A Ring Oscillator Based Truly Random Number Generator

by

Stewart Robson

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2013

Abstract

Communication security is a very important part of modern life. A crucial aspect of security is the ability to identify with near 100% certainty who is on the other side of a connection. This problem can be overcome through the use of random number generators, which create unique identities for each person in a network. The effectiveness of an identity is directly proportional to how random a generator is. The speed at which a random number can be delivered is a critical factor in the design of a random number generator.

This thesis covers the design and fabrication of three ring oscillator based truly random number generators, the first two of which were fabricated in 0.13μm CMOS technology. The randomness from this type of random number generator originates from phase noise in a ring oscillator.

The second and third ring oscillators were designed to have a low slew rate at the inverter switching threshold. The outputs of these designs showed vast increases in timing jitter compared to the first design. The third design exhibited improved randomness with respect to the second design.

Acknowledgements

I would like to thank my supervisors, Dr. Bosco Leung and Dr. Guang Gong, for their time and great assistance with my research.

I would also like to acknowledge my reviewers, Dr. Vincent Gaudet and Dr. Peter Lavine, for their helpful comments and feedback on my work.

I am grateful of fellow graduate student Mohamed Amin for his help and guidance with all Cadence and fabrication related issues as well as for his advice on the architecture of my circuit designs.

My gratitude goes to Canadian Microelectronics Cooperation (CMC) and its staff for fabricating my circuits. Special thanks to Mariusz Jarosz for providing me with the necessary testing equipment.

I would also like to acknowledge and thank Allison Bawden and my family for their love and support.

Dedication

To Allison, Rhonda and John.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Truly random number generators are a crucial part of everyday life in most modern cultures. In this information age, people send emails, call or message friends and make online transactions millions of times per day. Each of these everyday processes is assumed to be safe and confidential. The security of communication depends on the ability of these processes to verify that the people communicating are actually who they say they are. Security can only be accomplished through the distribution of private identities known only by the individual user, known as keys, so that malicious entities cannot impersonate anyone and/or cause some form of harm. A private key is a large randomly generated number that is unique to the user. To establish a safe connection, a public identity, or public key, that can be shared with others is created. An example of how to establish a safe connection is illustrated by the Diffie-Hellman Key Exchange protocol in [1]. A public key is created by taking a large prime number and raising it to the power of the value of the user's private key. This creates a very large number, ensuring the original key cannot be obtained easily. The randomness of private key numbers determines how safe the actual connections and public keys are from attacks and impersonations. The ability to generate random numbers is therefore a very important part of the security of communication systems.

Most random numbers in cryptology systems are generated using a linear feedback shift register (LFSR) or a combination of LFSRs. A simple LFSR is an n-bit long shift register with a series of XOR logic gates fed back to the first register. An LFSR will output every number from 1 to $2^n - 1$, where n is the number of registers in the LFSR, and the order in which these number are outputted is determined by feedback portion. The output of an LFSR is periodic and will follow a pattern indicated by the feedback; because this is a deterministic process, it is known as a pseudo-random number generator (PRNG). Many large numbers can be accessed quickly using an LFSR, but what makes these numbers truly random is the starting position. A truly random number generator (TRNG) is used to determine this point and can be designed in a number of different manners. Within a computing environment, many natural phenomena can be used to create a TRNG, including the number of mouse clicks and their locations on the screen or the number of times a hard drive is accessed within a certain period of time. Other methods develop hardware to create this randomness. This thesis focuses on using the phase noise of a voltage controlled oscillator (VCO) to create this randomness.

1.1 Thesis Organization

This thesis consists of six chapters. Chapter 2 covers the background theory of the oscillator-based TRNG. The different components of the TRNG, including large noise VCOs are discussed in Chapter 3. Chapter 4 illustrates the results of the TRNG system level tests. The extracted design from the microchip is analyzed in

Chapter 5. A discussion of conclusions and recommended improvements is provided Chapter 6.

Chapter 2

Background

2.1 Random Number Generation

Random number generators (RNG) are used to create private keys in modern communication security systems. There are two broad types of RNGs: the TRNG, which was the type designed for this thesis, and the PRNG. The TRNG uses real world random occurrences, such as the number of times a computer hard drive is accessed, the number of mouse clicks a user makes, or the thermal noise produced by the circuits themselves, to generate a stream of completely random numbers, or bits. PSRGs are more common because they are easy to implement using an LFSR based structure which will generate a random number using a predetermined list of numbers based on the LFSR feedback function. The randomness comes from selecting a number in the stream that is some value away from the seed or initial value generated by a TRNG. LFSRs can be implemented in both hardware and software.

2.1.1 Linear Feedback Shift Register

An LFSR is simply a shift-register where the input for the next clock edge is generated from some algebraic combination of the register's current contents [1]. A simple LFSR is given in Figure 2.1. Its size is three bits and the feedback polynomial is x^3+x^2+1. This means that the third and second bits are XORed to provide the feedback to the first. This is a maximal-length feedback polynomial because it will provide the most random numbers possible for an LFSR, which is $2^n - 1$.

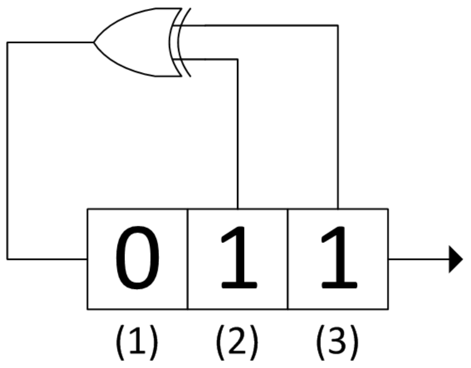


Figure 2.1: An example of a 3-bit LFSR with maximal feedback polynomial x^3+x^2+1.

The entire output sequence can be seen in Table 2.1, which shows $2^3 - 1$ distinct numbers that will repeat periodically.

Table 2.1: One period of a LFSR in Figure 2.1.

Iteration	Output
seed	110
1	100
2	001
3	010
4	101
5	011
6	111
7	110

2.1.2 Truly Random Number Generator

For the TRNG designed in this thesis, thermal noise was used to generate randomness. There are three main ways to use thermal noise to generate random bits [2, 3]. The first is to amplify the resistor thermal noise and then compare it to the DC value of the amplifier output. The final output of the comparator will be random. This design is illustrated below in Figure 2.2.

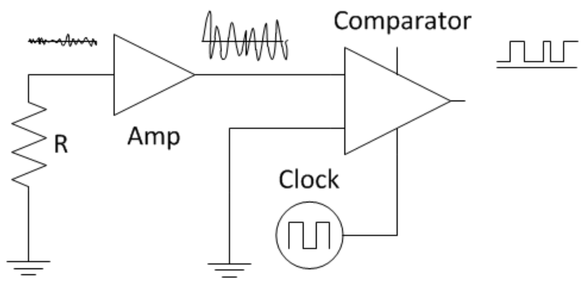


Figure 2.2: Direct amplification random number generator.

The second method for generating a random bit-stream is to use the phase noise of an oscillator to create a random noisy clock input to a delay flip-flop (DFF) that has a fast oscillating D input. If the clock is noisy enough, the rising edge of the clock is highly uncertain and the output will be random. A block diagram of a system that implements this is given in Figure 2.3. It consists of two oscillators and a DFF. One oscillator goes to the clock input while the other goes to the D input. If the D input oscillator (denoted as the fast fscillator [FO]) is fast enough compared to the clock input oscillator (denoted as the slow oscillator [SO]) such that the timing jitter (discussed in Section 2.3) of the SO is the same length of time as the period of the FO, the output bit will be equally likely to be a zero or a one. This assumes that the FO has a perfect 50% duty cycle.

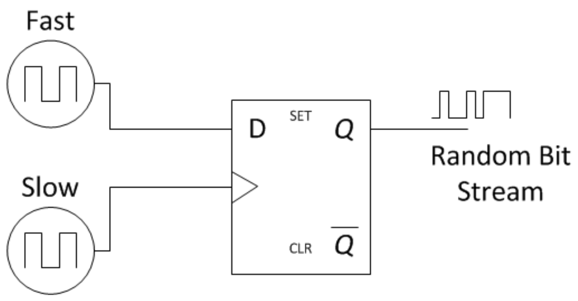


Figure 2.3: System level design of TRNG using phase noise.

An added concern when designing for randomness using the second TRNG method is whether the next value of Q can be determined from a known clock edge, given that the average frequency of both the D and clock inputs are known. This problem is illustrated in Figure 2.4, which shows the probability density function (pdf) for the clock jitter as well as the D and clock input waveforms. For a random output, the chance of the output being a one or a zero should be equal or 50% for each. Using the pdf, it is known that the total area under the curve is equal to 1,corresponding to 100% of all clock edge threshold crossings. The probability of the D input equalling 1 when the clock edge rises is $\text{P}(D = 1) = \text{P}(a < Z < b) + \text{P}(c < Z < d)$. Similar to a Z-test, $\text{P}(a < Z < b)$ and $\text{P}(c < Z < d)$ are equal to the area of the shaded regions a-b and c-d, respectively, over the whole area. From this, it can be determined that the standard deviation of the jitter should to be wide enough such that the combined sum of the shaded regions on the pdf will be equal to 0.5 or 50% of the pdf. In other words, the value D will be equally likely to be a one or a zero [4].

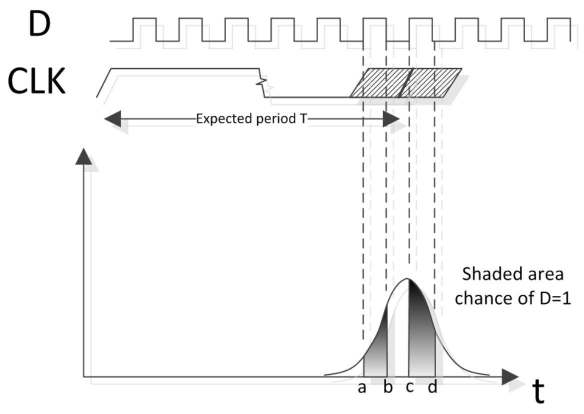


Figure 2.4: FO and SO waveforms with timing jitter PDF.

The last method to create a TRNG is to employ a metastable circuit that uses noise to push the output to one state or another. One design which is covered extensively by Intel is shown in Figure 2.5 [5].

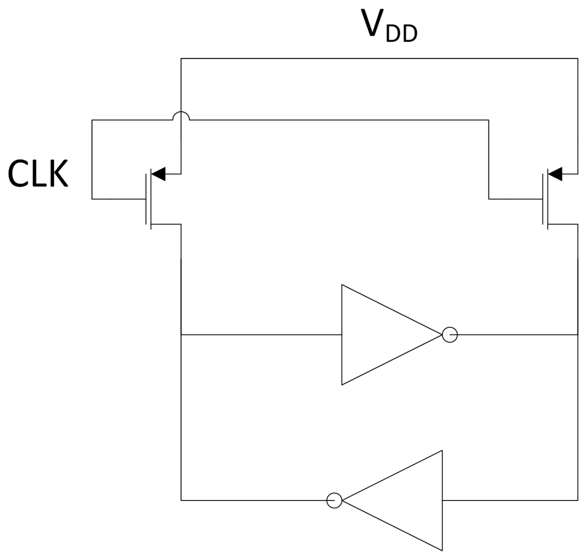


Figure 2.5: Metastability-based TRNG using two inverters.

The operation of this TRNG is simple in theory: two inverters are connected to each other's inputs. This type of configuration might usually be used as a refresher to hold the output for a dynamic latch to stop leakage, but since both are connected to V_{dd} through clock-controlled transistors, both the inputs and outputs will go high when the clock goes low. When the clock goes high and disconnects the V_{dd}, both sides force the other to lower to half V_{dd} due to both inverters acting on each other's input. This halfway point is the metastable state and the TRNG will stay here until thermal noise causes one inverter to overpower the other, forcing the output of the stronger inverter to go to zero and causing its input to swing to one. The challenging aspect of this configuration is making sure that it is highly process-voltage-temperature (PVT) variation resistant; otherwise, if the switching threshold is not identical and exactly $V_{dd}/2$, the metastable state will never be reached and the randomness of this TRNG will be ruined.

2.2 Randomness Tests

If the output bit-stream of a TRNG is predictable, such as if it always has a large percentage of ones, it would be more vulnerable to an attacker determining the seed and thus cracking the PRNG and decrypting the data. This would make for a very poor TRNG. To avoid this issue, there is a suite of tests that can be performed on a stream of bits to determine if the randomness is acceptable using statistical analysis. This package of tests was assembled by the National Institute of Standards and Technology (NIST) for application in the testing of random number generators [6, 7]. For each test, a data bit-stream (ϵ) with length n = 20,000 was used. This particular length was chosen because it was known to be achievable by the available lab testing equipment. Since the longest bit-stream possible was 20,000, certain tests in the NIST package were excluded due to lack of accuracy.

Each test generates a one-tail probability (P-value) for the null hypothesis that the bit-stream given is random. A confidence interval of 99% was used as outlined in [6]; a P-value greater than 0.01 would therefore result in a pass for that particular test. As another measure of precaution, NIST recommends that when using a 99% confidence level, 100 bit-streams of 20,000 bits be used from the number generator to verify that it is indeed random. If any number lower than 100 is tested, a lower confidence should be used. For a 99% confidence level including standard deviation, 96 of the 100 tests must pass for the number generator to be considered random. Some of the more advanced statistical functions are outlined in the NIST reference. In this section, the randomness tests used in this work are introduced.

2.2.1 Frequency Test

The purpose of the frequency test is to assess the distribution of ones and zeros in the bit-stream output. Ideally, for a random sequence, there should be the same number of ones as zeros, but that will not always be the case and the test suite outlines the acceptable error.

The procedure for the frequency test is to use Equation (2.1) to solve for the P-value;

$$P = erfc\left(\frac{|\sum 2\epsilon_i - 1|}{\sqrt{n}}\right) \tag{2.1}$$

where ϵ_i is one bit in the i^{th} position of the bit-stream and n is the length of the bit-stream. $erfc(z)$ is the complementary error function.

The frequency test is passed if there is no evidence to indicate that the tested sequence is non-random, i.e. the P-value is greater than or equal to 0.01 (or a 99% confidence level). For a bit-stream of length 20,000, the acceptable number of ones more than zeros and vice versa is 364.

2.2.2 Frequency within a Block Test

The frequency within a block test involves determining how many ones are within a block of length M bits and comparing this number to the frequency expected under the assumption of truly random input, $M/2$. The number of blocks N is defined as the length of the bit-stream n divided by the length of each block M. For these tests, n was set to 20,000 and M was set to $0.01n$, therefore M was determined to be 200 and the total number of blocks inspected N was 100. The frequency within a block test involves first calculating the proportion of ones in each block:

$$\pi_i = \frac{\sum_{j=1}^{M} \epsilon_{(i-1)M+j}}{M} \tag{2.2}$$

How close the proportions are to 50% is then determined:

$$\chi^2_{obs} = 4M \sum_{i=1}^{N} (\pi_i - \frac{1}{2})^2 \tag{2.3}$$

$$P = Q\left(\frac{N}{2}, \frac{\chi^2_{obs}}{2}\right) \tag{2.4}$$

The Q function is the complementary incomplete gamma function. The frequency within a block test is passed if the P-value is greater than or equal to 0.01.

2.2.3 Runs Test

The runs test looks for long strings of either ones or zeros that are uninterrupted. It will analyze the bit-stream to determine if the oscillation between zeros and ones is occurring too quickly (a deterministic signal that resembles a clock) or too slowly (a constant dc signal that is also deterministic). The number of switches can be determined by the following two equations:

$$V_{n(obs)} = \sum_{k=1}^{n-1} r(k) + 1 \tag{2.5}$$

$$r(k) = \begin{cases} 0 & \text{if } \epsilon_k = \epsilon_{k+1} \\ 1 & \text{otherwise} \end{cases} \tag{2.6}$$

The deciding criteria for the passing of this test can be obtained through Equation (2.7).

$$P = erfc\left(\frac{|V_{n(obs)} - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}\right) \tag{2.7}$$

where erfc is the complementary error function, $V_{n(obs)}$ is the total number of runs in the bit-stream and π is the proportion of ones in the whole stream. At a 99% confidence level, the number of switches for a 20,000 bit-stream of data was determined to lie within 9,816 and 10,180 switches.

2.2.4 Longest Run of Ones

The longest runs test looks for every longest run of ones within blocks of length M. This distribution of longest runs is then compared to the expected distribution for a random sequence. For the bit-stream length defined by the test, a block length of

128 bits was used. The frequencies of the longest runs for each block were counted and distributed into the bins outlined in Table 2.2.

Table 2.2: Long run frequency bins.

v_i	Run Length
v_0	≤ 4
v_1	5
v_2	6
v_3	7
v_4	8
v_5	≥ 9

Using these frequencies, the chi-square value was obtained:

$$\chi^2(obs) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i} \tag{2.8}$$

where K=5 and N=49 for M=128. The P-value was found with the complementary incomplete gamma function:

$$P = Q\left(\frac{K}{2}, \frac{\chi^2(obs)}{2}\right). \tag{2.9}$$

2.2.5 Discrete Fourier Transform Test

The purpose of the discrete Fourier transform (DFT) test is to convert the bit-stream into a spectral graph to determine if there are any high peaks, indicating recurring or periodic patterns.

The DFT test involves first converting all zeros in ϵ to -1. The magnitude, M, of the DFT of the new bit-stream is then calculated. The 95% threshold value, T, is determined by:

$$T = \sqrt{n \log \frac{1}{0.05}} \tag{2.10}$$

Assuming the bit-stream is random, 95% of the values in M should not exceed this value. The normalized difference between the observed and expected number of frequency components, d, is then calculated:

$$d = \frac{(N_1 - N_0)}{\sqrt{n(0.95)(0.05)/4}} \tag{2.11}$$

where $N_0 = 0.95n/2$ is the expected number of points above the value T and N_1 is the actual observed number. The P-value is found using the complementary error function:

$$P = erfc\left(\frac{|d|}{\sqrt{2}}\right). \tag{2.12}$$

2.2.6 Serial Test

Similar to the frequency test, the serial test checks the frequency of m-bit patterns and compares them to the expected number for an assumed random sequence. If m = 1, this test is identical to the frequency test.

The serial test uses three different block lengths: m, (m-1) and (m-2). Three new bit-streams are obtained for each block length by appending the first (block length - 1) bits to the end. This creates exactly n blocks for each block length. The frequencies of all overlapping m-, (m-1)- and (m-2)-blocks which are denoted as $v_{i_1 \ldots im}$, $v_{i_1 \ldots im-1}$ and $v_{i_1 \ldots im-2}$, respectively. Equations (2.13) and (2.14) are used to prepare to solve for the P-values:

$$\begin{aligned}
\Psi^2_m &= \tfrac{2^m}{n} \sum_{i_1 \ldots i_m} \left(v_{i_1 \ldots i_m - \frac{n}{2^m}}\right)^2 \\
\Psi^2_{m-1} &= \tfrac{2^{m-1}}{n} \sum_{i_1 \ldots i_{m-1}} \left(v_{i_1 \ldots i_{m-1} - \frac{n}{2^{m-1}}}\right)^2 \\
\Psi^2_{m-2} &= \tfrac{2^{m-2}}{n} \sum_{i_1 \ldots i_{m-2}} \left(v_{i_1 \ldots i_{m-2} - \frac{n}{2^{m-2}}}\right)^2
\end{aligned} \tag{2.13}$$

$$\begin{aligned}
\bigtriangledown \Psi^2_m &= \Psi^2_m - \Psi^2_{m-1} \\
\bigtriangledown^2 \Psi^2_m &= \Psi^2_m - 2\Psi^2_{m-1} + \Psi^2_{m-2}.
\end{aligned} \tag{2.14}$$

Both P-values from Equation (2.15) must be greater than 0.01 to pass this test.

$$\begin{aligned}
P_1 &= Q(2^{m-2}, \bigtriangledown \Psi^2_m) \\
P_2 &= Q(2^{m-3}, \bigtriangledown^2 \Psi^2_m).
\end{aligned} \tag{2.15}$$

Q is the complementary incomplete gamma function.

2.2.7 Approximate Entropy

The approximate entropy test entails counting the frequency of m and (m+1)-bit strings and comparing these results against the expected frequency from a random sequence. Firstly, for the m-bit block length, the bit-stream is appended by the first m-1 bits in that stream such that there are exactly n overlapping m-bit blocks. The frequency of each m-bit number that occurs is counted from all n blocks and is represented as $\#i$, where i is the decimal number from 0 to $2^m - 1$. The ratio of each number compared to n is determined by: $C^m_i = \frac{\#i}{n}$.

$$\Phi^{(m)} = \sum_{i=o}^{2^m-1} \pi_i \log \pi_i \tag{2.16}$$

where $\pi_i = C_i^m$. This is then repeated for $m+1$ to find $\Phi^{(m+1)}$, the χ^2 test in Equation (2.17) is used to compare the observed values to the expected values for randomness:

$$\chi^2 = 2n[log2 - (\Phi^{(m)} - \Phi^{(m+1)})]. \tag{2.17}$$

The P-value is found using the complementary incomplete gamma function:

$$P - value = Q\left(2^{m-1}, \frac{\chi^2}{2}\right). \tag{2.18}$$

2.2.8 Cumulative Summation Test

The purpose of this test is to determine if the random walks starting from both ends of the bit-stream deviate form the average too quickly. The test is enacted by taking the sums of successively larger subsequences from the bit-stream starting from one side. The test statistic z is the maximum value in the set of sums. The P-value is found with the following equation:

$$\begin{aligned} P - value = 1 - \sum_{k=\left(\frac{-n}{z}+1\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left[\Phi\left(\frac{z(4k+1)}{\sqrt{n}}\right) - \Phi\left(\frac{z(4k-1)}{\sqrt{n}}\right)\right] + \\ \sum_{k=\left(\frac{-n}{z}-3\right)/4}^{\left(\frac{n}{z}-1\right)/4} \left[\Phi\left(\frac{z(4k+3)}{\sqrt{n}}\right) - \Phi\left(\frac{z(4k+1)}{\sqrt{n}}\right)\right] \end{aligned} \tag{2.19}$$

where Φ is the normal cumulative distribution function.

2.2.9 Poker Test

The poker test is no longer a part of the NIST suite, although it is similar to the approximate entropy test. It is used in [2] and provides a graphical representation of the randomness of the stream by plotting the frequency of every non-overlapping 4-bit binary number i in a bar graph. The desired output of this test is for each column in the bar graph to have the same height, indicating that each number is equally likely to occur. If the output displays primarily decimal zeros (0000) or fifteens (1111), it can be inferred that there is a dominant amount of zero or one runs, respectively, in the bit-stream. Alternatively, a large number of fives (0101) and tens (1010) would indicate a deterministic clock-like signal.

2.3 Definition of Phase Noise and Timing Jitter

Phase noise is the frequency domain representation of random changes in the frequency of the carrier signal. It is defined as the ratio of power at a chosen sideband frequency to the power of the carrier. Single-sideband phase noise is calculated using Equation (2.20) as described in [8]:

$$L(f_m) = 10log\left(\frac{P_{sideband}(f_c + f_m, 1Hz)}{P_{carrier}}\right) \tag{2.20}$$

where $P_{sideband}$ is the power of the sideband frequencies, f_c is the carrier frequency or oscillating frequency of an ideal oscillator, f_m is the frequency offset from the carrier to the sideband, and $P_{carrier}$ is the power of the ideal oscillator signal. Phase noise is measured in dBc/Hz; dBc refers to decibels relative to the carrier, or more simply, how many decibels lower the sideband power is than the carrier.

Frequency spectrum plots of (a) an ideal oscillator and (b) a noisy oscillator are shown in Figure 2.6. The ideal oscillator contains only one tone exactly the frequency of oscillation. In reality, noise can alter the period of oscillation creating other frequencies centred on the carrier. These random frequencies form what is shown as a bell curve in Figure 2.6(b).

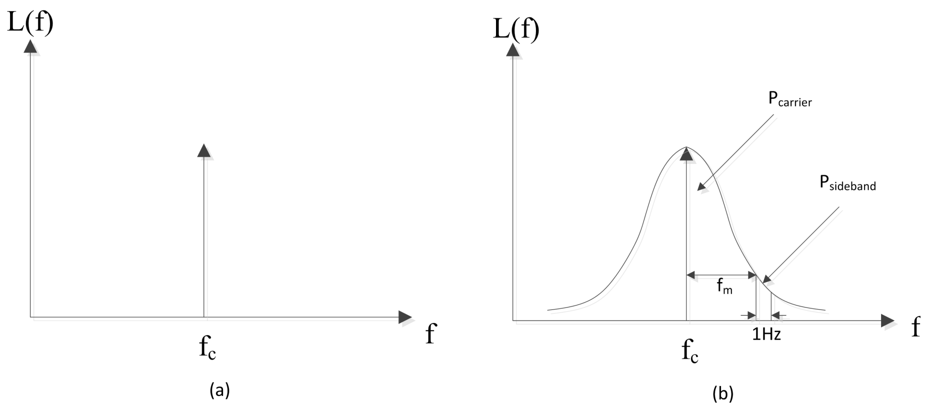


Figure 2.6: Frequency spectrum plots for (a) an ideal periodic signal with frequency f_c and (b) periodic signal with phase noise.

Phase noise of an oscillator can be described below using a Lorentzian spectrum:

$$L(f_m) = 10log\left(\frac{1}{\pi}\frac{\pi f_c^2 c}{f_m^2 + (\pi f_c^2 c)^2}\right) \tag{2.21}$$

where c is a scalar constant that defines the shape of the phase noise. Equation (2.21) can be simplified if $f_m f_c^2 c$ to Equation (2.22)

$$L(f_m) = 10\log\left(\frac{f_c^2 c}{f_m^2}\right) \tag{2.22}$$

A relationship can be formed between phase noise and cycle-to-cycle jitter in the following equation:

$$L(f_m) = 10\log\left(\frac{\sigma_c^2 f_c^3}{f_m^2}\right) \tag{2.23}$$

where σ_c is the timing jitter. The previous equations for phase noise assume that the noise source is completely white, meaning flicker (1/f) noise was ignored.

Timing jitter is the measurement of the noise from an oscillator in the time domain. There are two main components of timing jitter: random jitter and deterministic jitter. Only random jitter was considered in this thesis. Jitter is the random deviation in the period length of a periodic signal. Random jitter can be broken down further into cycle-to-cycle jitter and absolute jitter. Cycle-to-cycle jitter, denoted by σ_c, is the threshold crossing deviation after one period of oscillation; an example of cycle-to-cycle jitter is shown in Figure 2.7.

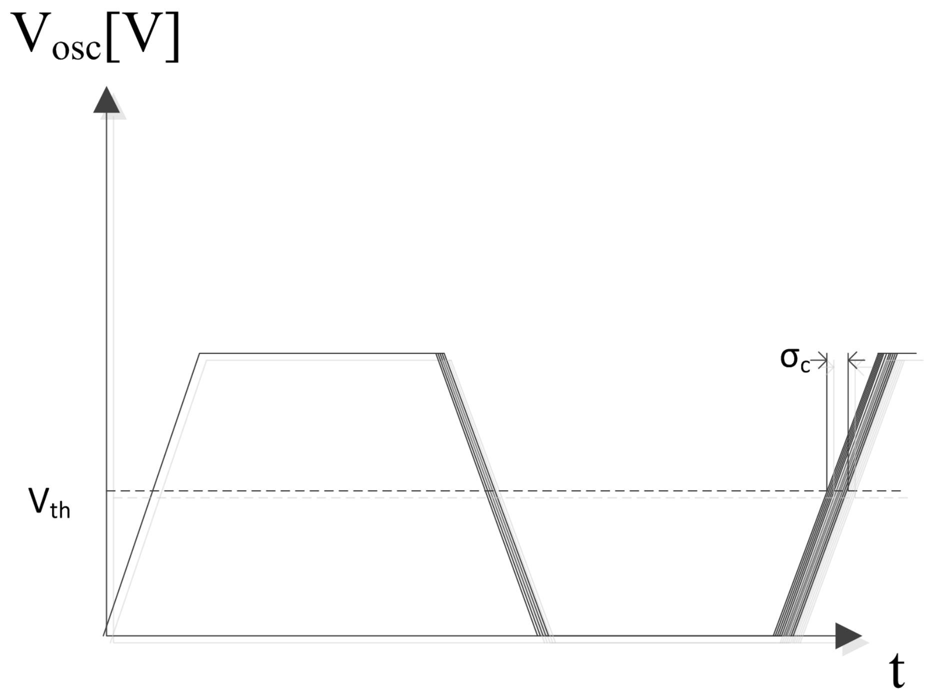


Figure 2.7: One period of oscillation with jitter included.

Absolute jitter is the accumulation of cycle-to-cycle jitter, and therefore depends on the number of cycles observed. Absolute jitter can be defined as:

$$\sigma_{abs}(t = N\tau_{avg}) = \sum_{n=1}^{N} \tau_n - \tau_{avg} \tag{2.24}$$

where N is the number of cycles, $\sigma_{abs}(t = N\tau_{avg})$ is the absolute jitter after N cycles, τ_{avg} is the average period of oscillation and τ_n is the actual period for a specific cycle. Absolute jitter only becomes a problem when using a free-running oscillator, which is an oscillator whose frequency is not corrected with negative feedback, such as is the case with a phase-locked loop. In a free-running oscillator, it does not matter when the threshold is crossed; it will continue as if nothing has changed. For an illustration of absolute jitter, refer to Figure 2.8.

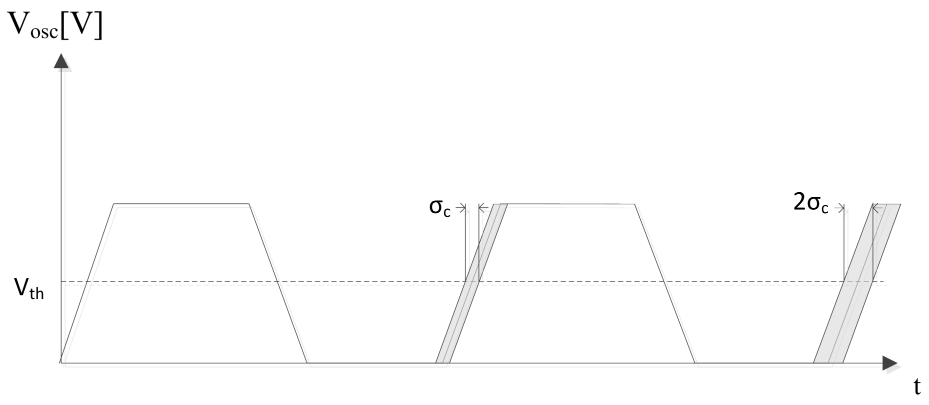


Figure 2.8: Time domain plot of absolute jitter.

The equation for cycle-to-cycle jitter can be obtained using the absolute jitter and making sure enough samples (cycles) are taken.

$$\sigma_c^2 = \lim_{N\to\infty}\left(\frac{1}{N}\sum_{n=1}^{N}(\tau_n - \tau_{avg})^2\right) \tag{2.25}$$

2.4 Phase and Jitter Models for Ring Oscillators

2.4.1 First Passage Time

Jitter can be approximated using the first passage time (FPT) method covered in Abidi [9]. This method uses the noise current that integrates over a load capacitance looking at a single delay cell for a ring oscillator. For the first simple case, a two transistor digital CMOS inverter was used. This method is known as FPT because the jitter is measured from the first point that the actual voltage waveform crosses the threshold level to the expected point that the waveform will cross. Refer to Figure 2.9 for an example of FPT.

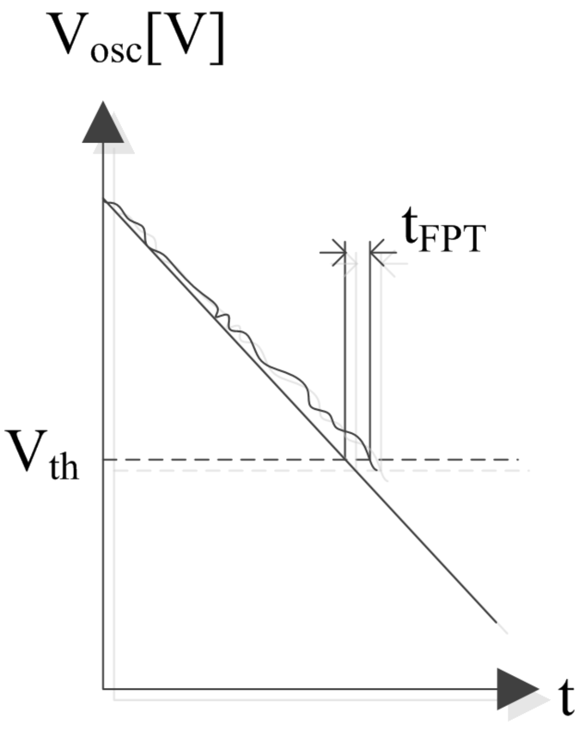


Figure 2.9: Threshold crossing plot for FPT

The variance of the time deviation at the threshold crossing of the inverter output is given in Equation (2.26):

$$\sigma_c^2 = \frac{\overline{v_n^2}}{\left(\frac{I}{C}\right)^2} \tag{2.26}$$

where $\overline{v_n^2}$ is the noise voltage on the output capacior and $\left(\frac{I}{C}\right)^2$ is the slew rate of the output squared. The noise voltage on the load capacitance is simply the noise current from the MOS transistors divided by the capacitance; this equation is discussed in the paper by Leung [10]

$$\bar{v_n^2} = \frac{t_d\overline{i_n^2}}{C^2} = \frac{t_d 4kT\frac{2}{3}g_m}{C^2} \tag{2.27}$$

where t_d is the time to reach the switching threshold, g_m is the transconductance of the transistor, k is the Boltzmann constant, T is the temperature in Kelvin, C is the capacitance of the load, $\overline{i_n^2}$ is the rms noise current and $\overline{v_n^2}$ is the rms noise voltage. Equation (2.27) shows the current noise of a MOS transistor in saturation [11]. Only one noise source is used to simplify the problem displayed in Figure 2.10; the transistors M_1 and M_2 and the capacitor C are considered noiseless. Only the noise current source $\bar{i_{np}^2}$ is considered since the g_m of M_2 will be considerably smaller as it will have been turned off, making $\bar{i_{n2}^2}$ much smaller.

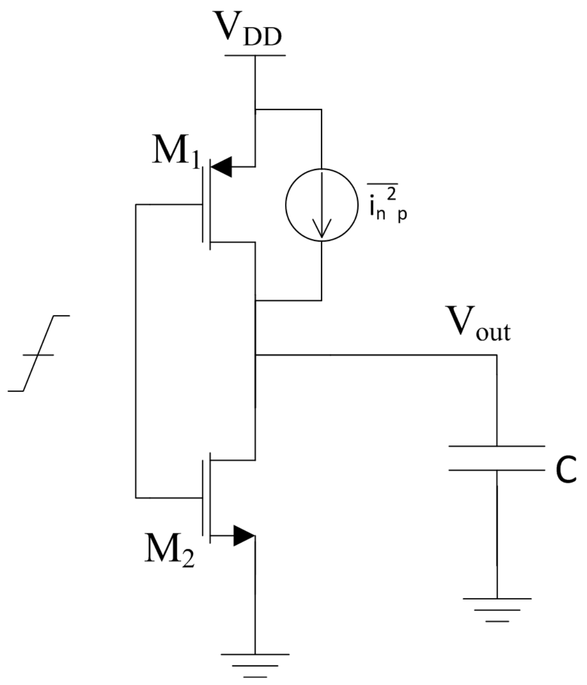


Figure 2.10: Schematic of a simple inverter delay-cell with noise current.

Once the jitter has been acquired for one stage and one rise or fall, the following equation can be used to calculate the total FPT jitter of a ring oscillator:

$$\sigma_{FPT} = \sqrt{2N \times \sigma_c} \tag{2.28}$$

where N is the number of stages in the ring oscillator. The factor of 2 comes from the fact that, while the jitter calculated in Equation (2.27) was for only one edge, the PMOS and NMOS are assumed to generate the same noise current and therefore the jitter from both the rise and fall times are equal.

2.4.2 Last Passage Time

Another consideration with respect to jitter is last passage time (LPT). The difference between FPT and LPT is that LPT assumes that the actual waveform crosses the threshold level many times (as opposed to just once), thus increasing the jitter for that crossing. An exaggerated example of one crossing showing LPT is given in Figure 2.11.

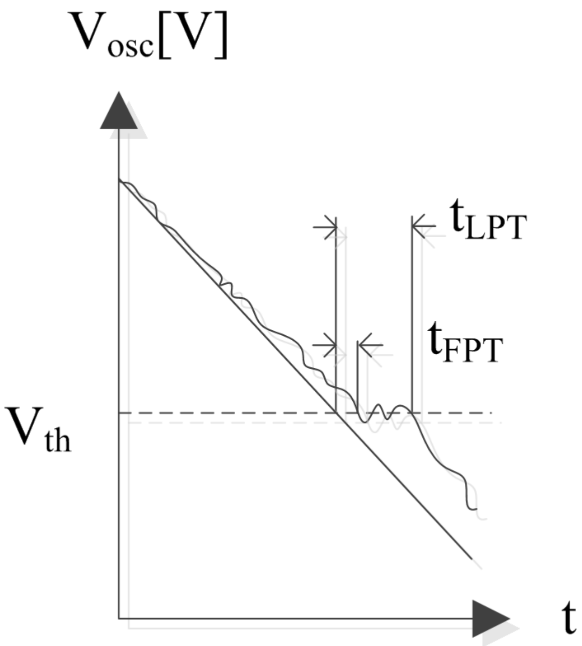


Figure 2.11: Threshold crossing plot for LPT

The analysis of LPT provided by Leung [10] is complex; there is no closed-form solution for jitter using Leungs LPT calculations, but the cumulative distribution function (CDF) is described by Equation (23) in that paper. From this equation, it can been seen that the important factors for LPT are the slew rate at the threshold voltage, the threshold voltage, and the time that it takes to cross the threshold. A simplified closed-form solution was later devised by Leung in [12]:

$$\sigma_{LPT} = \sqrt{2\tilde{\sigma_c}^4 + \theta\tilde{\sigma_c}^2} \tag{2.29}$$

where σ_{LPT} is the LPT for one stage and one edge, θ is the time to reach the barrier or voltage threshold, and $\tilde{\sigma_c}$ is the total current noise divided by the load capacitance and slew rate for one stage and one edge. From Equation (2.29) it can be seen that the total jitter is a combination of the FPT variance $\theta\tilde{\sigma_c}^2$ and a new term $\tilde{\sigma_c}^4$ which demonstrates that LPT can be much greater than FPT because of the term raised to the fourth power.

From both models, it becomes apparent that a low slew rate is the key to increasing noise in a ring oscillator. The trade-off is the frequency of the oscillator, since more noise is introduced as the speed is reduced.

2.5 Impact of Phase Noise on Random Number Generators

The more noise the SO can produce, the slower the FO needs to be to still perform at the required levels. This is important because the FO frequency is upper bounded by the fabrication technology. The speed at which the seed can be delivered is determined by the SO which is required to recover the DFF output signal. The desired waveform would therefore need to be fast enough to achieve the speed requirements of the TRNG but also have a relatively low slew rate at the threshold level to increase timing jitter and improve random number generation results. The approach covered in Section 3.3 seeks to accomplish these tasks.

Chapter 3

Truly Random Number Generator

In this chapter, individual components of the TRNG are designed and tested. The Cadence software was used to produce simulations using the IBM 0.13μm technology provided by Canadian Microelectronics Corporation (CMC). Timing jitter for the SO was calculated using the noisetran function in the Eldo software [13]. One period was run multiple times to obtain the threshold crossing distribution. Timing jitter is the standard deviation of the normal distribution.

3.1 Fast Ring Oscillator Design

For the D input of the DFF a specifically fast oscillator was required. The oscillator was required to be sufficiently fast so as to have one period of oscillation contained within the timing jitter of the clock input to the DFF, as was illustrated in Figure 2.4. This ensured that if at any time the FO had a 50% duty-cycle, the output would have had an equally likely chance of a one or a zero.

The easiest way to achieve the FO requirements was to implement a 3-stage simple inverter ring oscillator(RO). A RO is able to achieve fast speeds, as well as having a saturated output making the design of the DFF simpler. An odd number of stages is required to allow for oscillation since output is single ended and needs to be inverted. The minimum number of stages, three, was chosen to minimize the delay. One stage of a ring oscillator provides one unit of delay and is usually denoted as the delay cell.

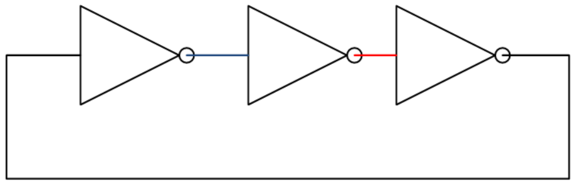


Figure 3.1: A simple 3-stage ring oscillator

The frequency of the simple ring oscillator is determined from the delay of each stage [14]. A time domain graph of the three node voltages from Figure 3.1 is displayed in Figure 3.2. The equation used to calculate frequency of a simple ring oscillator is as follows:

$$f_o = \frac{1}{2Nt_p} \tag{3.1}$$

where N is the number of stages and t_p is the propagation delay of one cell. t_p can be replaced with 69% of the inverter's time constant shown in Equation (3.2) using R as the equivalent resistance of the 'on' transistor in one of the inverters, and C, the total capacitance at the node.

$$f_o = \frac{1}{2N \times 0.69RC} \tag{3.2}$$

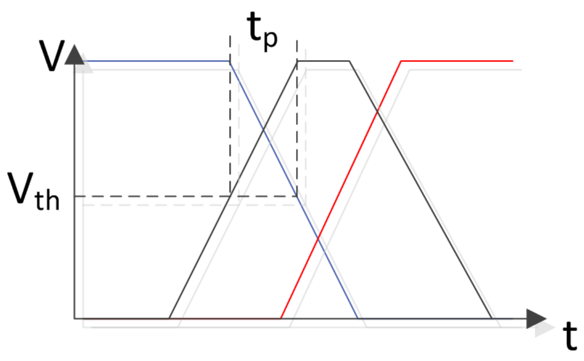


Figure 3.2: Transient graph of the ring oscillator from Figure 3.1.

3.1.1 Transistor Level Simulation

The FO was designed to be as fast as a saturated ring oscillator can be. Since the simple inverter is single-ended, a minimum of three stages were needed to obtain the feedback inversion for the ring oscillator to oscillate. The strength of each delay cell was increased until the frequency gains levelled off due to increased capacitive load. The supply voltage was set to 1.2V, the recommend voltage level for the 0.13μm IBM CMOS technology, but could also be raised to increase speed if necessary. One of the three delay cells is shown in Figure 3.3; sizes were chosen to ensure adequate trade-off between driving power and load capacitance.

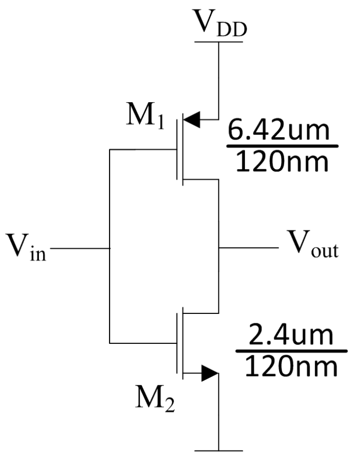


Figure 3.3: Delay cell for a fast 3-stage ring oscillator

The final design output waveform is given in Figure 3.4. The output is almost sinusoidal and has a frequency of 9.5GHz and a duty cycle of 50%.

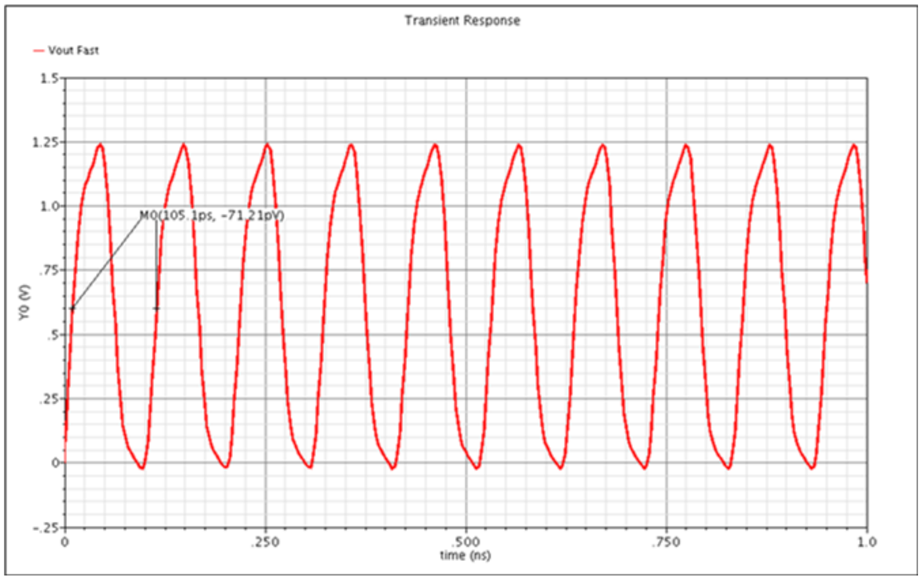


Figure 3.4: Transient simulation of the simple inverter ring oscillator. Frequency = 9.51GHz.

Since this ring oscillator and its slew rate are so fast, noise was considered negligible and ignored during the system level simulations.

3.2 Design 1 - Current-Starved Voltage Controlled Oscillator

The current-starved VCO is a versatile oscillator that allows control over both the rise and fall delays of the inverter by adjusting the bias voltages of the top and bottom transistors. A single delay cell is shown in Figure 3.5. All top and bottom transistors for the VCO are controlled by a current mirror with external control of the resistor values. This control allows for easy adjustment of the slew rate of each delay cell, which affects the jitter.

3.2.1 Transistor Level Simulation

A nine-stage VCO was created using the delay cell in Figure 3.5. Both current mirrors were fixed to supply $150\mu A$ in order to create a 50% duty cycle clock signal. One period of the current starved VCO output is shown in Figure 3.6. The frequency of operation was approximately 75MHz.

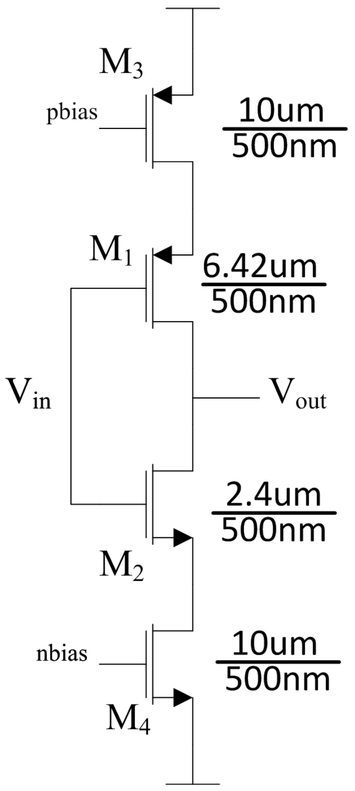


Figure 3.5: Transistor level schematic of one delay cell for a current-starved inverter VCO.

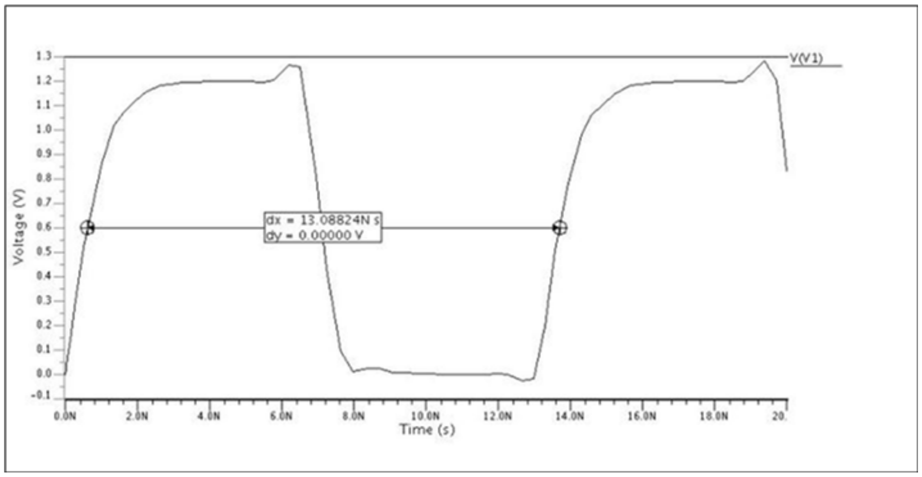


Figure 3.6: Eldo transient simulation of current-starved ring oscillator. Frequency=76.4MHz

The output capacitance of each delay cell was found to be approximately 35fF. Using this capacitance and the slew rates found at the switching threshold of the rising and falling edges of Figure 3.6, the jitter was estimated using FPT:

$$\overline{v_n^2} = \frac{t_d \overline{i_n^2}}{C_L^2} \tag{3.3}$$

$$\sigma_{tot} = \sqrt{\frac{n}{2}} \sqrt{\frac{\overline{v_{nrise}^2}}{SR_{rise}^2} + \frac{\overline{v_{nfall}^2}}{SR_{fall}^2}} \tag{3.4}$$

Table 3.1: Jitter calculation for current-starved VCO

	Falling	Rising
$\overline{i^2_{ntot}}$	2.03×10^{-24}	2.98×10^{-24}
$\overline{t_d}$	600ps	542ps
$\overline{v^2_{ntot}}$	9.94×10^{-7}	1.32×10^{-6}
Slew Rate	7.8×10^{8}	9.53×10^{8}
σ^2	1.63×10^{-24}	1.45×10^{-24}
Total Jitter(N=9)	3.73ps	

Using the jitter obtained from Equation (3.4), and Table 3.1 the number of samples for a noise run was obtained. Equation 3.5 from [15] was used to obtain a sample size that would provide a 95% confidence with an error(E) of $\pm$ of 0.5ps with a standard deviation or jitter of 4ps:

$$n = \left(\frac{z_{\alpha/2}\sigma}{E}\right)^2 \tag{3.5}$$

An n of approximately 250 was obtained; this value was used in the Eldo noisetran simulation below.

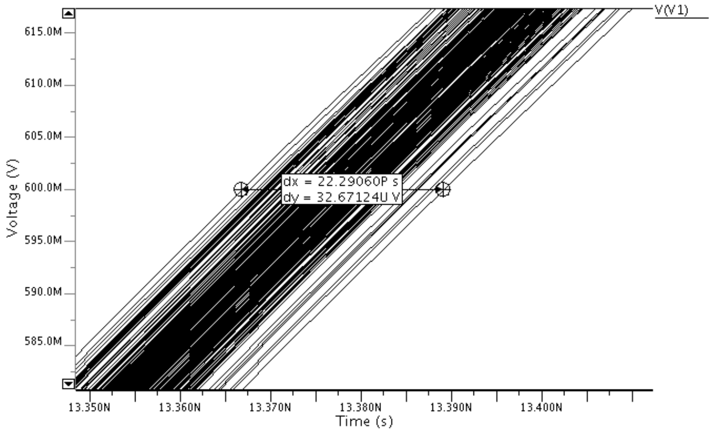


Figure 3.7: Zoomed-in view of the threshold crossing spread after one period of the current-starved VCO with 250 noise runs.

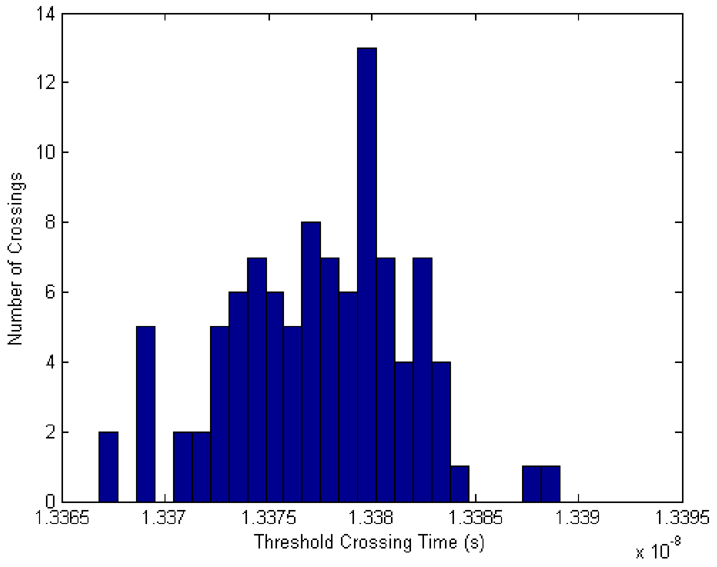


Figure 3.8: Threshold crossing histogram of Figure 3.7 at 0.6V.

The standard deviation, and hence the jitter, of the threshold crossing histogram in

Figure 3.8 was calculated to be 4.32ps.

3.3 Design 2 - Current-Stealing VCO

In general, in order to increase the amount of noise in a VCO, the slew rate must be decreased at the threshold. Decreasing the slew rate in turn makes the VCO slower. It was desired to make a faster VCO since it was going to be used as a clock input to the DFF. This same clock signal will be used to recover the noisy output bits. The speed of the noisy VCO was therefore the same speed at which the RNG seed was delivered. A trade off was usually required between increasing the speed of system and increasing the randomness of the ring oscillator based TRNG.

One way to alleviate the issue of low slew rate/fast VCO is to create a fast clock has a low slew rate only as it passes the switching threshold. This was achieved using switch controlled current-stealing. Essentially, as a delay cell charges or discharges the capacitive load at the output, a switch triggers a mechanism to steal away that charging current from the delay cell. Less charging current results in a decreased slew rate thus fulfilling the goal of the circuit.

A system level design of one delay cell is given in Figure 3.9. The switch S1 controls when I_{STEAL} turns on and is itself controlled by the two circuits. The first circuit is the rising edge control path and controls the precise moment at which S1 is triggered on. The second path, called the falling edge control path governs the transmission gate which in turn controls when S1 is turned off and the low slew phase ends.

The equations for Design 2 FPT jitter are similar to those in Design 1, but because of the low-slewing phase of Design 2, LPT has a greater impact on overall timing jitter. The equations for LPT for a current-stealing oscillator are covered in detail in the technical report from Leung [12]

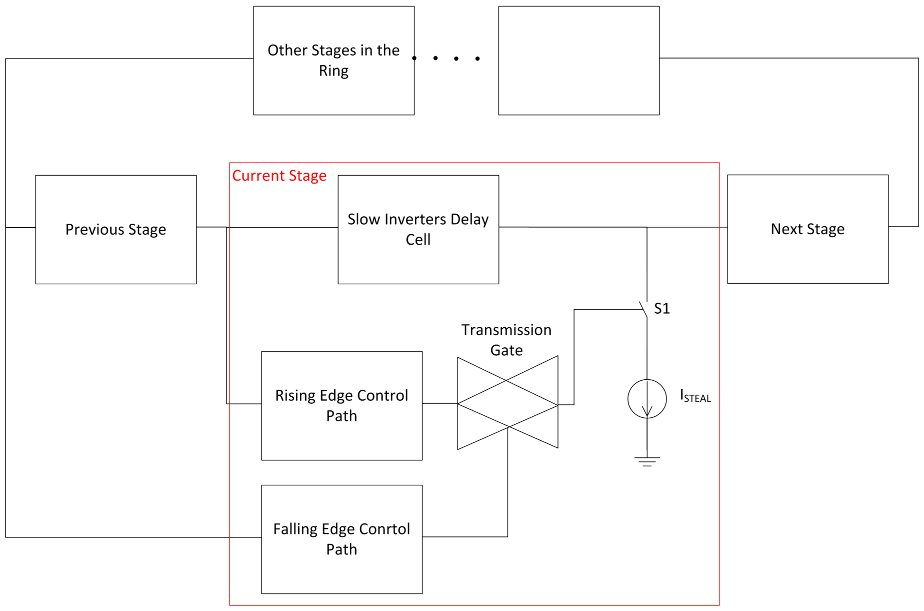


Figure 3.9: System design of the current stealing delay cell

3.3.1 Jitter Calculation

The output capacitance of each delay cell was found to be approximately 240fF. Using this value and the slew rates found at the switching threshold of the rising and falling edges of Figure 3.11, the jitter was estimated using FPT.

Table 3.2: Jitter calculation for current-stealing VCO

	Falling	Rising
$\overline{i_{ntot}^2}$	7.45×10^{-24}	3.8×10^{-24}
$\overline{t_d}$	750ps	700ps
$\overline{v_{ntot}^2}$	9.46×10^{-7}	1.32×10^{-6}
Slew Rate	6.09×10^{8}	2.58×10^{8}
σ^2	2.62×10^{-25}	7.08×10^{-24}
Total Jitter(N=7)	5.29ps	

Using equation (3.5) and the results from Table 3.2, n was determined to be 250 while the error was approximately the same as in Design 1 at 0.6ps.

3.3.2 Transistor Level Simulation

A transistor level schematic of the system level design is illustrated in Figure 3.10. The main path consists of the primary delay cell, M_1 and M_2, and the stealing-transistor M_3. This path behaves similar to a regular delay cell in a VCO but with the added control of the stealing-transistor. The stealing-transistor is governed by the control circuitry which consists of the rising and falling edge control paths. The falling edge control path uses the previous signal of the VCO to correctly time the opening and closing of the transmission gates to denote how long the low-slew phase will be active. The rising edge control path is a delay path of the input signal to the stealing-transistor. The rising edge control path was designed such that it was moderately faster than the main path delay, thus ensuring the signal V_c would go high before V_{out}, thereby turning on the stealing transistor and activating the low-slew phase around the switching threshold. Only half of each of the transmission gates are present in Figure 3.10 because they are only concerned with passing one level.

The PMOS transmission gate M_{10} is used to pass a "1" through to the stealing-transistor, and since PMOS can pass a one without the V_{th} decrease, the NMOS of the transmission gate is not needed. The same applies for the NMOS transmission gate, since it only passes a 0 which an NMOS can accomplish alone [16]. The sizes of the current-stealing delay-cell are given in Table 3.3.

For the simulations, the voltage supply was set to the recommended value of 1.2V and the simulation was run for 20ns.

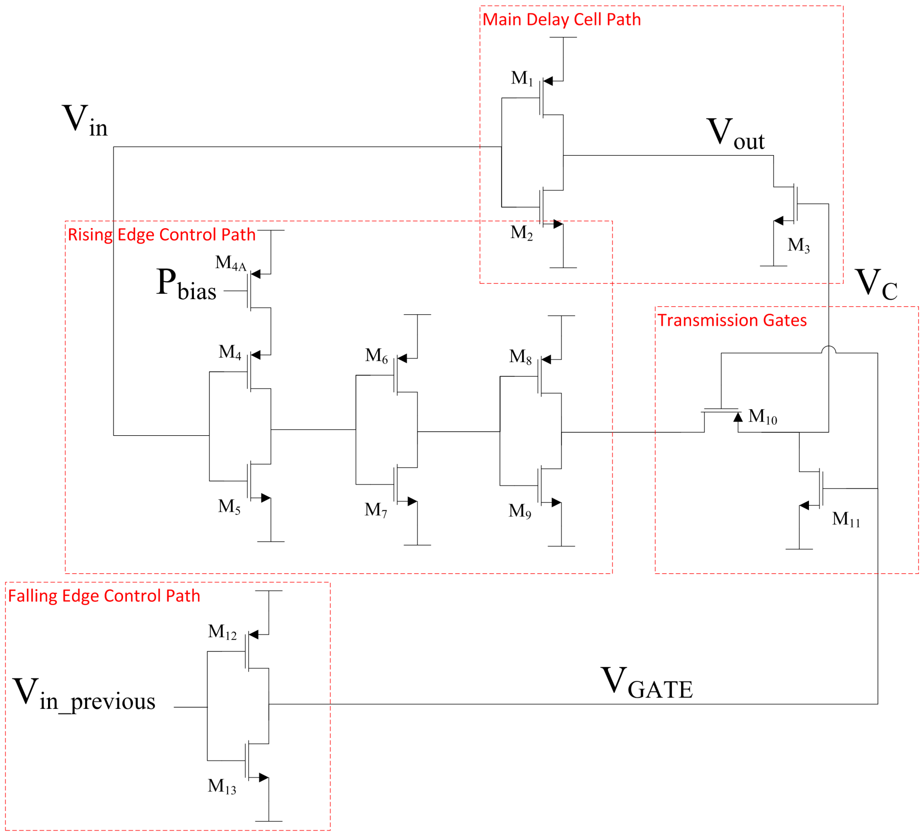


Figure 3.10: Transistor level schematic for one current stealing delay cell

Table 3.3: Transistor sizing chart for Design 2 delay cell.

Main Delay Path		
Main Inverter	M1	50um/0.6um
	M2	4.96um/0.6um
Stealing Transistor	M3	7.48um/0.6um
Rising Edge Control Path		
First Inverter(starved)	M4A	10um/0.12um
	M4	3.84um/0.12um
	M5	1.44um/0.12um
Second Inverter	M6	1.28um/0.12um
	M7	0.48um/0.12um
Third Inverter	M8	3.84um/0.12um
	M9	1.44um/0.12um
PMOS Transmission	M10	4um/0.12um
NMOS Transmission	M11	1um/0.12um
Falling Edge Control Path		
Gate Inverter	M12	1.28um/0.12um
	M13	0.48um/0.12um

A simulation frequency of 60MHz was achieved for the complete ring oscillator. The operation of the current-stealing VCO is further explained in the Figure 3.11. Figure 3.11(b) clearly shows that the gate signal is the inversion of the input from the previous stage, and that V_{gate} creates a window for the control signal to pass through. Figure 3.11(c) shows that the waveform has a low-slew phase at around 0.8V controlled by the signal V_c, this voltage was targeted to be the threshold value for the main inverter.

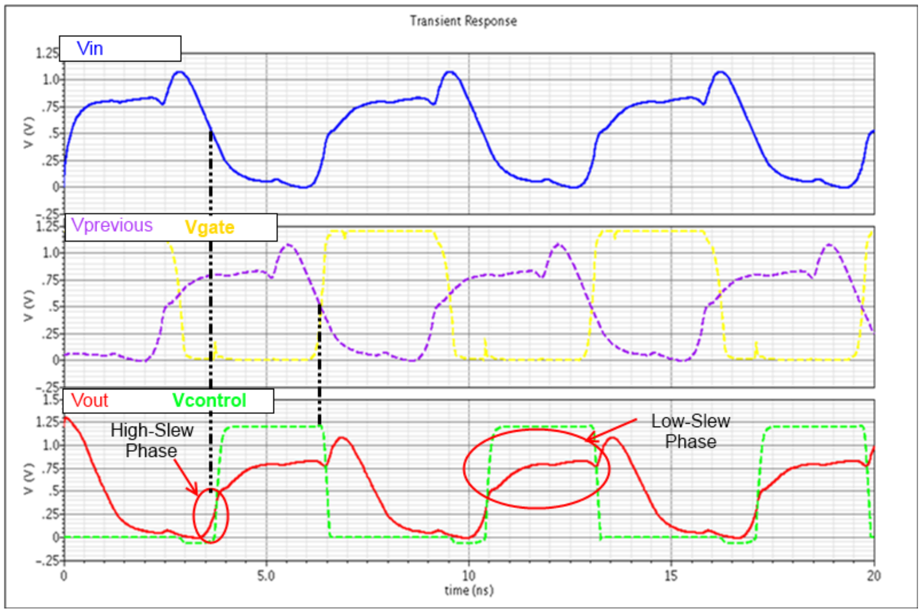


Figure 3.11: Transient operation a current-stealing VCO.

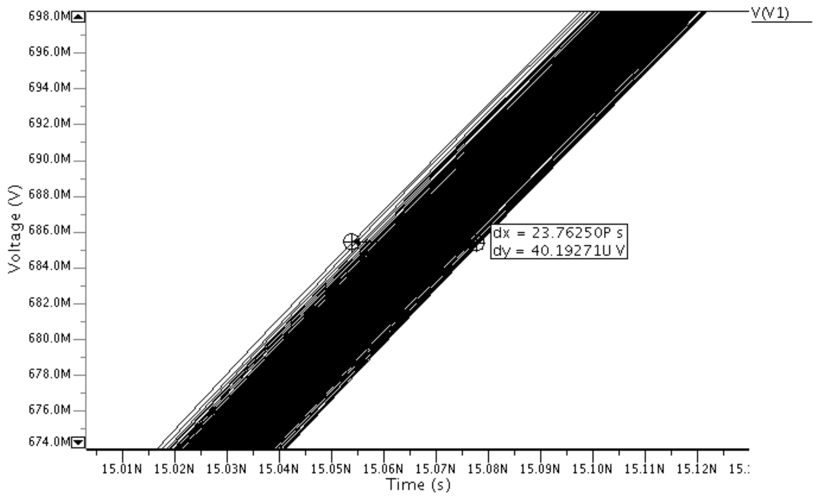


Figure 3.12: Zoomed-in view of the threshold crossing spread after one period of the current-stealing VCO with 250 noise runs.

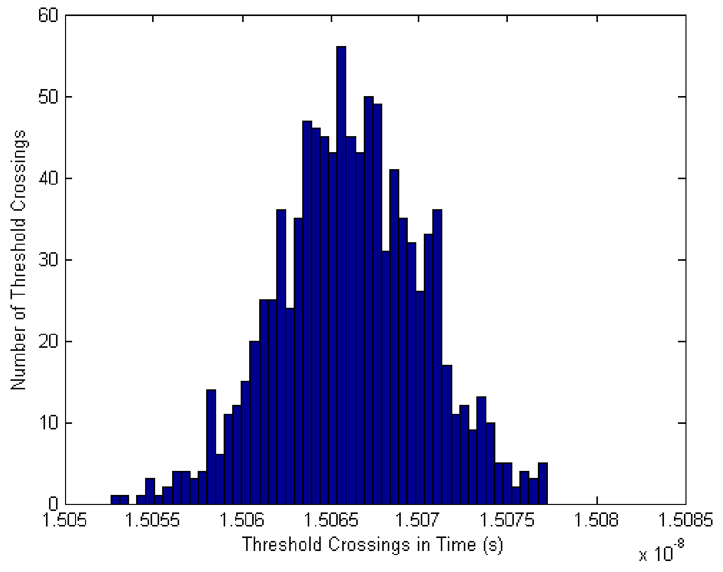


Figure 3.13: Threshold crossing histogram of Figure 3.12 at 0.685V

The standard deviation, and hence the jitter, of the threshold crossing histogram in Figure 3.13 was calculated to be 4.13ps.

A high switching threshold at the level of the low-slew phase was desired. Increasing an inverting switching threshold can be achieved by either increasing the strength of the PMOS or decreasing the strength of the NMOS. Altering the strength of a transistor can be accomplished by number of different methods. The first and simplest method for a full-custom design is to vary the size ratio (W/L) of the transistor. This changes the equivalent on resistance of the transistor and thereby alters the charging current. When the strength of the PMOS transistor in an inverter is increased, a higher input voltage is needed to turn the PMOS off and allow the inverter output to ground. Designing the main delay path inverter in the current-stealing delay cell to achieve a high threshold voltage proved to be problematic, nevertheless a solution is proposed in section 3.4. The problem stemmed from increasing the W of M_1 to increase its strength. The capacitive load of the previous stage increased as W increased based on the equation for capacitance of the gate given below:

$$C_{gs1} = \frac{2}{3} W L C_{ox}. \tag{3.6}$$

The increase in capacitance load affected the speed and timing of each stage and made achieving the desired results difficult, resulting in the inability of the design to oscillate.

3.4 Design 3 - Current-Stealing VCO with Modifications

A simple solution to the problem discussed in Section 3.3 was to insert simple inverters with higher thresholds in between two current-stealing stages and allow the current-stealing main delay path inverter to obtain a balanced size ratio.This permitted more control over the threshold value This new VCO is illustrated in Figure 3.14. The Design 3 VCO was be able to produce more jitter than Design 2 because it fully took advantage of the low slew rate portion of the current-stealing cell waveform.

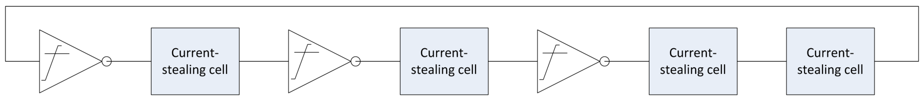


Figure 3.14: Block diagram of Design 3.

The final modification to Design 2 aimed to add additional noise to the current-stealing delay cell without altering the slew rate. This was accomplished by connecting the drain of two transistors, an NMOS and a PMOS, to the output of the current-stealing stage. The transistor level schematic of the Design 3 delay cell is shown in Figure 3.15. The two transistors were controlled with a current mirror which forced an equal current so that, when performing KCL at the output node, no additional current was allowed to enter or leave the output capacitance, assuming no channel length modulation. Since no current was added or removed the slew rate remained unaffected. The total noise of the cell, however, did increase since noise is additive. The drain capacitance C_{db} was much smaller than the gate capacitances of the following stage, hence the total load capacitance was not be altered significantly. The jitter increased with the g_m of these two new transistors. Since the on/off status of the new transistors was controlled by the output node voltage, Design 3 was slightly more complicated due to additional changes in current at specific times in the output node. The current mirrors for these extra noise sources were set to draw $20\mu A$ of current. The sizings for the transistors were similar to Design 2 with a few changes, the sizes can be viewed in Table 3.4.

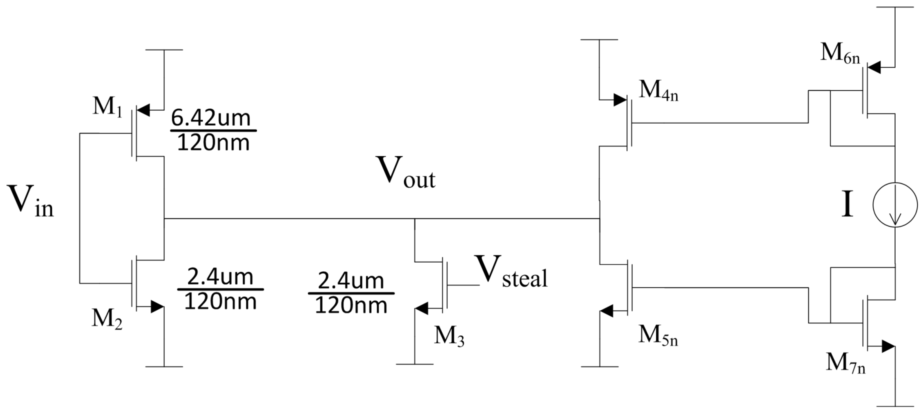


Figure 3.15: Transistor level schematic for the main path of the Design 3 delay cell

Table 3.4: Transistor sizing chart for Design 3 delay cells

Main Delay Path		
Main Inverter	M1	13.375um/0.6um
	M2	4.96um/0.6um
Stealing Transistor	M3	1.7um/0.6um
Noise Transistors	M4n	50um/0.6um
	M5n	50um/0.6um
	M6n	50um/0.6um
	M7n	50um/0.6um
Rising Edge Control Path		
First Inverter(starved)	M4A	10um/0.12um
	M4	3.84um/0.12um
	M5	1.44um/0.12um
Second Inverter	M6	1.28um/0.12um
	M7	0.48um/0.12um
Third Inverter	M8	3.84um/0.12um
	M9	1.44um/0.12um
PMOS Transmission	M10	4um/0.12um
NMOS Transmission	M11	1um/0.12um
Falling Edge Control Path		
Gate Inverter	M12	1.28um/0.12um
	M13	0.48um/0.12um
High-Threshold Delay-Cell		
Shift Inverter	M1b	8um/0.12um
	M2b	0.5um/0.12um

One period of the Design 3 output is given in Figure 3.16 and shows an oscillation frequency of 37MHz.

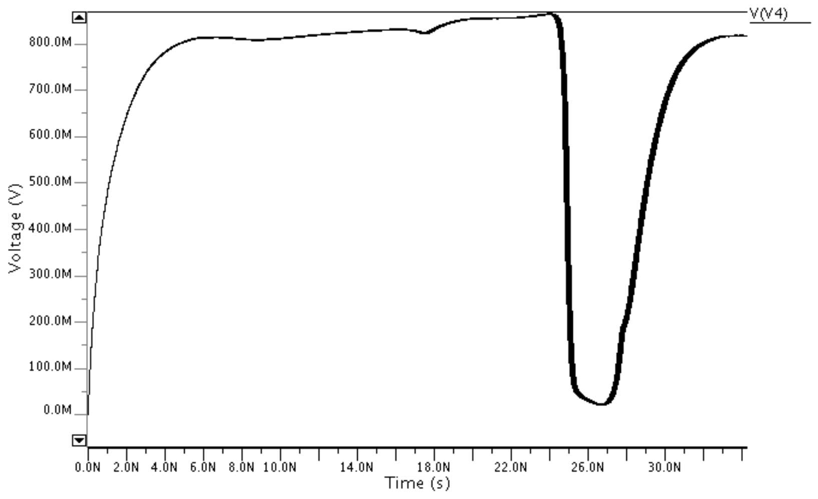


Figure 3.16: Waveform of one period of the Design 3 VCO.

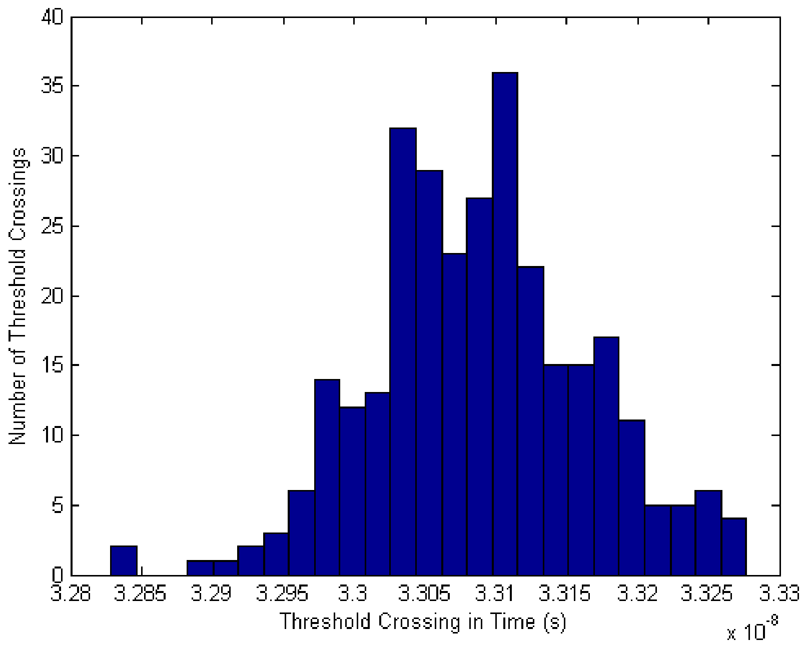


Figure 3.17: Timing jitter distribution for Design 3 at 300 noise runs and a threshold of 0.8V.

The standard deviation, and hence the jitter, of the threshold crossing histogram in Figure 3.17 was calculated to be 76.4ps.

3.5 D Flip-Flop

The DFF used for the TRNG and shown in Figure 3.18 was a sense-amplifier flip-flop covered in [17, 18]. The DFF operates using the clock signal and the sense-amplification of the D input and its compliment to control the SR latch at the bottom. While the clock is low S_b and R_b are both set high so that the NAND-based SR latch holds the current state. As soon as the clock goes high, the differential pair for the two D inputs turns on setting the source of either M_5 or M_7 to ground and activating one of the inverters (M_4-M_7), bringing its output,S_b or R_b, to ground. Since a NAND SR latch is active low, Q is set to equal D, and the circuit operates as a positive edge triggered DFF. Although setup and hold times are usually crucial factors in DFF and register design, they are not as significant for the TRNG. This is attributed to the of the nature of the system, it is not necessary for the input to pass all setup and hold conditions; as long as the times are smaller than the period of th D input, most of the output will propagate through the DFF as expected.

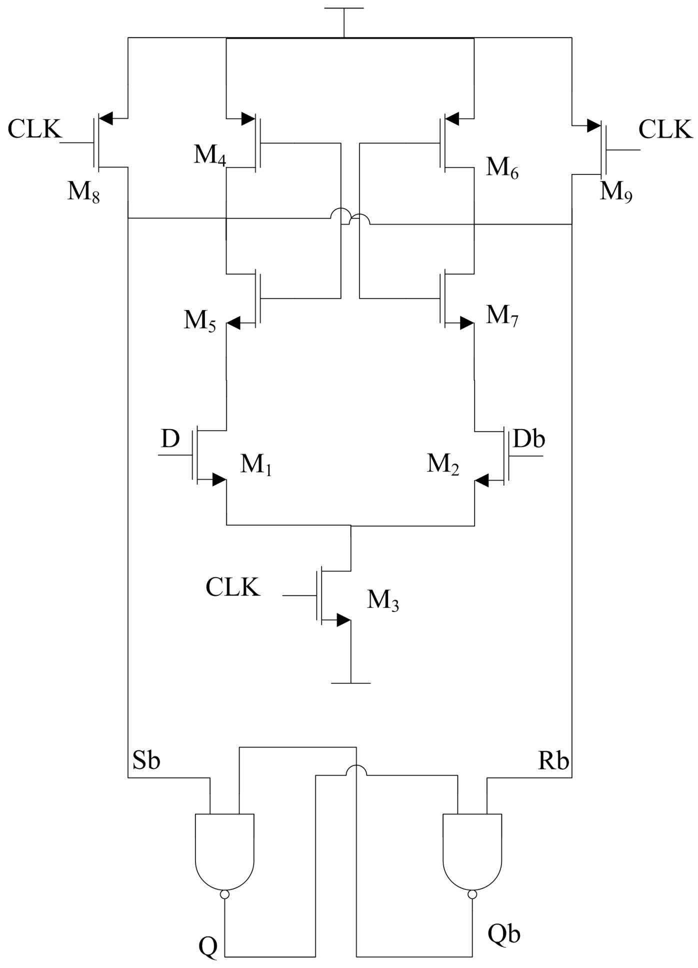


Figure 3.18: Sense amplifier DFF schematic.

3.6 Simulation Jitter Summary

Table 3.5: Summary of timing jitter from Eldo simulations.

Design	Jitter(ps)
D1	4.3173
D2	4.1293
D3	76.4

Chapter 4

TRNG Transistor Level Simulation

While the previous section demonstrated the functionality of the individual components of the TRNG, this section presents the results of the entire system. For the randomness tests, it was computationally inefficient to run 20,000 cycles of the whole system in Eldo to obtain the DFF output. Instead the timing jitter of the SO of each design was used with an ideal FO and DFF to produce the 20,000 bit output-stream. This bit-stream was then tested with the randomness suite. A FO with frequencies 1GHz, 5.5GHz, and 9GHz was used to calculate the output bit-stream from the SO jitter. 1GHz was the highest speed the extracted output buffers in Section 5.1 could transmit. 5.5GHz and 9GHz were the fastest frequencies that the FO could produce,with and without the consideration for parasitics, respectively.

4.1 Design 1

4.1.1 Transistor Level Simulation

Design 1 consists of the FO and the current-starved VCO. A test bench was created and is shown in Figure 4.1.

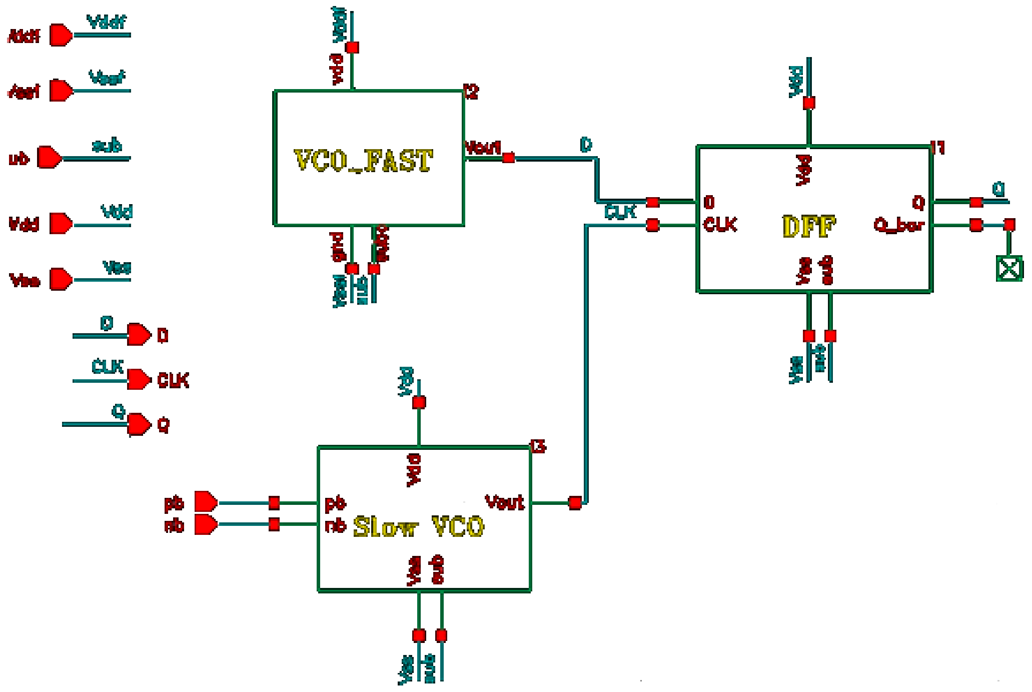


Figure 4.1: Simulation test bench for designs 1 and 2

The two components used as the D and clock inputs to the DFF, respectively, resulted in Figure 4.2. From top to bottom, graphs show the D input, the clock input and the Q output of the DFF. Figure 4.2 shows that the whole system operated correctly. Since this was a Cadence simulation, no noise was applied and the output waveform was deterministic. The clock frequency for this configuration was approximately 170MHz.

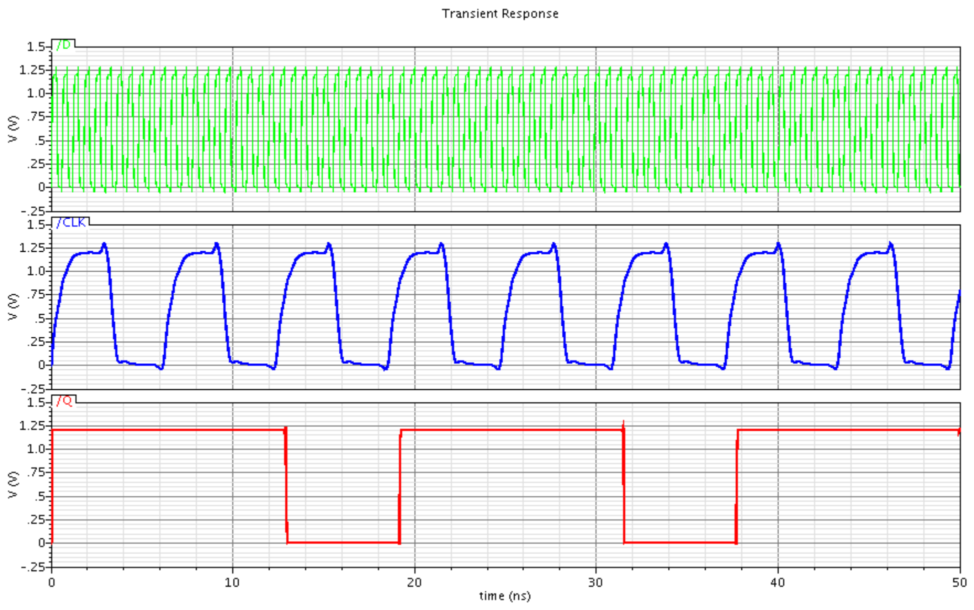


Figure 4.2: Transistor level simulation of Design 1.

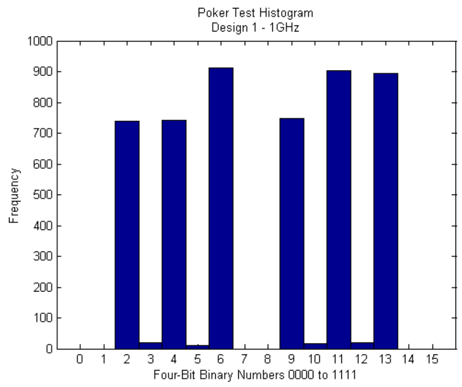
4.1.2 Randomness Test

For these tests an ideal FO in Matlab was used as the D input. For the clock, a 75MHz signal with a jitter of 4.32ps was used, as derived from Figure 3.8. A summary of NIST tests performed with 3 FO frequencies for 100 bit-streams is given in Table 4.1.

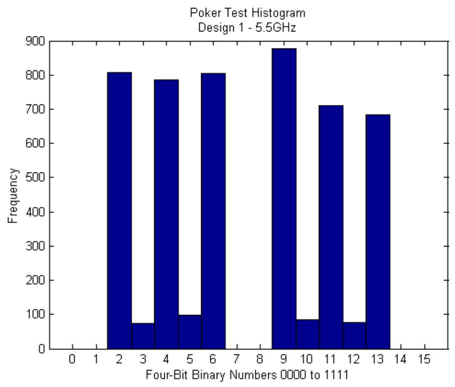
From the suite of NIST tests [6], it was determined that a number generator with this setup would not be considered random since all the tests did not pass. The frequency histograms, for one sequence, in Figure 4.3 are shown to be sporadic and uneven, indicating that the distribution of bits was deterministic.

Table 4.1: Summary of randomness tests for Design 1

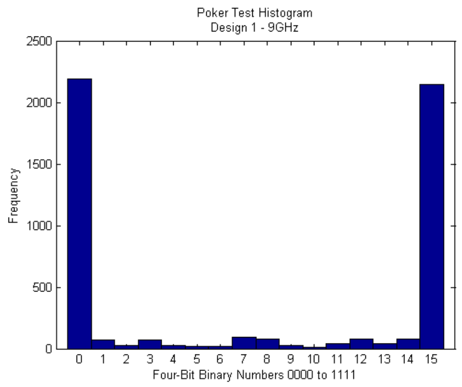
	1GHz		**5.5GHz**		**9GHz**	
Test	% Pass	Result?	% Pass	Result?	% Pass	Result?
Frequency	23/100	FAIL	92/100	FAIL	33/100	FAIL
Block Frequency	0/100	FAIL	4/100	FAIL	0/100	FAIL
Cumulative Sums (For.)	0/100	FAIL	87/100	FAIL	0/100	FAIL
Cumulative Sums (Rev.)	0/100	FAIL	85/100	FAIL	0/100	FAIL
Runs	0/100	FAIL	0/100	FAIL	0/100	FAIL
Longest Run	0/100	FAIL	0/100	FAIL	0/100	FAIL
FFT	0/100	FAIL	0/100	FAIL	0/100	FAIL
Approx. Entropy	0/100	FAIL	0/100	FAIL	0/100	FAIL
Serial 1	0/100	FAIL	0/100	FAIL	0/100	FAIL
Serial 2	0/100	FAIL	0/100	FAIL	0/100	FAIL



(a)



(b)



(c)

Figure 4.3: Four-bit distribution poker test for Design 1.

4.2 Design 2

4.2.1 Transistor Level Simulation

The system-wide test simulation was repeated for the second design. Design 2 consisted of the fast RO D input and the current-stealing CLK input. The output waveform is given in Figure 4.4. From top to bottom, graphs show the FO, the output of the current-stealing VCO (V1, blue) and its buffered output (clock, purple), and the

DFF Q output. This is again shown to be in working order but deterministic as no noise was introduced. The clock frequency for this configuration was approximately 200MHz.

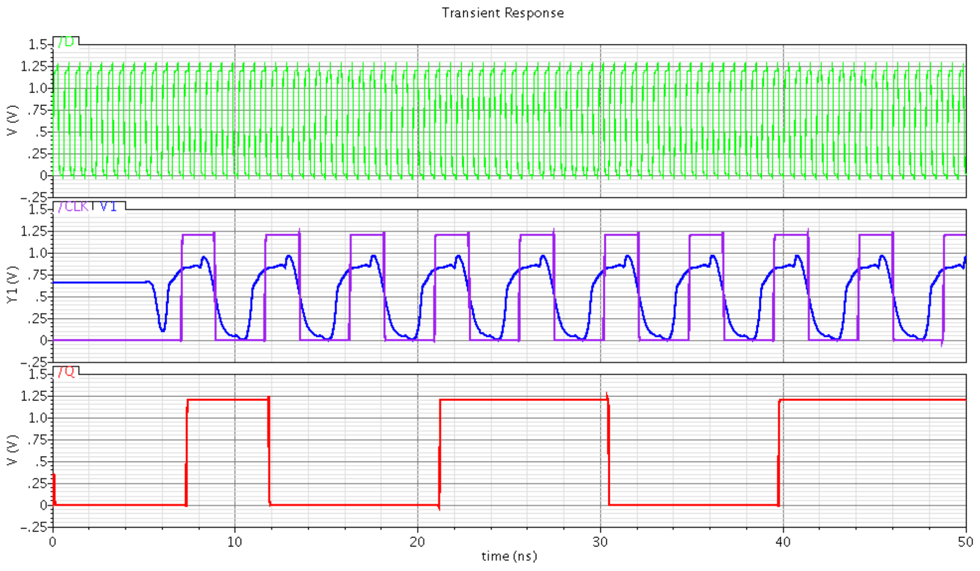


Figure 4.4: Transistor level simulation for Design 2

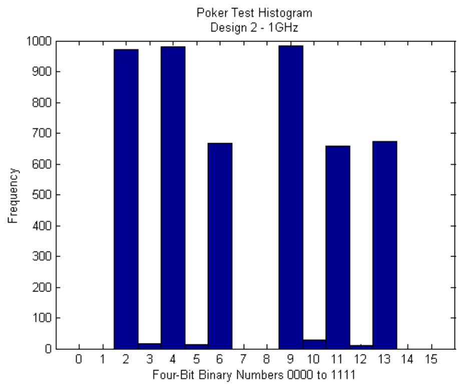
4.2.2 Randomness Tests

For these tests an ideal FO in Matlab was used as the D input. For the clock, a 60MHz signal with a jitter of 4.13ps was used, as derived from Figure 3.13. A summary of NIST tests performed with 3 FO frequencies for 100 bit-streams is given in Table 4.2.

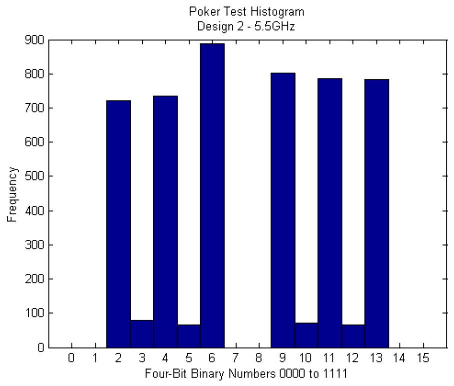
From the suite of NIST tests [6], it was determined that a number generator with this setup would not be considered random since all the tests did not pass. The frequency histograms, for one sequence, in Figure 4.5 are shown to be sporadic and uneven, indicating that the distribution of bits was deterministic.

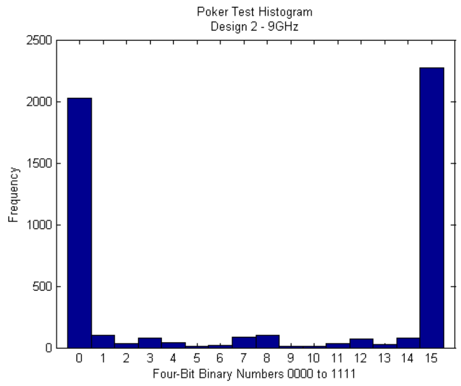
Table 4.2: Summary of randomness tests for Design 2

	1GHz		**5.5GHz**		**9GHz**	
Test	% Pass	Result?	% Pass	Result?	% Pass	Result?
Frequency	100/100	PASS	100/100	PASS	100/100	PASS
Block Frequency	100/100	PASS	100/100	PASS	100/100	PASS
Cumulative Sums (For.)	100/100	PASS	100/100	PASS	100/100	PASS
Cumulative Sums (Rev.)	100/100	PASS	100/100	PASS	100/100	PASS
Runs	0/100	FAIL	0/100	FAIL	0/100	FAIL
Longest Run	0/100	FAIL	0/100	FAIL	0/100	FAIL
FFT	0/100	FAIL	0/100	FAI	38/100	FAIL
Approx. Entropy	0/100	FAIL	0/100	FAIL	0/100	FAIL
Serial 1	0/100	FAIL	0/100	FAIL	0/100	FAIL
Serial 2	0/100	FAIL	0/100	FAIL	0/100	FAIL



(a)





(b) (c)

Figure 4.5: Four-bit distribution poker test for Design 2.

4.3 Design 3

The system-wide test was not repeated for design 3 because of the similarity in SO waveforms. The randomness tests from the SO were the only item of interest for this design.

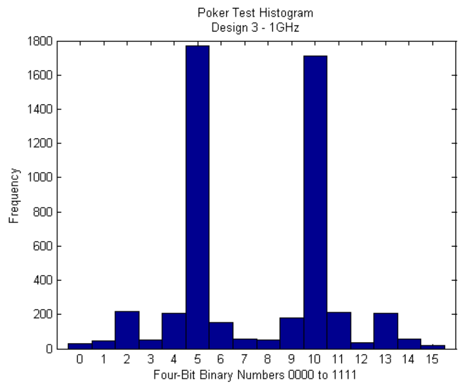
4.3.1 Randomness Tests

For the test an ideal FO in Matlab was used as the D input. For the clock, a 37.7MHz signal with a jitter of 76.4ps was used, as derived from Figure 3.17. A summary of NIST tests performed with 3 FO frequencies for 100 bit-streams is given in Table 4.3

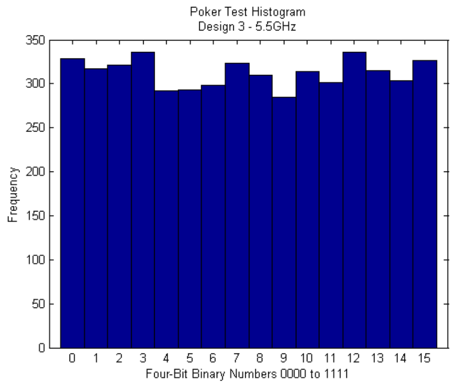
From the suite of NIST tests [6], it was determined that the number generator could be considered random at 9GHz because all tests passes at least 96 times. The poker test frequency histograms, for one sequence, in Figures 4.6(b) and 4.6(c) are shown to be level and even, indicating that the distribution of the bits appear to be random. The 5.5GHz tests show that almost all pass except for the entropy test, this shows that this is close to the lowest FO frequency possible. The 1GHz FO was not adequate for producing randomness in the output stream, even with a substantial amount of timing jitter.

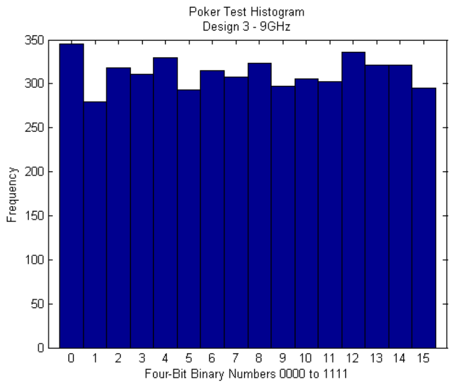
Table 4.3: Summary of randomness tests for Design 3

	1GHz		**5.5GHz**		**9GHz**	
Test	% Pass	Result?	% Pass	Result?	% Pass	Result?
Frequency	100/100	PASS	100/100	PASS	100/100	PASS
Block Frequency	100/100	PASS	97/100	PASS	99/100	PASS
Cumulative Sums (For.)	100/100	PASS	100/100	PASS	100/100	PASS
Cumulative Sums (Rev.)	100/100	PASS	100/100	PASS	100/100	PASS
Runs	0/100	FAIL	39/100	FAIL	99/100	PASS
Longest Run	0/100	FAIL	98/100	PASS	98/100	PASS
FFT	0/100	FAIL	100/100	PASS	97/100	PASS
Approx. Entropy	0/100	FAIL	93/100	FAIL	97/100	PASS
Serial 1	0/100	FAIL	98/100	PASS	98/100	PASS
Serial 2	0/100	FAIL	99/100	PASS	98/100	PASS



(a)





(b)

(c)

Figure 4.6: Four-bit distribution poker test for Design 3.

Chapter 5

Fabrication and Testing

All fabrication and layout designs were intended for use with the 0.13μm IBM CMOS technology. Minimum sizing for this technology is 120nm length and 160nm width. The standard power supply is 1.2V but can be increased to as high as 1.6V to improve the speed of the oscillators if required [19].

5.1 Buffer Design

All instruments used in testing had input impedances of 10-15pF, therefore in order to analyze the signals properly, each chip output required buffered. To achieve the correct driving ability, a simple inverter chain was used. Using the logical effort method for sizing a chain of inverters, it was determined that the optimum number of stages for a 15pF load was eight using an effective fan-out of three [16]. The layout for this buffer is given in Figure 5.1. Effective fan-out is is defined as the difference in sizes of two consecutive stages of an inverter chain. A fan-out of three therefore means that the widths of the second inverter are three times larger than the widths of the first inverter. For an effective fan-out of three the RC time constant for an inverter becomes too large and the delay for one stage is longer than the half period of the signal to be buffered, resulting in truncation of the output signal. A slow-speed buffer was therefore used for the sub-1GHz frequency outputs, such as the clocks and Q.

Another buffer was designed with a fan-out of 1.8, allowing the FO (5GHz-10GHz) to be analyzed off chip. Using this f a 17 stage buffer was created. A problem was encountered after the 13^{th} stage; there was not enough current being delivered to drive the capacitance of the next stages and the signal was consequently dying. To resolve this issue, only the first 13 stages were used, meaning that output signal was not rail-to-rail. Since the only value that was to be extracted for the FO output was the running frequency, this was decided to be an acceptable loss. The high-speed buffer is shown in Figure 5.2.

5.1.1 Layout

The slow-speed buffer used an area of $150\mu m x 60\mu m$ including the large guard ring. The guard ring was included to isolate large fluctuations in inverter supply voltage from the rest of the chip. The number of fingers for each transistor was increased at each stage so as to spread the large charging current onto many wires. This also helped to maintain a compact buffer.

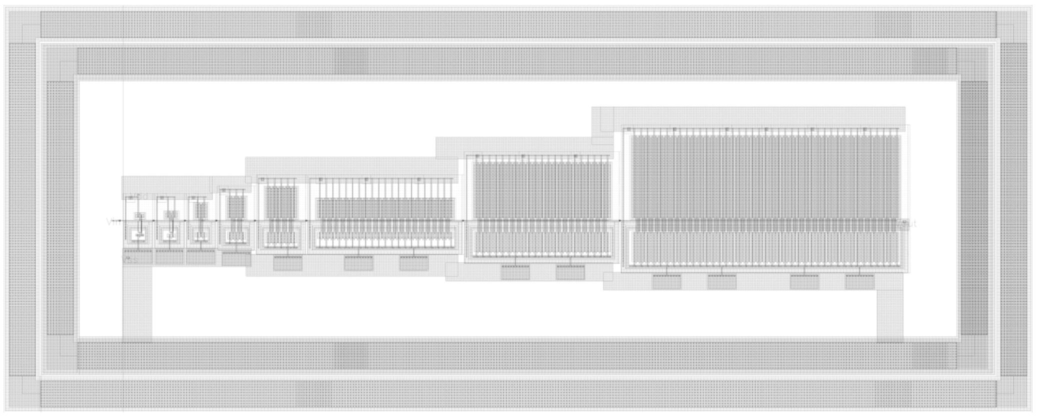


Figure 5.1: Layout of 8-stage slow-speed buffer.

The high-speed buffer was slightly larger at $160\mu m x 60\mu m$ because of the extra stages.

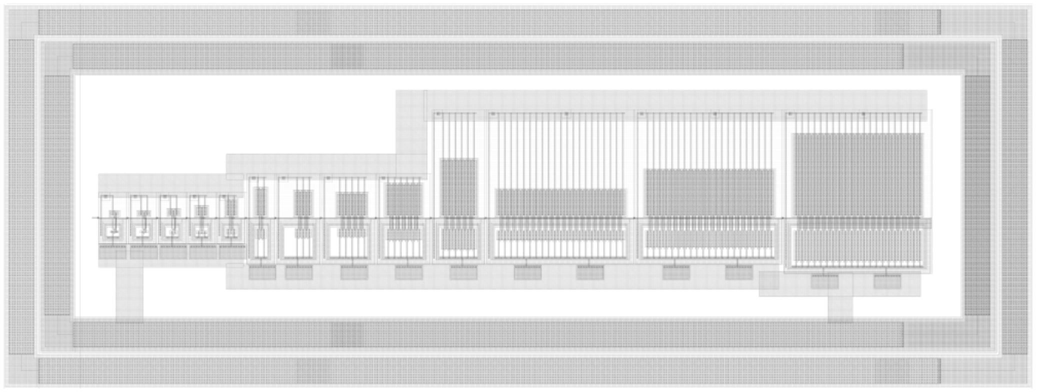


Figure 5.2: Layout of 13-stage high-speed buffer.

5.1.2 Parasitic Extraction and Simulations

Each buffer was laid out and the parasitic capacitance and resistance were extracted into a new netlist. These new extracted circuits were simulated to determine the performance of each buffer in a situation as close to the actual microchip as possible.

The slow-buffer was tested by passing an ideal sine wave at 200MHz to determine how well the slow VCO and DFF Q output could drive a 15pF scope load. The results of this test are given in Figure 5.3.

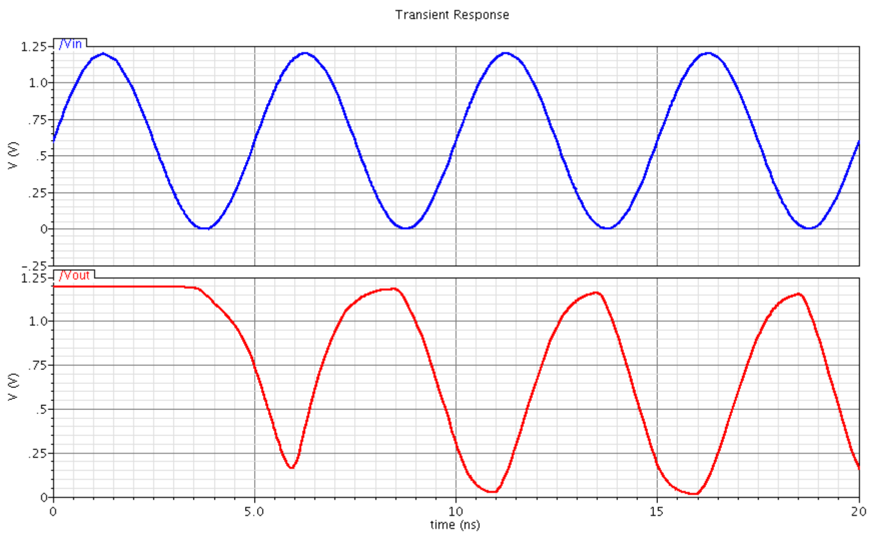


Figure 5.3: Input and output signal for the slow-speed buffer at 200MHz with a 15pF load.

The same test was repeated, where the frequency of the input was changed from 200MHz to 1GHz. The buffer had a difficult time producing a large signal. A peak-to-peak voltage of 200mV was desired in order for a clear signal to appear on the testing oscilloscope. Figure 5.4 shows that the slow-speed buffer could only produce a 150mVp-p signal at 1GHz.

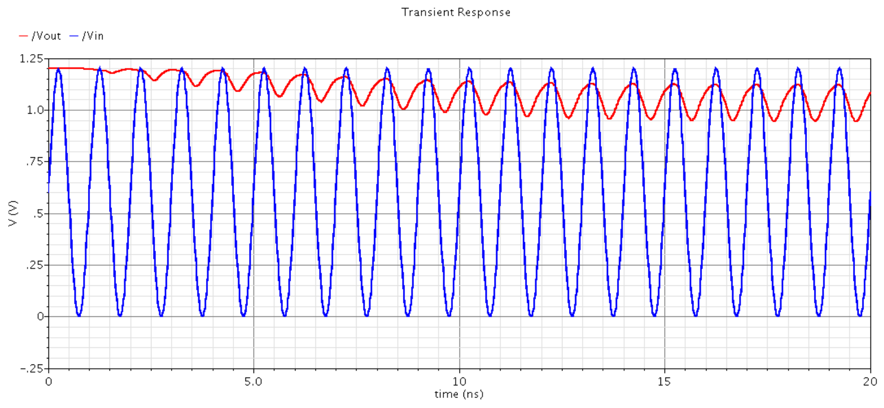


Figure 5.4: Input and output signal for the slow-speed buffer at 1GHz with a 15pF load.

The high-speed buffer was tested also at 1GHz and was able to produce a 350mVp-p output, as displayed in Figure 5.5. The high-speed buffer could not however go above 1.5GHz without the signal attenuated to an unacceptable level. The FO for both Design 1 and 2 were therefore redesigned to produce an extracted signal frequency of only 1GHz. This was much smaller than the 5.5GHz signal that the extracted fast VCO could produce originally and thus greatly affected the randomness of the Q output for each system.

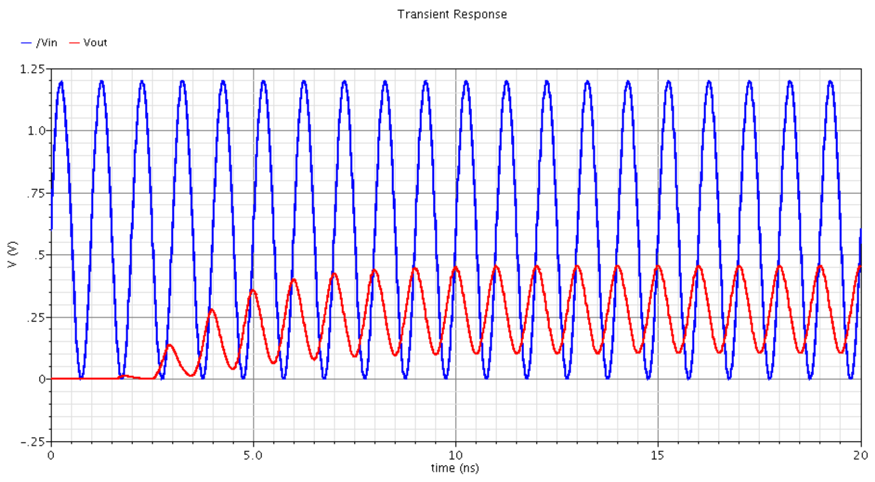


Figure 5.5: Input and output signal for the high-speed buffer at 1GHz with a 15pF load.

5.2 Design 1

After testing the high-speed buffer with extracted parasitics, it was determined that it could still not produce a 5GHz output signal, FO for both designs was therefore modified to approximately 1GHz in order to be able to read the output.

5.2.1 Layout

The layout for Design 1 with labelled sections is given in Figure 5.6. Two internal buffer chains were introduced to isolate each oscillator from its load and to supply sharp edge so that the inputs to the DFF were clear digital signals, either 0V or 1.2V, improving function and reducing glitches. An example of one delay cell for the current-starved VCO is illustrated in Figure 5.7. Each group of NMOS transistors was surrounded by a guard ring to prevent latch-up from occurring. The guard ring design could have been optimized for size by including all the NMOS transistors for the VCO, but it was decided to err on the side of caution and produce a layout that had the best chance of producing results. Design 1 occupied an area of $155\mu m by 55\mu m$.

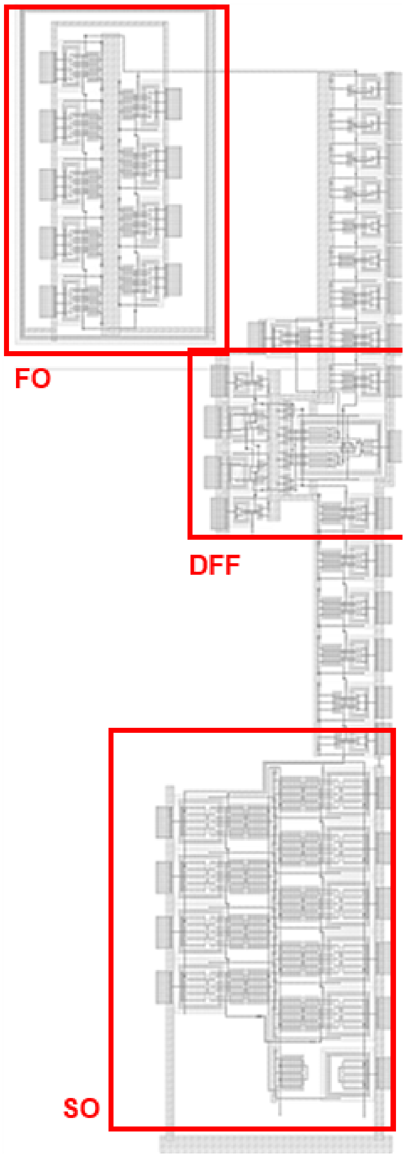


Figure 5.6: Layout of Design 1 TRNG.

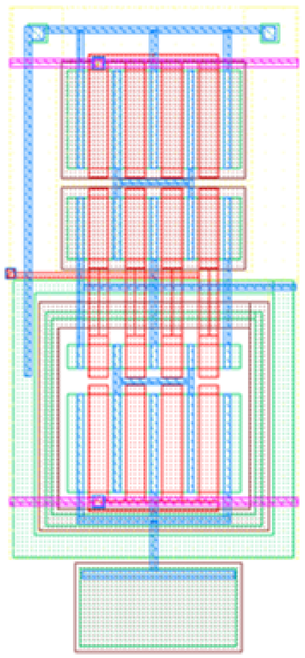


Figure 5.7: Layout of one current-starved delay cell.

5.2.2 Parasitic Extraction and Simulations

Design 1 was connected to two slow-speed buffers for the clock and Q signals and one high-speed buffer for the D input, and laid out in its exact form on the microchip to be submitted. The parasitic capacitances of each node were extracted using the CALIBRE tool on Cadence to create a new netlist with all elements included. This netlist was simulated and provided the waveforms shown in Figure 5.8. The D input swung rail-to-rail internally, had a peak-to-peak voltage of 300mV at a frequency of 1.02GHz. The extracted frequency of the noisy clock was 132.12MHz which as expected was smaller than the 170MHz simulated without the parasitic capacitance models. The Q output is shown to have a non-clock like waveform, but was still deterministic since no noise was introduced into the full system simulations.

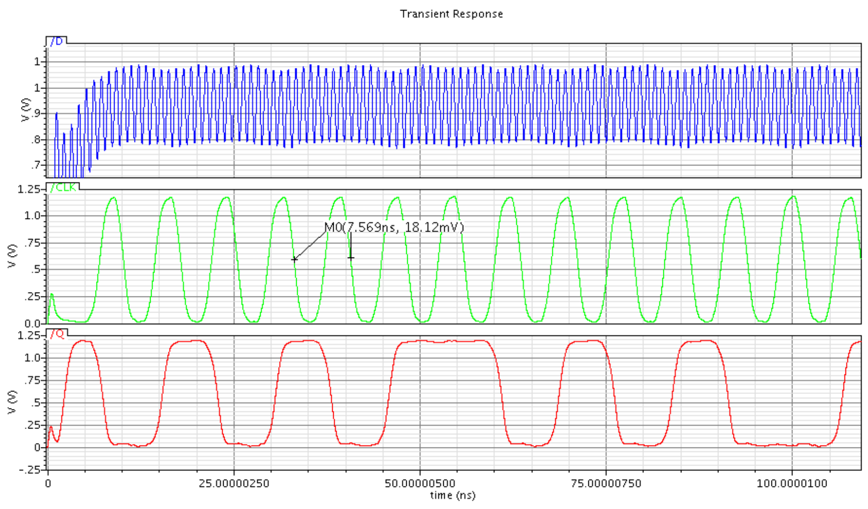


Figure 5.8: Design 1 full extraction simulation with 15pF load on each output.

5.3 Design 2

5.3.1 Layout

The layout for Design 2 with labelled sections is given in Figure 5.9. The area of Design 2 was $6800\mu m^2$. An example of one delay cell for the current-stealing VCO is illustrated in Figure 5.10.

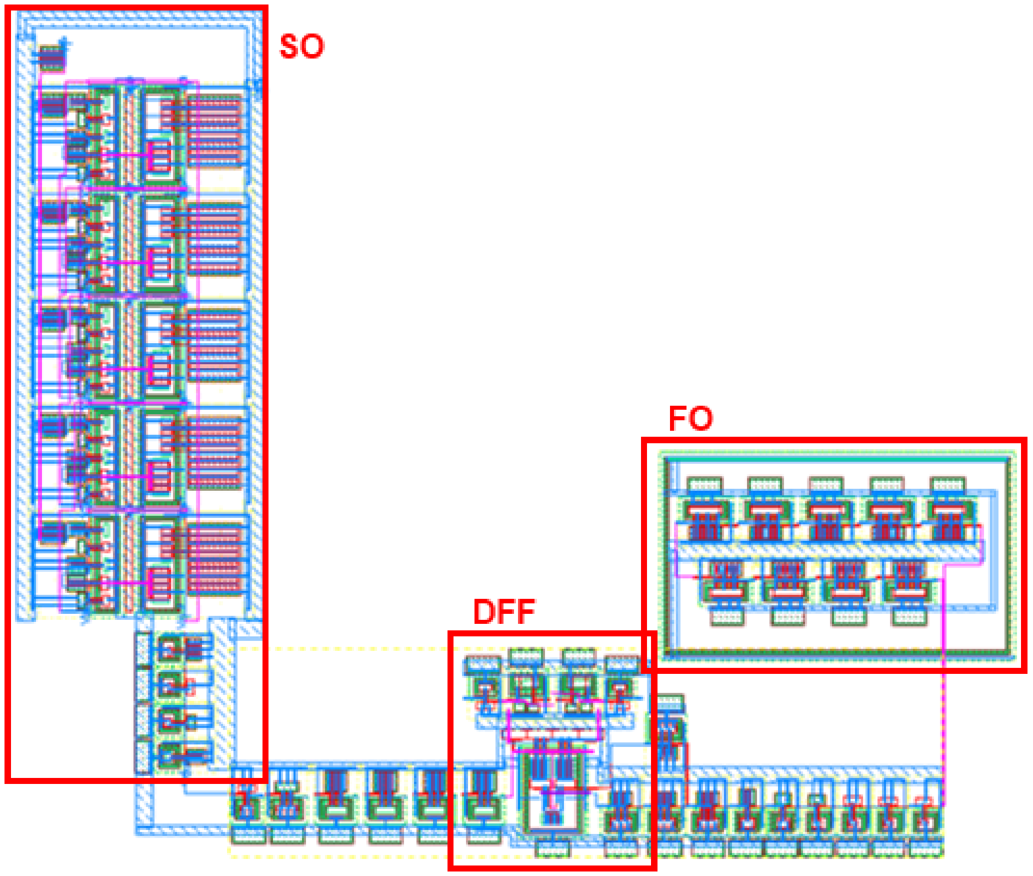


Figure 5.9: Layout of Design 2 TRNG.

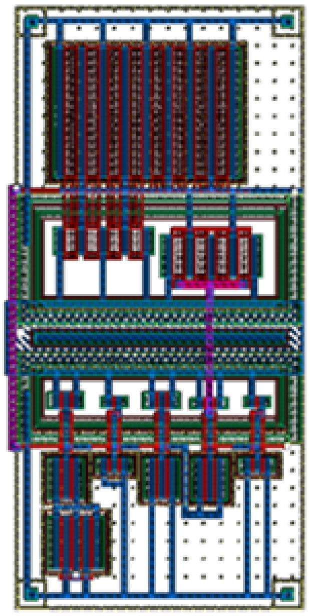


Figure 5.10: Layout of one current-stealing delay cell.

5.3.2 Parasitic Extraction and Simulations

Design 2 was connected to two slow-speed buffers for the clock and Q signals and one high-speed buffer for the D input, and was laid out in its exact form on the microchip to be submitted. The parasitic capacitances of each node were extracted using the CALIBRE tool on Cadence to create a new netlist with all elements included. This netlist was simulated and provided the waveforms shown in Figure 5.11. The D input that swung rail-to-rail internally, had a peak-to-peak voltage of 300mV at a frequency of 1.02GHz. The extracted frequency of the noisy clock was 86MHz, as expected, was smaller than the 200MHz simulated without parasitic capacitance. Since no noise was introduced into this simulation, the Q output had a deterministic, clock-like waveform. Figure 5.11 shows that Design 2 did function correctly.

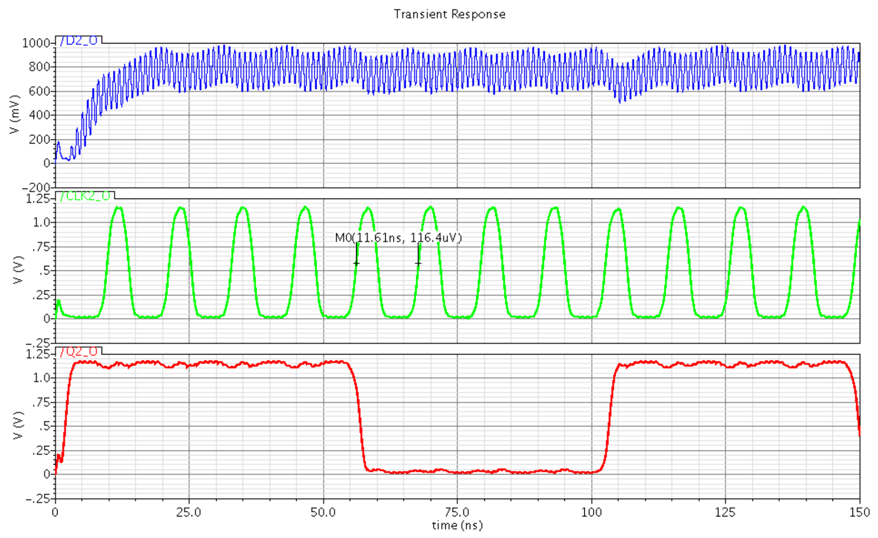


Figure 5.11: Design 2 full extraction simulation with 15pF load on each output

5.4 Layout considerations

The full microchip layout, including designs, buffers, ESD protection and metal filling, is given in Figure 5.12. The chip dimensions are 1mm by 1mm. The various parts are highlighted on the figure. The buffers were positioned at the top of the chip in order to isolate the large fluctuations in voltage from the design through the substrate (This placement could have potentially skewed the results of the test by adding more uncertainty making certain VCOs appear better at producing noise than others). Each design, as well as the group of buffers had their own V_{DD} and V_{SS} to further isolate the fluctuations. This also provided the ability to increase or decrease the supply voltage and consequently the speed of the design, allowing for finer control over the operation.

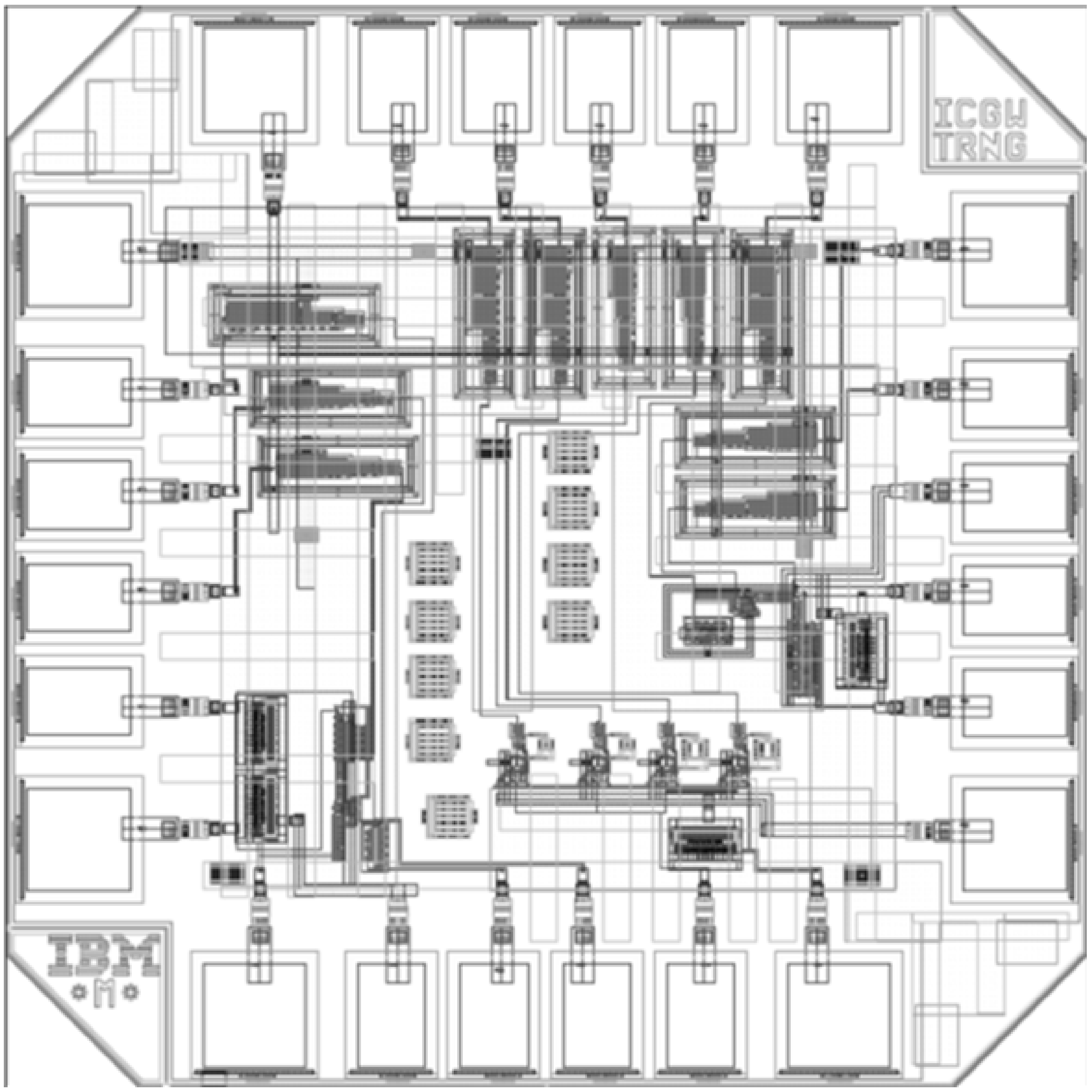


Figure 5.12: Full submitted chip layout for ICGWTRNG in 0.13um IBM technology.

5.4.1 ESD protection

Electrostatic discharge (ESD) protection was included by adding double-diodes provided by the IBM ESD library [20]. The double-diodes were used for all control input signals. If any input signals became higher than V_{DD} or lower than V_{SS}, the diodes turned on and redirected the current to the V_{DD} or V_{SS} pads, thus protecting the then gate oxides of the current mirror inputs. Figure 5.13 provides a schematic of the double-diode ESD protection.

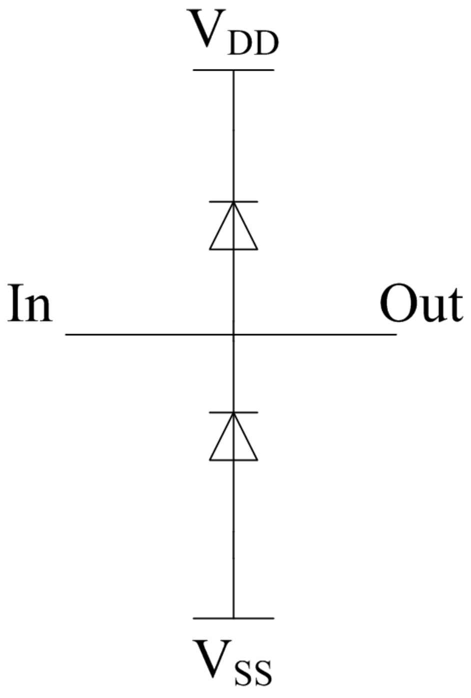


Figure 5.13: Schematic for the double-diode ESD protection.

For all other pads the very large drains of the last stage of each buffer were considered sufficient protection. For latch-up, all NMOS transistors connected to V_{SS} were separated from PMOS transistors connected to V_{DD} by a guard ring. This prevented

PNP to NPN connections from forming and sinking too much current, which would otherwise lead to those sections of the chip burning up.

5.5 PCB Layout

A 3" x 3" two-layer PCB was designed to test the chip. SMA connectors and single pins were used for each output to allow for easy testing setup. Jumpers were used for most supply paths as well as for connection of bias inputs to the current mirrors. This provided the ability to easily control what was turned on, as well as measure current in each of these paths. The PCB was fabricated by Albert Printed Circuit Boards.

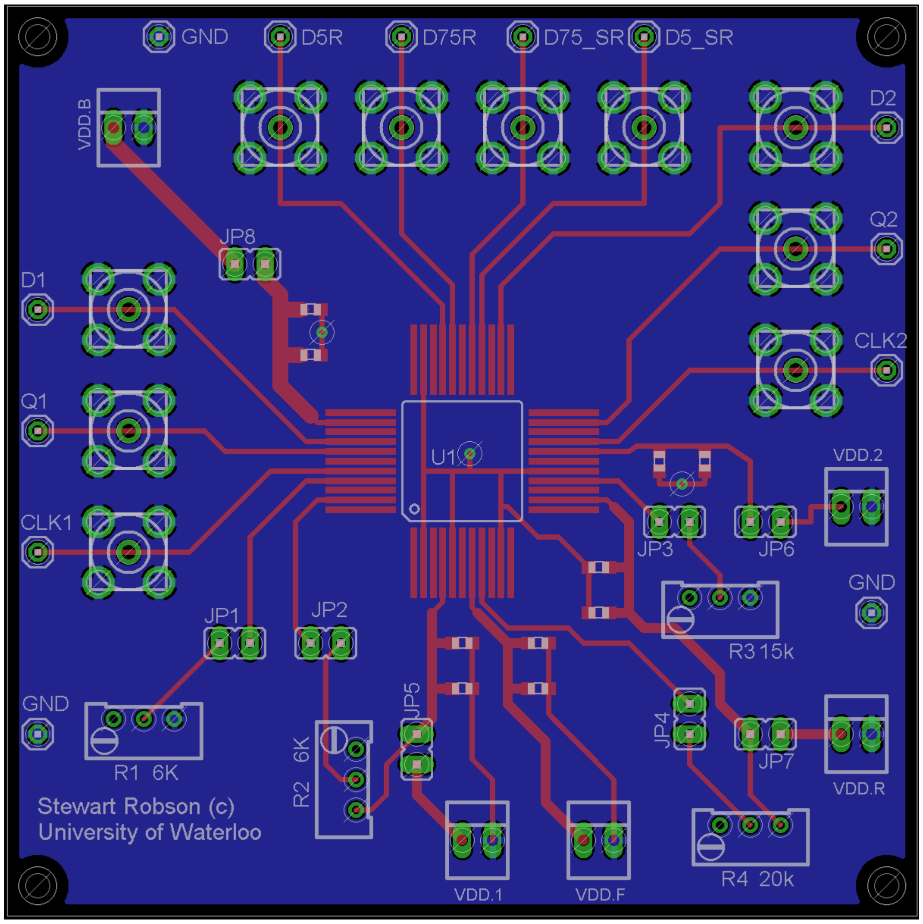


Figure 5.14: Screen shot of PCB design for testing the chip

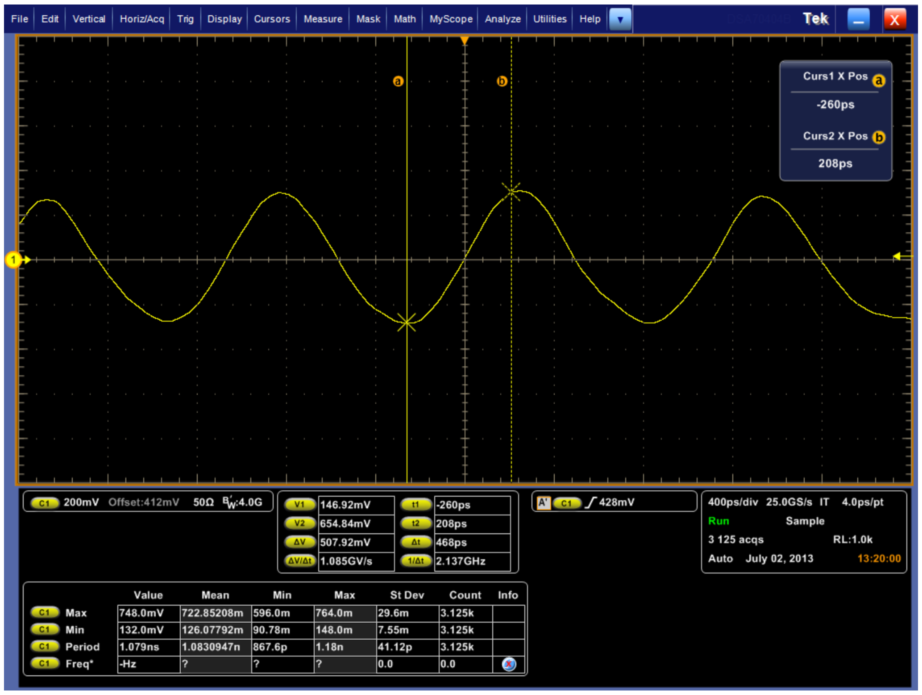
5.6 Testing

The 0.13μm chip was fabricated through CMC and The MOSIS Service company. The layout was successfully tested in previous sections in this chapter to show that Designs 1 and 2 would still function after fabrication. Design 3 was not finalized in time for the design submission deadline, so it was excluded from the fabrication.

5.6.1 Design 1

Due to constraints on the number of output pads on the chip only Design 1 had separate supply control for the FO. This extra control was implemented to help troubleshoot any problems that could be faced during testing. Figure 5.15 show the D input to both designs.

The biasing for the clock of Design 1 was altered to lower the frequency to 70MHz, as to make the comparisons to Design 2 better. The first waveform to be captured was the clock output without the fast RO being turned on. This allowed for a clean signal to be observed without any supply coupling from the other RO to affect the frequency of oscillation.



Curs1 X Pos a
-260ps
Curs2 X Pos b
208ps

C1 200mV Offset:412mV 50Ω BW:4.0G

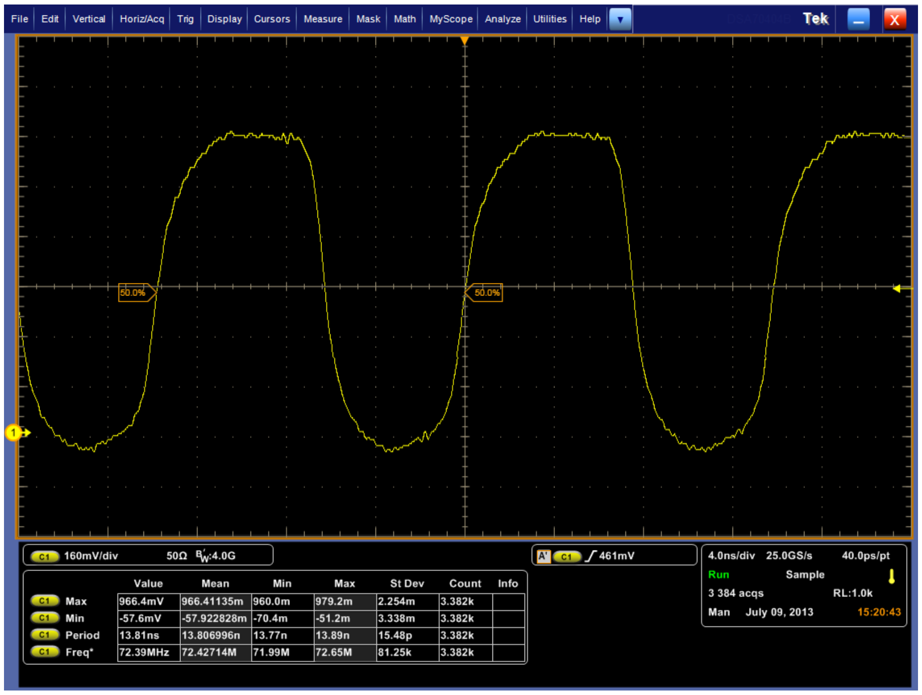
V1	146.92mV	t1	-260ps
V2	654.84mV	t2	208ps
ΔV	507.92mV	Δt	468ps
ΔV/Δt	1.085GV/s	1/Δt	2.137GHz

A' C1 ∫ 428mV

400ps/div 25.0GS/s IT 4.0ps/pt
Run Sample
3 125 acqs RL:1.0k
Auto July 02, 2013 13:20:00

	Value	Mean	Min	Max	St Dev	Count	Info
C1 Max	748.0mV	722.85208m	596.0m	764.0m	29.6m	3.125k	
C1 Min	132.0mV	126.07792m	90.78m	148.0m	7.55m	3.125k	
C1 Period	1.079ns	1.0830947n	867.6p	1.18n	41.12p	3.125k	
C1 Freq*	-Hz	?	?	?	0.0	0.0	

Figure 5.15: Fast RO output from Design 1. Running frequency = 923MHz



C1 160mV/div 50Ω B'W:4.0G

A' C1 461mV

4.0ns/div 25.0GS/s 40.0ps/pt
Run Sample
3 384 acqs RL:1.0k
Man July 09, 2013 15:20:43

	Value	Mean	Min	Max	St Dev	Count	Info
C1 Max	966.4mV	966.41135m	960.0m	979.2m	2.254m	3.382k	
C1 Min	-57.6mV	-57.922828m	-70.4m	-51.2m	3.338m	3.382k	
C1 Period	13.81ns	13.806996n	13.77n	13.89n	15.48p	3.382k	
C1 Freq*	72.39MHz	72.42714M	71.99M	72.65M	81.25k	3.382k	

Figure 5.16: On-chip Design 1 clock waveform with FO turned off. Running frequency = 72.4MHz

A 20,000 bit long waveform from the clean clock in Figure 5.16 was extracted into Matlab and the cycle-to-cycle jitter was calculated to be 17.33ps. The jitter distribution of this clean clock is given in Figure 5.17.

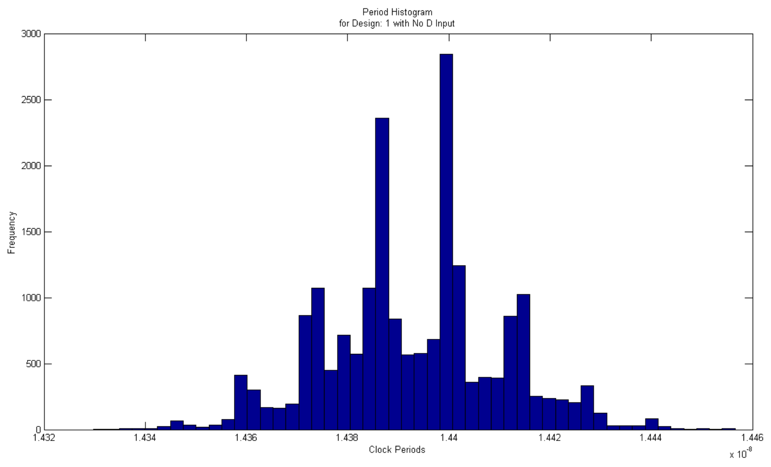


Figure 5.17: Threshold crossing histogram for a clean CLK signal with D turned off

The Tektronix application DPOjet was also used to obtain timing jitter statistics. In Figure 5.18 the eye diagram and time interval error [21] for 50,000 cycle of the clean clock were derived.

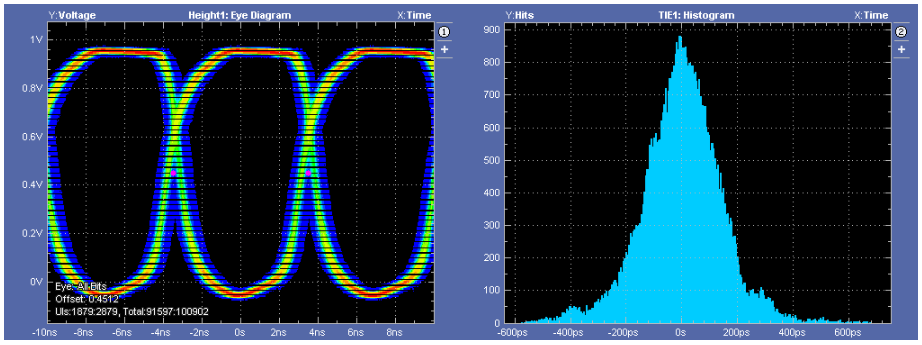
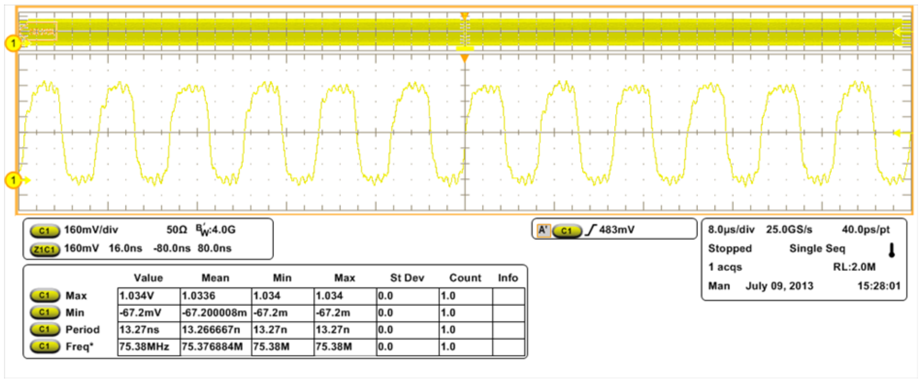


Figure 5.18: DPOJet eye-diagram and time interval error distribution of the clean clock waveform

Table 5.1: Summary of DPOjet jitter stats for Design 1 for clean clock

Description	Mean	Std Dev	Population
TIE	0.0000s	143.58ps	50446
RJdd1	19.097ps	3.6055ps	35
DJdd1	187.20ps	99.122ps	35

The FO was then connected. The clock waveform in Figure 5.19 showed many distortions and the rails that would effect overall timing.


C1 160mV/div 50Ω BW:4.0G
Z1C1 160mV 16.0ns -80.0ns 80.0ns

A' C1 483mV

8.0µs/div 25.0GS/s 40.0ps/pt
Stopped Single Seq
1 acqs RL:2.0M
Man July 09, 2013 15:28:01

	Value	Mean	Min	Max	St Dev	Count	Info
C1 Max	1.034V	1.0336	1.034	1.034	0.0	1.0	
C1 Min	-67.2mV	-67.200008m	-67.2m	-67.2m	0.0	1.0	
C1 Period	13.27ns	13.266667n	13.27n	13.27n	0.0	1.0	
C1 Freq*	75.38MHz	75.376884M	75.38M	75.38M	0.0	1.0	

Figure 5.19: Screen shot of PCB design for testing the chip. Running frequency = 72.4MHz

A 20,000 bit-string from the regular clock in Figure 5.19 was recorded and Matlab was used to calculate the cycle-to-cycle jitter which was 951.713ps. The jitter distribution of this regular clock is given in Figure 5.20. The jitter did not follow a normal distribution so the calculated jitter isn't as meaningful in regards to comparing numbers to the simulated calculation from from Figure 3.8.

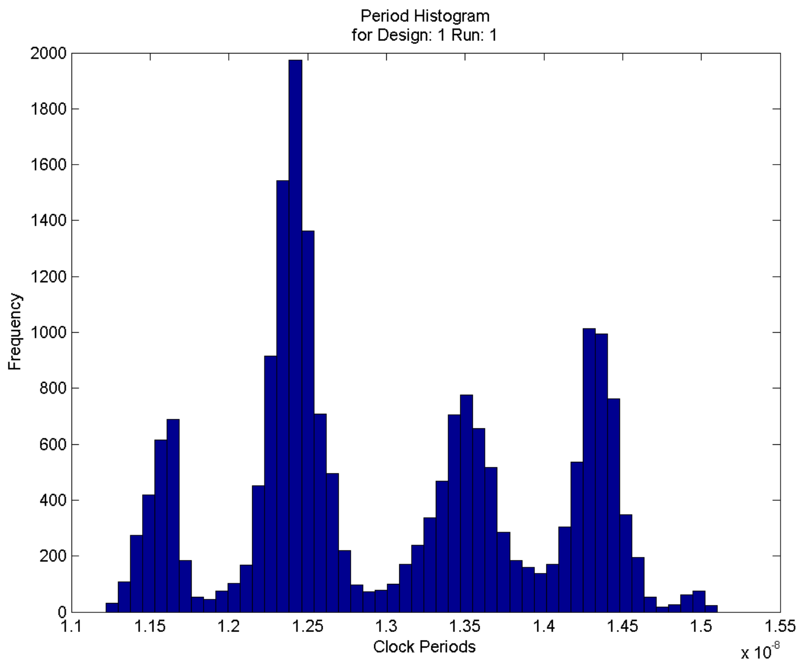


Figure 5.20: Threshold crossing histogram for a clean CLK signal with D turned off

The Tektronix application DPOjet was also used to obtain timing jitter statistics. in Figure 5.18 the eye diagram and time interval error [21].

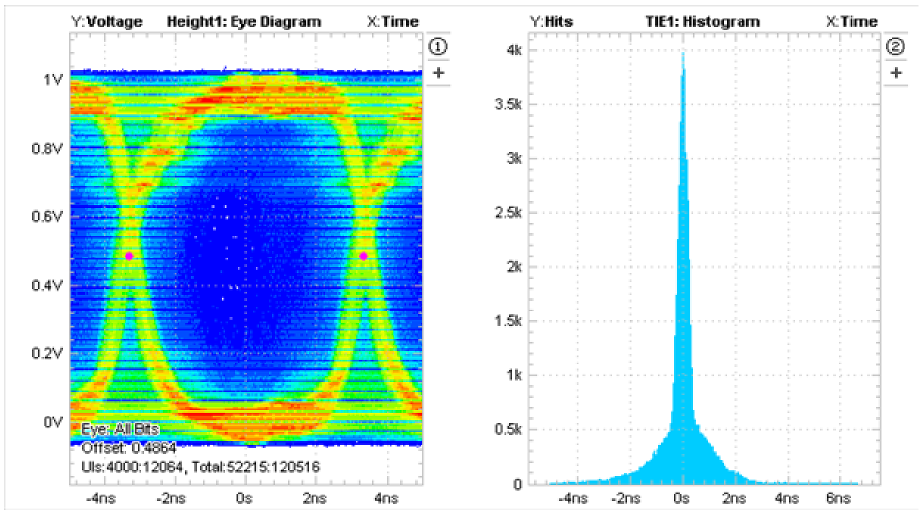
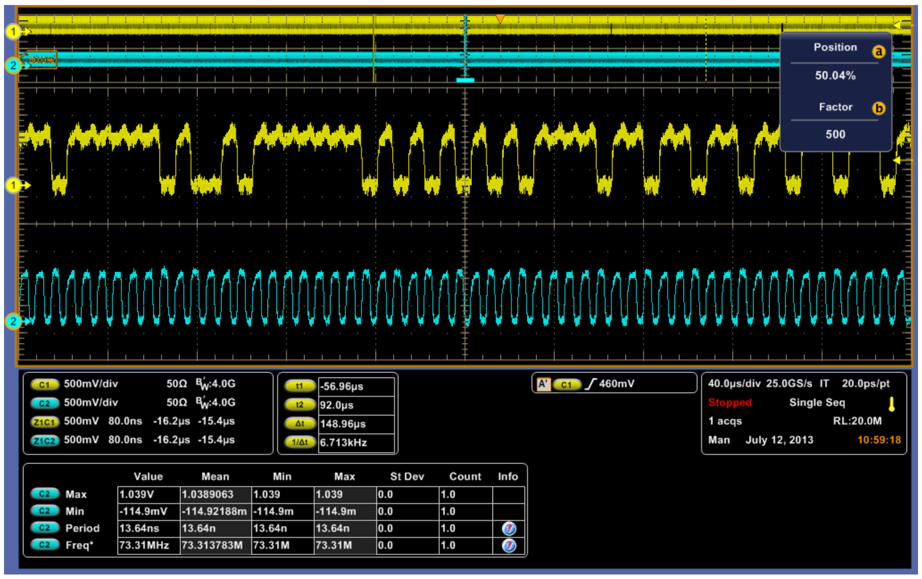


Figure 5.21: DPOJet eye-diagram and time interval error distribution of the clean clock waveform

Table 5.2: Summary of DPOjet jitter stats for Design 1 for regular clock

Description	Mean	Std Dev	Population
TIE	0.0000s	949.78ps	60259
RJdd1	152.31ps	148.87ps	10
DJdd1	1.6215ns	1.2567ns	10

Figure 5.22 shows an example of the Q and clock on-chip outputs.



```
C1   500mV/div        50Ω  BW:4.0G
C2   500mV/div        50Ω  BW:4.0G
Z1C1 500mV  80.0ns  -16.2μs  -15.4μs
Z1C2 500mV  80.0ns  -16.2μs  -15.4μs

t1    -56.96μs
t2    92.0μs
Δt    148.96μs
1/Δt  6.713kHz

A' C1  460mV

40.0μs/div 25.0GS/s  IT   20.0ps/pt
Stopped          Single Seq
1 acqs                  RL:20.0M
Man     July 12, 2013   10:59:18
```

		Value	Mean	Min	Max	St Dev	Count	Info
C2	Max	1.039V	1.0389063	1.039	1.039	0.0	1.0	
C2	Min	-114.9mV	-114.92188m	-114.9m	-114.9m	0.0	1.0	
C2	Period	13.64ns	13.64n	13.64n	13.64n	0.0	1.0	?
C2	Freq*	73.31MHz	73.313783M	73.31M	73.31M	0.0	1.0	?

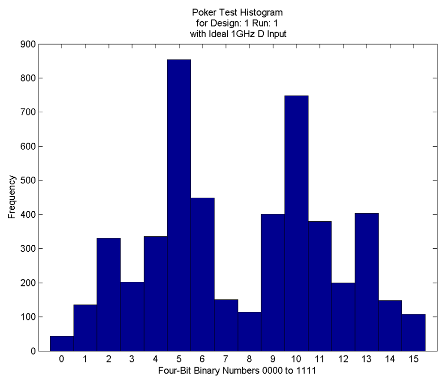
Figure 5.22: Design 1 clock and Q ouput from the chip

Randomness Tests

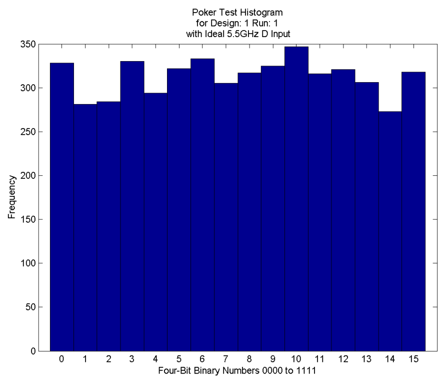
The Design 1 clock was compared to three ideal FO frequencies to obtain three sets of 10 bit-streams to be tested against the 10 bit-streams obtained on-chip. Due to lack of time only 10 bit-streams could be acquired from the clock and Q of Design 1. Table 5.3 provides a summary of the results obtained for the randomness tests. Figure 5.23 shows one poker test distribution for each set of bit-streams tested.

Table 5.3: Summary of randomness tests for chip output of Design 1

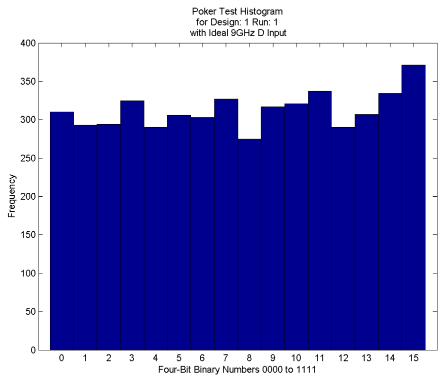
	1GHz		**5.5GHz**		**9GHz**		**On-Chip**	
Test	% Pass	Result?	% Pass	Result?	% Pass	Result?	% Pass	Result?
Frequency	8/10	PASS	10/10	PASS	10/10	PASS	0/10	FAIL
Block Frequency	10/10	PASS	10/10	PASS	10/10	PASS	0/10	FAIL
Cumulative Sums (For.)	8/10	PASS	10/10	PASS	10/10	PASS	0/10	FAIL
Cumulative Sums (Rev.)	9/10	PASS	10/10	PASS	10/10	PASS	0/10	FAIL
Runs	0/10	FAIL	3/10	FAIL	6/10	FAIL	0/10	FAIL
Longest Run	0/10	FAIL	6/10	FAIL	9/10	PASS	0/10	FAIL
FFT	2/10	FAIL	10/10	PASS	9/10	PASS	3/10	FAIL
Approx. Entropy	0/10	FAIL	2/10	FAIL	7/10	FAIL	0/10	FAIL
Serial 1	0/10	FAIL	5/10	FAIL	9/10	PASS	0/10	FAIL
Serial 2	0/10	FAIL	9/10	PASS	10/10	PASS	0/10	FAIL



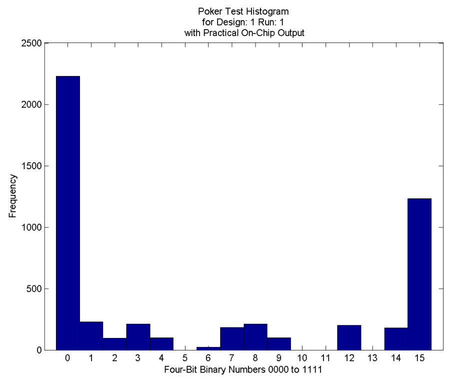
(a)



(b)



(c)

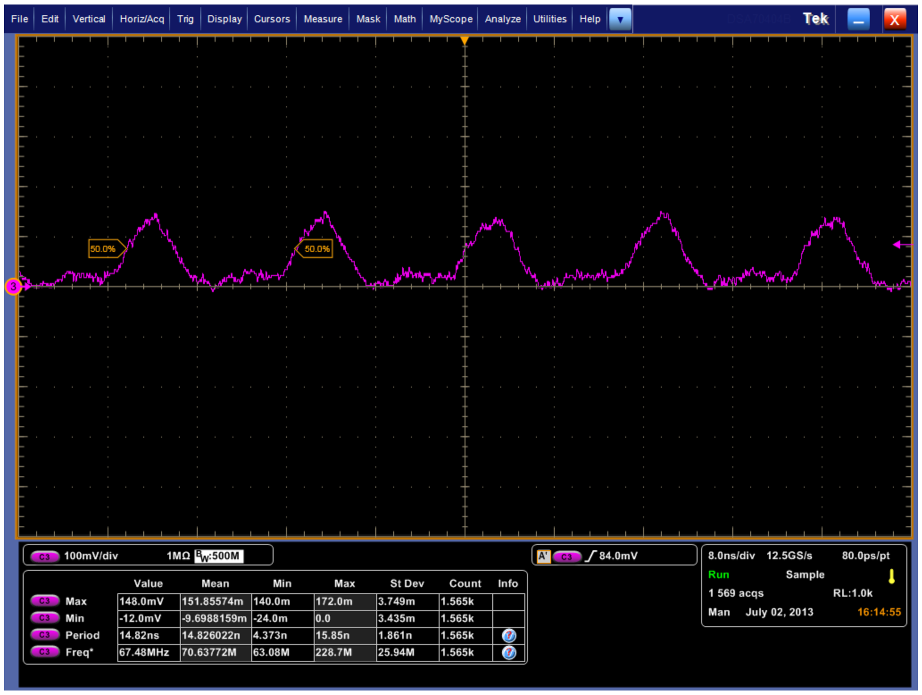


(d)

Figure 5.23: Four-bit distribution poker test for chip ouput for Design 1

5.6.2 Design 2

The on-chip clock for Design 2 is shown in Figure 5.24. It had a much smaller peak-to peak voltage than expected but the frequency of 60MHz was close to the extracted simulation frequency. Further testing was required to troubleshoot the operation of the clock output.



	Value	Mean	Min	Max	St Dev	Count	Info
C3 Max	148.0mV	151.85574m	140.0m	172.0m	3.749m	1.565k	
C3 Min	-12.0mV	-9.6988159m	-24.0m	0.0	3.435m	1.565k	
C3 Period	14.82ns	14.826022n	4.373n	15.85n	1.861n	1.565k	?
C3 Freq*	67.48MHz	70.63772M	63.08M	228.7M	25.94M	1.565k	?

```
8.0ns/div   12.5GS/s        80.0ps/pt
Run             Sample
1 569 acqs                  RL:1.0k
Man     July 02, 2013       16:14:55
```

Figure 5.24: Design 2 clock output from chip. Frequency = 61MHz.

A 20,000 bit-string from the regular clock in Figure 5.24 was recorded and Matlab was used to calculate the cycle-to-cycle jitter which was 1.506ns. The jitter distribution of this regular clock is given in Figure 5.25.

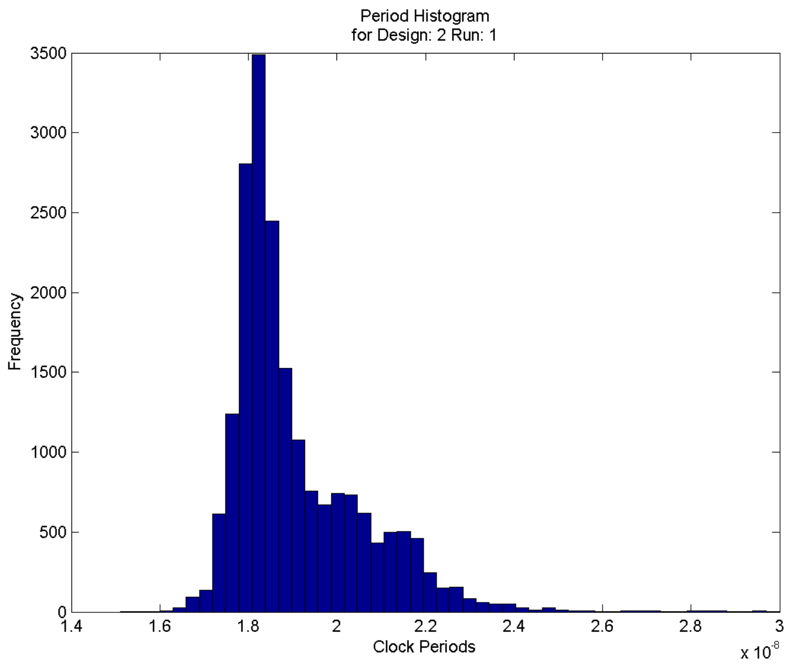


Figure 5.25: Threshold crossing histogram for the Design 2 clock from chip

The Tektronix application DPOjet was also used to obtain timing jitter statistics. in Figure 5.26 the eye diagram and time interval error [21].

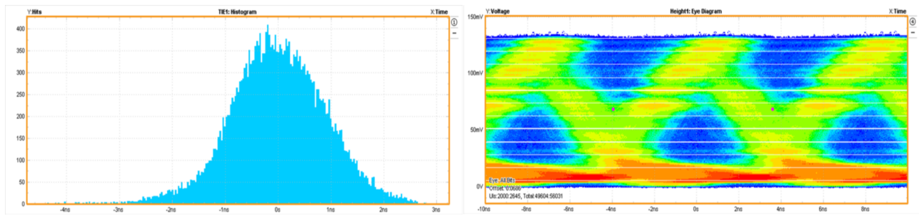
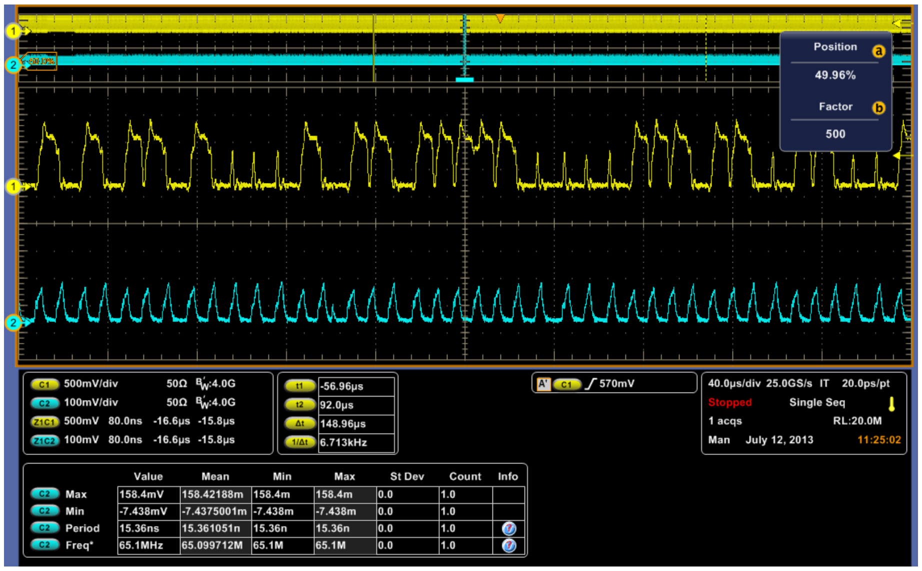


Figure 5.26: DPOJet eyediagram and time interval error distribution of the Design 2 clock from chip

Table 5.4: Summary of DPOjet jitter stats for Design 3 clock from chip

Description	Mean	Std Dev	Population
TIE1, Ch1	0.0000s	891.72ps	27845
RJdd1, Ch1	340.08ps	86.897ps	22
DJdd1, Ch1	1.2655ns	881.45ps	22

Figure 5.27 shows an example of the Q and clock on-chip outputs.



```
C1    500mV/div        50Ω  BW:4.0G       t1    -56.96μs
C2    100mV/div        50Ω  BW:4.0G       t2    92.0μs
Z1C1  500mV  80.0ns  -16.6μs  -15.8μs     Δt    148.96μs
Z1C2  100mV  80.0ns  -16.6μs  -15.8μs     1/Δt  6.713kHz

A' C1 570mV

40.0μs/div 25.0GS/s  IT   20.0ps/pt
Stopped          Single Seq
1 acqs                RL:20.0M
Man    July 12, 2013      11:25:02
```

	Value	Mean	Min	Max	St Dev	Count	Info
C2 Max	158.4mV	158.42188m	158.4m	158.4m	0.0	1.0	
C2 Min	-7.438mV	-7.4375001m	-7.438m	-7.438m	0.0	1.0	
C2 Period	15.36ns	15.361051n	15.36n	15.36n	0.0	1.0	?
C2 Freq*	65.1MHz	65.099712M	65.1M	65.1M	0.0	1.0	?

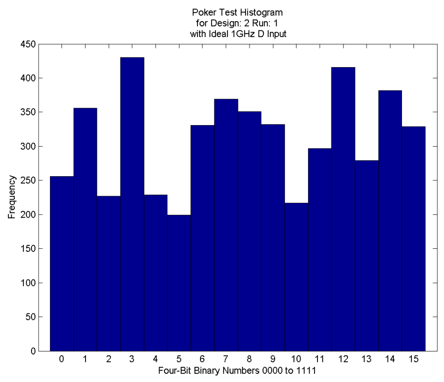
Figure 5.27: Design 2 CLK and Q ouput from the chip

Randomness Tests

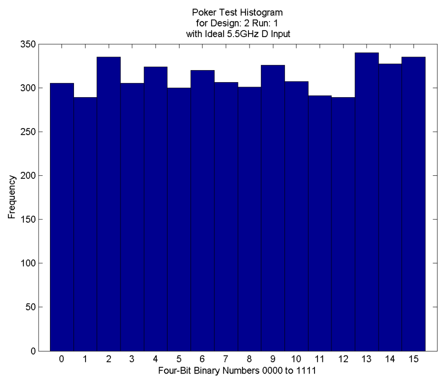
The Design 2 clock was compared to three ideal FO frequencies to obtain three sets of 100 bit-streams to be tested against the 5 bit-streams obtained on-chip. Due to time constraints only 5 sets of bit-streams was acquired from the on-chip Q for Design 2. Table 5.5 provides a summary of the results obtained for the randomness tests. Figure 5.28 shows one poker test distribution for each set of bit-streams tested.

Table 5.5: Summary of randomness tests for chip output of Design 2.

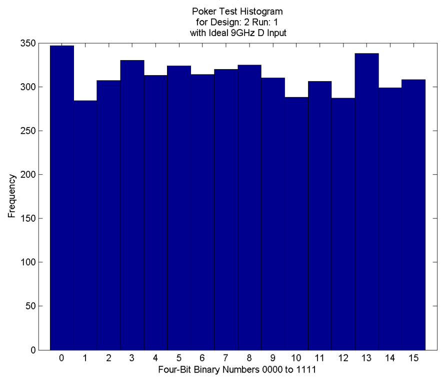
	1GHz		**5.5GHz**		**9GHz**		**On-Chip**	
Test	% Pass	Result?	% Pass	Result?	% Pass	Result?	% Pass	Result?
Frequency	35/100	FAIL	100/100	PASS	97/100	PASS	0/5	FAIL
Block Frequency	90/100	FAIL	100/100	PASS	98/100	PASS	0/5	FAIL
Cumulative Sums (For.)	39/100	FAIL	100/100	PASS	96/100	PASS	0/5	FAIL
Cumulative Sums (Rev.)	37/100	FAIL	100/100	PASS	96/100	PASS	0/5	FAIL
Runs	26/100	FAIL	99/100	PASS	100/100	PASS	0/5	FAIL
Longest Run	72/100	FAIL	98/100	PASS	98/100	PASS	0/5	FAIL
FFT	99/100	PASS	98/100	PASS	100/100	PASS	1/5	FAIL
Approx. Entropy	45/100	FAIL	98/100	PASS	98/100	PASS	0/5	FAIL
Serial 1	94/100	FAIL	99/100	PASS	98/100	PASS	0/5	FAIL
Serial 2	98/100	PASS	100/100	PASS	100/100	PASS	0/5	FAIL



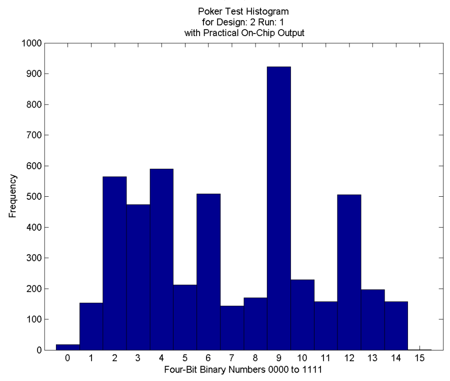
(a)



(b)



(c)



(d)

Figure 5.28: Four-bit distribution of poker test for chip output for Design 2

5.6.3 Summary

Both designs failed to produce random bit-streams at the FO frequency fabricated. However, when compared to an ideal FO with higher frequency both design showed randomness.

Design 2 showed a slightly more uniform distribution than Design 1 as shown in the poker test in Figure 5.23(d) and Figure 5.28(d). This was expected because of the increased amount of jitter observed in Figure 5.25 over Figure 5.17.

Chapter 6

Conclusions

Three ring oscillator based TRNGs were designed using a noisy VCO to create randomness. Design 1 used a standard current-starved delay-cell as the RNG clock, and had the lowest timing jitter of all the designs created. Design 2 used a newly designed current-stealing, low-slewing delay-cell. The exploitation of multiple crossings and LPT resulted in improved jitter over the previous design, but not quite to the desired extent. The difficulty rose from setting the switching threshold of the subsequent stage to the low-slew phase level. This issue was alleviated by creating Design 3 a modification of Design 2. Design 3 involved inserting simple two-transistor inverters in between each current-stealing cell, allowing for easier control of the threshold. In addition, more noise was introduced through extra transistors on each current-stealing delay-cell. Design 3 provided exceptional timing jitter, 75ps, proving that multiple crossings and LPT were being utilized.

The outputs of each design were tested under a suite of tests outlined by the NIST. The results of the tests indicated that the first two designs were not sufficiently random. Only Design 3 provided adequate noise to obtain the required randomness.

Excessive amount of noise in the final design allows for further customization of the TRNG, as speed can be increased while still delivering acceptable randomness. This would improve the overall speed in which the seed from the TRNG is delivered.

Designs 1 and 2 were both fabricated onto a $0.13\mu m$ process chip and tested with an oscilloscope and Matlab. The results showed that both on-chip outputs with FO of 1GHz were not random. Design 2 was slightly more random than design 1. The SO waveform was extracted for both designs and used in conjunction with Matlab to test FOs with frequency of 5.5GHz and 9 GHz, resulting in random bit-streams from both designs.

In comparison to other oscillator based RNG research [2, 22] The speed achieved of 30-75MHz seems very reasonable. These designs were built with focus on the novel idea of utilizing last passage time for the increase in phase noise. The frequency was kept around the same value for each design so they could be compared with each other. Also power consumption was not considered for this work.

6.1 Future work

Design 3 was not prepared in time for fabrication and thus for direct comparison of results with Design 1 and 2. Theoretically, Design 3 should provide vast improvements in the jitter production, as simulations showed substantial increase in performance. Applying Design 3 on a chip would therefore be a worthwhile endeavour. The design would be similar in size to the original Design 2.

References

[1] L. Chen and G. Gong, *Communication System Security.* Boca Raton, Florida: CRC Press, first ed., 2012.

[2] C. S. Petrie and J. A. Connelly, "A noise-based ic random number generator for applications in cryptography," *IEEE Transactions On Circuits And Systems I: Fundamental Theory And Applications*, vol. 47, no. 5, pp. 615–621, 2000.

[3] C. S. Petrie and J. A. Connelly, "Modelling and simulation of oscillator based random number generators," in *IEEE International Symposium on Circuits and Systems*, vol. 4, pp. 324–327, 1996.

[4] R. C. Fairfield, R. L. Mortenson, and K. B. Coulthart, "An lsi random number generator," in *Proc. Advances in Cryptology Conf.*, pp. 203–230, 1984.

[5] S. K. Mathew, S. Srinivasan, and M. A. Anders, "2.4 gbps, 7 mw all-digital pvt-variation tolerant true random number generator for 45 nm cmos high-performance microprocessors," *IEEE Journal of Solid-State Circuits*, vol. 47, no. 11, pp. 2807–2821, 2012.

[6] N. I. of Standards and Technology, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," Tech. Rep. 800-22, NIST Special Publication, 2010.

[7] N. I. of Standards and Technology, "Security requirements for cryptographic modules," Tech. Rep. 140-1, FIPS Publication, 1994.

[8] A. Demir, A. Mehrotra, and J. Roychowdhury, "Phase noise in oscillators: A unifying theory and numerical methods for characterization," *IEEE Transactions On Circuits And SystemsI: Fundamental Theory And Applications*, vol. 47, no. 5, pp. 655–674, 2000.

[9] A. A. Abidi, "Phase noise and jitter in cmos ring oscillators," *IEEE Journal of Solid-State Circuits*, vol. 41, no. 8, pp. 1803–1806, 2006.

[10] L. H., "A novel model on phase noise of ring oscillator based on last passage time," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 3, pp. 471–482, 2004.

[11] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, *Analysis and Design of Analog Integrated Circuits*. Toronto, Canada: Wiley, fifth ed., 2009.

[12] B. Leung, D. McLeish, and S. Robson, "Novel last passage time based jitter model with application to low slew rate/high noise ring oscillator," Tech. Rep. 2013-2, University of Waterloo Electrical and Computer Engineering, 2013.

[13] "Eldo user manual," Tech. Rep. AMS 2008.2, Mentor Graphics Corporation, 2008.

[14] B. Leung, *VLCI for Wirless Communication*. New York, New York: Springer, second ed., 2011.

[15] A. G. Bluman, *Elementary Statistics - A Step by Step Approach*. New York, New York: McGraw Hill, fourth ed., 2008.

[16] J. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits - A Design Perspective*. Upper Saddle River, New Jersey: Prentice-Hall, second ed., 2003.

[17] V. Stojanovic and V. G. Oklobdzija, "Comparative analysis of masterslave latches and flip-flops for high-performance and low-power systems," *IEEE Journal of Solid-State Circuits*, vol. 34, no. 4, pp. 536–548, 1999.

[18] M. Matsui, H. Hara, Y. Uetani, and K. Lee-Sup, "A 200 mhz 13 mm2 2-d dct macrocell using sense-amplifying pipeline flip-flop scheme," *IEEE Journal of Solid-State Circuits*, vol. 29, no. 12, pp. 1482–1490, 1994.

[19] "Cmos8rf design manual," tech. rep., IBM Corporation, November 2010.

[20] "Cmos8rf esd reference guide," tech. rep., IBM Corporation, April 2005.

[21] Tektronix, "Jitter and eye-diagram analysis tools - online help," July 2012.

[22] M. Bucci, L. Germani, R. Luzzi, A. Trifiletti, and M. Varanonuovo, “A high-speed oscillator-based truly random number source for cryptographic applications on a smart card ic,” *IEEE Transactions On Computers*, vol. 52, no. 4, pp. 403–409, 2003.