section 3. Measured results and an analysis are presented in Section 4. Finally, conclusions are drawn in Section 5.

2. BACKGROUND

Analog integrated circuit identification circuits have been previously proposed. Lofstrom et. al. [2] used an array of addressable NMOS transistors loaded with a common resistive load. Drain current mismatch caused the voltage across the load to be different for different transistors in the array. By addressing the transistors in the array in a sequential fashion, a sequence of voltages was generated and successive values converted to a binary sequence via an auto-zeroing comparator to form an ID. A 112-bit ID circuit was shown to have a drift of less than 4% over a wide supply, voltage and temperature range. Su et. al. [4] reported on an improved circuit which used cross-coupled logic gates to simultaneously generate, amplify and digitise transistor mismatch. This circuit was able to produce a 128-bit, 96% stable ID using only 1.6 pJ/bit.

Guajardo et. al. [5] use the initialisation state of static RAM cells in an FPGA and showed that they had good statistical properties for producing an ID. Experiments showed that 4% of the startup bits from the same RAM changed over time. Over a $-20°C$ to $80°C$ temperature range, bit strings had a maximal fractional Hamming distance of 12% compared to a reference at 20°C.

Suh and Devadas [3] used ring oscillators to generate secret keys for cryptographic operations. They used the difference in speed of an array of ring oscillators to derive bits of an ID. They proposed using ring oscillators with frequencies that are far apart to improve the robustness of the generated ID. In particular, a 1-out-of-k masking scheme with $k = 8$ was used. A sequence of ring oscillator pairs which is k times longer than the desired binary output length is generated. For each k pairs, the pair with maximum distance was chosen. A bit vector of these selections is saved so that the same pairs can be used to re-generate the output. A BCH error correcting code was also used to further improve repeatability.

Common features of all of the research reviewed in this section are that their unstable bit percentage was approximately 5%, and that all measurements were based on a single reading which has low signal to noise ratio. Other issues such as negative bias temperature instability(NBTI) may also affect ID generation but is beyond the scope of this paper.

3. DESIGN

3.1. Measurement Circuits

The circuitry to generate the chip ID consists of two main parts: an array of identical ring oscillators, and a controller to measure and compare their differential delay. A top level block diagram of the circuit is shown in Figure 1. The design of the controller is based a design by Sedcole and Cheung [6]. A total of 64 individual ring oscillator cells, labelled ABCD, are selected via decoding logic and the ID generation circuitry is shared. A finite state machine controls an internal timer and address generation so that the ID can be generated sequentially. In the physical layout, the control circuitry is separated from the ring oscillator array to reduce the chance of coupling between the two. As common in analogue layout, dummy cells were also used around the outside of the array so that each of the ring oscillator cells has identical neighbouring elements.

A ring oscillator is formed from a loop of K inverters, where K is odd. The output is a square wave with frequency $f = \frac{1}{2KD}$ where D is the propagation delay of the inverter. Ring oscillators are often used to characterise ICs. Transistor parameters can vary between devices on a chip due to random variations in the IC fabrication process. It is possible to characterise process variations on an FPGA by measuring ring oscillator frequency of different lookup tables (LUTs) on the device [6].

The work in this paper was performed using Xilinx Spartan 3e FPGA devices but could be adapted to any programmable logic device or integrated circuit. In the Spartan 3e, registers require a high and low pulse width of at least 0.8 ns and a frequency below 572 MHz. Our design meets this specifica-

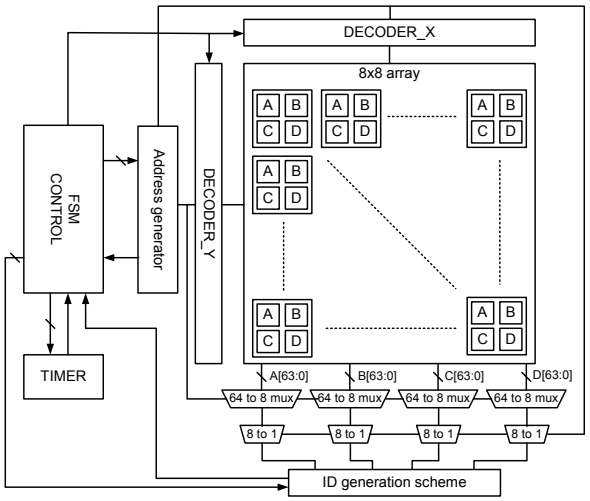

Fig. 1. Block diagram of the measurement circuit.

tion. Each ring oscillator is connected to the clock input of a counter and the number of ring oscillator clock periods occurring during a timer interval T is measured. The output of the counter, C, is hence proportional to the frequency f of the ring oscillator. Although the wire delays from the ring oscillator outputs to the counter clock input are different, they do not influence the measurements because the delay has no effect on the ring oscillator frequency. Likewise, the wire delay of the enable signal has no influence on the measurements, assuming that the delays for the rising and the falling transition of the enable signal are roughly the same.

Spatial gradients in doping concentration or other properties on a die can cause a systematic change in transistor speed across a chip or wafer. In order to reduce such effects, common centroid designs are commonly used in analogue layout to improve matching properties of devices and cells [7]. We employ such a scheme, the four neighbouring ABCD ring oscillators being arranged in a square and occupying the same lookup table position in their individual CLB. The counter outputs for bit i, A_i, B_i, C_i and D_i are combined to produce an output R_i, via the formula:

$$R_i = (A_i + D_i) - (B_i + C_i). \quad (1)$$

Measurements can be influenced by temperature, both internal and external. As the ring oscillators cause self-heating, they are only turned on when being measured. This is implemented by changing one of the inverters in the ring oscillator to a NAND gate with one input used as an enable signal.

3.2. ID Generation

A *simple scheme* for generating an ID is to assign the output bit $B_i = sgn(R_i)$ where sgn is zero if R_i is negative, otherwise one [4]. Unfortunately, this method has several shortcomings: it is based on a single reading and may be affected by various sources such as the power supply and thermal transistor noise; the measurements are only made at one temperature; and the signal to noise ratio may be low as we are trying to generate the ID by measuring the frequency of ring oscillators whose values are very close together.

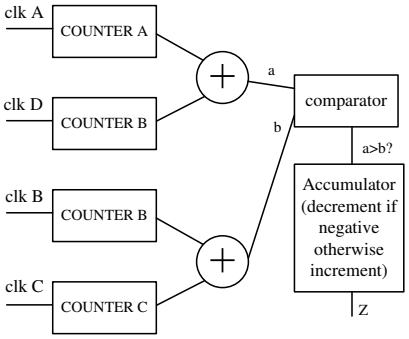

Fig. 2. Datapath for one bit of the ID generation scheme.

In our proposed approach, randomness is reduced by averaging and we compare the $A_i + D_i$ and $B_i + C_i$ values of Equation 1 until the difference is larger than a given threshold. This is done by introducing an additional counter Z_i which is initialised to zero and incremented if $R_i > 0$ in the timer period and decremented otherwise. We repeat the measurements until the absolute value of Z_i is above a global threshold P, and set $B_i = sgn(Z_i)$. The datapath used to implement this scheme is shown in Figure 2.

This scheme was designed so that if the mean magnitude of the R_i value is large, it will quickly cause the Z_i to pass the threshold P. For cases where R is close to zero, more evaluations are needed, causing more averaging to be done and hence improving the signal to noise ratio. We note that the simple scheme is a special case of this scheme, i.e. the two schemes are exactly the same for $P = 1$.

Finally, the postprocessing scheme given in pseudocode form below was employed. A binary digit, B, is first generated using the method described above and the final decision as to the binary output for a bit is made by considering whether there are more 0's or 1's within N consecutive bits.

```
// Total bitwidth is W=64
// Number of ids to be averaged is N
// jth binary digit generated
//  in bit position i is B[i][j]
for i = 0 to W {
    bit = 0;
    for (j = 1 to N)
        bit += B[i][j];
    if (bit > N/2)
        id[i] = 1;
    else
        id[i] = 0;
}
```

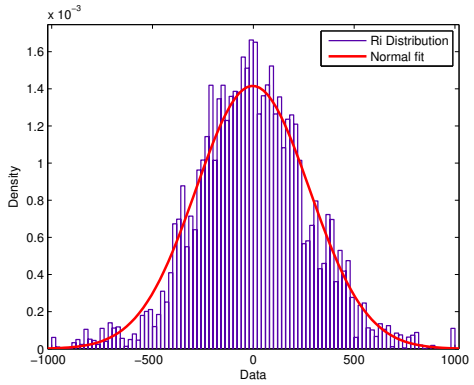

Fig. 3. R_i distribution.

4. RESULTS

The design described was implemented using the Xilinx ISE Design Suite 9.2i. Nine similar boards designed in our lab were randomly selected for testing, each board having an NXP LPC2131 ARM processor and a Xilinx Spartan 3s250-epq208-4 FPGA. IDs were generated on the FPGA and read back using the ARM processor. Relationally placed macros (RPMs) and placement constraints were used to manually specify the location of the ring oscillator array and controller. Of the 2448 slices in the FPGA, 245 (10%) are used by the design and interface and a maximum clock frequency of 180 MHz was reported by the tools.

Distribution of R_i. The distribution of bits generated by the ID scheme are assumed to be independent and identically distributed (IID) with equal probability of a zero or one. The residue R_i (calculated using Equation 1), was recorded for all of the boards used in this study and a histogram is shown in Figure 3. As can be seen from the figure, the distribution is Gaussian in shape with a mean value of -2.814 and a standard deviation of 281.73. An Anderson-Darling test [8] for a normal distribution was conducted. It confirmed that the sample R_i is normally distributed at a 5significance level. Unstable bits occur in the chip id because the probability density is highest in the region where the mean is close to 0, degrading stability.

For n chips using W-bit IDs, the probability of a collision is $1/2^W$ and the probability that a given chip ID will not collide with the remaining $n-1$ others is $1-\frac{n-1}{2^W}$. Thus the probability of an ID collision among Y chips is $P_{coll} = 1 - \prod_{n=1} Y(1-\frac{n-1}{2^W})$ [4].

Effect of P, T, N. The effect of varying the averaging parameters was studied and is presented in this subsection. In all experiments in the rest of this paper, measurements were taken over 50 consecutive trials. Table 1 shows the effect of varying P and T. As expected, increasing either of these parameters increases averaging which reduces the number of unstable bits observed. The choice of $P = 16$ and $T = 2000$ was made as this gives the best results with the smallest execution time and subsequent results were obtained with this setting.

P/T	1000	2000	4000
1	8	4	6
16	4	2	4
256	2	2	2

Table 1. Number of unstable bits observed in board 4 for different P (row) and T (column) values.

Board	1	5	10	20	40	80
1	4	1	1	1	1	0
2	2	0	0	0	0	0
3	2	0	0	0	0	0
4	4	3	2	1	1	0
5	4	2	2	1	0	0
6	4	4	3	2	1	1
7	3	1	1	0	0	0
8	6	5	3	3	3	2
9	2	1	1	1	1	1

Table 2. Effect of increasing N on the number of unstable bits.

The effect of varying N on the number of unstable bits is summarised in Table 2. Although not completed extinguished, the total percentage of unstable bits across all boards is reduced from 5.3% $(31/(64 \times 9)$ for $N = 1$ to 0.9% $(4/(64 \times 9)$ for $N = 80$. Table 3 gives a more quantitative view of the effect of N. It can be seen that increasing the N parameter serves to drive the outputs towards all zeros or all ones and hence better stability of the output. Among the 9 boards tested, 6 had no ID errors.

	Index		
	12	26	34
N=1	0.33:1	0.052:1	0.285:1
N=5	0.19:1	0.03:1	0.219:1
N=10	0.15:1	stable	0.12:1
	Index		
	42	59	63
N=1	1:0.004	1:0.068	0.66:1
N=5	stable	1:0.020	0.56:1
N=10	stable	stable	0.32:1

Table 3. Effect of N on the 1:0 ratio.

Taking board 11 as an example, we illustrate how the averaging scheme can improve stability. Figure 4 shows consecutive B_i outputs for $i = 59$ and $i = 63$. In the top two figures, it can be seen that bit 63 oscillates between the two outputs, while bit 59 has a much lower number of transitions with a significant bias. This improves as N is increased to

5 (middle two figures) and then 10 (bottom two figures). In this case, increasing N serves to stabilise bit 59 making its readings reliable. In the case of bit 64, it can be seen that increasing N improves its stability, however, the output is still not stable for $N = 10$. It is difficult to make bit 64 stable even if a very large N is used.

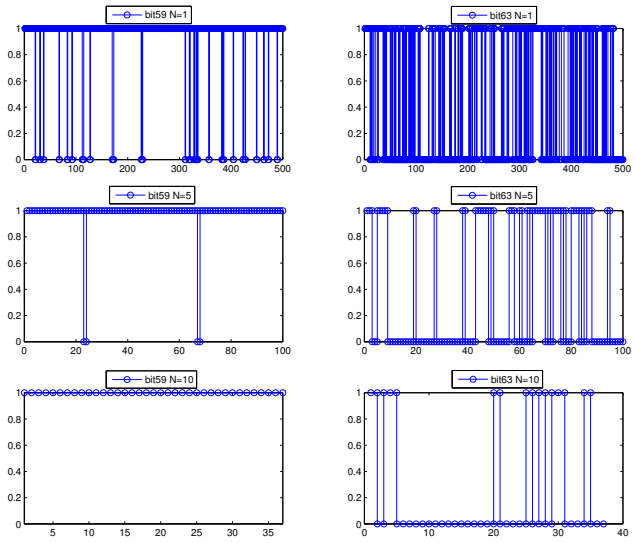

Fig. 4. Figure illustrating how postprocessing can stabilise the output in some cases (left) and not others (right). The trial number is shown on the X axes and the output bit on the Y axis.

Distribution of 1's and 0's. Using settings of $P = 16, T = 2000$ and $N = 80$, measurements averaged across all of our boards gave a ones to zeros ratio of 1.0799 and their distribution is shown in Figure 5. These measurements are consistent with what would be expected if the bits are IID without any bias.

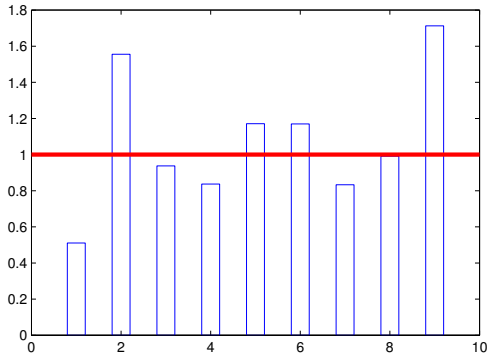

Fig. 5. One to zero ratio of bits for different boards.

	1	2	3	4	5	6	7	8	9
1	0	27	23	24	34	27	29	33	32
2	27	0	32	31	29	28	30	26	31
3	23	32	0	21	27	26	20	28	33
4	24	31	21	0	34	29	27	29	26
5	34	29	27	34	0	23	25	29	28
6	27	28	26	29	23	0	16	30	27
7	29	30	20	27	25	16	0	30	25
8	33	26	28	29	29	30	30	0	29
9	32	31	33	26	28	27	25	29	0

Table 4. Hamming distance matrix.

	T=1000	T=2000	T=4000
P=1	1.295 ms	2.575 ms	5.135 ms
P=16	20.74 ms	41.16 ms	84.35 ms
P=256	373.8 ms	661.2 ms	1449 ms

Table 5. Mean execution time.

Hamming Distance. In order to verify that the method for generating IDs had minimal systematic spatial dependence, the Hamming distance between IDs was measured and is shown in Table 4. For W IID bits, we expect that the average Hamming distance be $W/2$ and the measured average of 28 is consistent with this value.

Execution Time. The average execution time for evaluating a 64-bit ID is given in Table 5 and it can be seen to be proportional to N and P. In the current design, a single bit is computed at a time. ID generation could be sped up by evaluating bits in parallel and in the current design, much of the hardware resources are unused. Moreover, we note that in many applications, once an ID has been generated and stored, partial reconfiguration of the FPGA without the ID generation circuit can be applied to save resources.

Variation with Temperature. The effect of temperature was measured by heating the chip with a hair-dryer and recording the R_i value (Equation 1). A thermocouple connected to a digital multimeter was used to monitor the chip surface temperature and it was found that the maximum temperature we could achieve was $60°C$. The results are shown in Figure 6. Since R_i is a differential measurement, local changes in ring oscillator frequency due to temperature track each other making it less sensitive to temperature. However, as the temperature increases, the R_i values are inevitably affected, particularly those close in value to zero. The ovals in Figure 6 show the unstable bits caused by changing temperature, the other bits do not change with temperature. More detailed views of particular unstable bits are also shown in Figure 7 for which it can be seen that bits 4, 19 and 60 change sign with increasing temperature. In Table 6, it can be seen that the percentage of unstable bits for $N = 1$ and $N = 80$ increase to 6.4% (37/(64×9)) and

Board	N=1	N=80	Board	N=1	N=80
1	4	3	8	5	2
2	4	2	10	6	2
4	1	0	11	6	3
5	2	1	12	3	1
6	6	2			

Table 6. Number of unstable bits in board 11 over the $20-60°C$ temperature range.

2.8% (16/(64×9)) respectively within the temperature range of $20-60°C$.

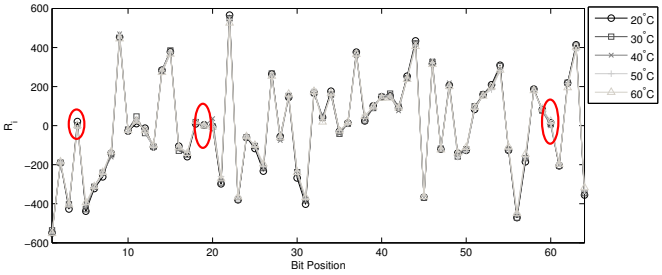

Fig. 6. R_i over different temperatures.

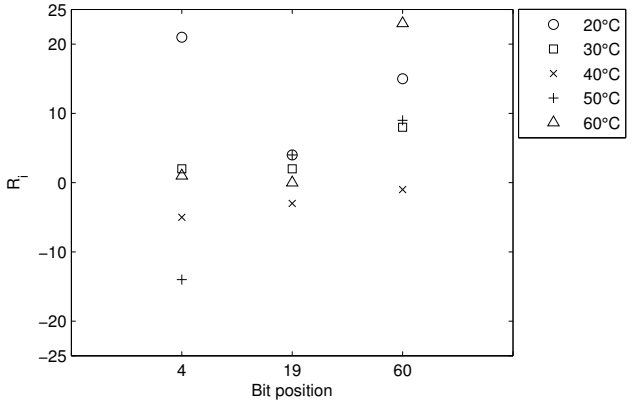

Fig. 7. R_i of unstable bits over different temperatures.

5. CONCLUSIONS

Previous chip ID schemes required approximate matching schemes due to a relatively high probability of unstable bits (around 5%). In this work, we demonstrated the feasibility of greatly improving this value through a new circuit and averaging scheme. The proposed ID circuit used a common centroid layout to reduce the effect of spatial gradients on the generated ID. Adaptive amounts of averaging were applied to improve the signal to noise ratio in determining which of two R_i values is larger in the presence of noise. By varying the parameters P, T and N, it was shown that a balance between ID generation time and reliability can be achieved.

Further research is required before unique IDs can be generated with low probability of error.

In future work we hope to further improve the ID generation scheme. We will study the effect of different ring oscillator array layouts and whether self-heating effects may be occurring as well as develop more sophisticated post-processing schemes.

6. REFERENCES

[1] A. Telikepalli, "Is your FPGA design secure?" *Xilinx XCELL*, vol. Fall, 2003.

[2] K. Lofstrom, W. R. Daasch, and D. Taylor, "IC identification circuit using device mismatch," in *Proceedings of the International Solid State Circuits Conference (ISSCC)*, 2000, pp. 372–373.

[3] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *DAC '07: Proceedings of the 44th annual conference on Design automation*. New York, NY, USA: ACM, 2007, pp. 9–14.

[4] Y. Su, J. Holleman, and B. Otis, "A digital 1.6 pJ/bit chip identification circuit using process variations," *Solid-State Circuits, IEEE Journal of*, vol. 43, no. 1, pp. 69–77, Jan. 2008.

[5] J. Guajardo, S. S. Kumar, G.-J. Schrijen, and P. Tuyls, "FPGA intrinsic PUFs and their use for IP protection," in *CHES '07: Proceedings of the 9th international workshop on Cryptographic Hardware and Embedded Systems*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 63–80.

[6] P. Sedcole and P. Y. K. Cheung, "Within-die delay variability in 90nm FPGAs and beyond," *Field Programmable Technology, 2006. FPT 2006. IEEE International Conference on*, pp. 97–104, Dec. 2006.

[7] A. Hastings, *The Art of Analog Layout*. Prentice Hall, 2001.

[8] T. W. Anderson and D. A. Darling, "Asymptotic theory of certain "goodness of fit" criteria based on stochastic processes," in *Ann. Math. Statist.*, vol. Volume 23, 1952, pp. 193–212.