PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA

DISEÑO DE UN GENERADOR DE NÚMEROS ALEATORIOS PARA APLICACIONES DE CRIPTOGRAFÍA EN TARJETAS INTELIGENTES

Tesis pata optar el Título de Ingeniero Electrónico, que presenta el bachiller:

Eduardo Alberto Martin Bejar Espejo

ASESORES: Julio César Saldaña Pumarica

Erick Raygada Vargas

Lima, 2015

Resumen

La generación de números aleatorios es un punto clave en los sistemas criptográficos, su desempeño depende del nivel de aleatoriedad que son capaces de generar. Particularmente, en aplicaciones móviles estos generadores de números aleatorios están sujetos a fuertes restricciones a nivel de diseño de circuito integrado. En la presente tesis se realizó el diseño y simulación de un circuito generador de números aleatorios en tecnología CMOS $0.35\mu m$ para el procesador criptográfico de una tarjeta inteligente (*Smart Card*). El método de generación consiste en el muestreo de un oscilador con *jitter* elevado, el cual permite dividir al circuito en tres bloques principales. El primero de ellos es el oscilador que fija la frecuencia de muestreo cuyo periodo debe ser mucho más pequeño, en promedio, que el del oscilador con *jitter* elevado. El segundo bloque consiste en el circuito muestreador, implementado mediante un *flip flop* tipo T. El tercer bloque es el oscilador afectado por *jitter* del cual depende, en gran medida, la calidad de los números aleatorios generados. Este consiste en un oscilador triangular donde el ruido térmico, introducido por un par de resistencias, es amplificado. Estos tres bloques, trabajando de manera conjunta, generan los números aleatorios cuya calidad se analizó mediante los algoritmos propuestos por el *National Institute of Standards and Technology* (NIST) para verificar si el generador es lo suficientemente aleatorio como para ser utilizado en aplicaciones criptográficas.

La estructura del presente documento se detalla a continuación. En el primer capítulo se definió el problema a resolver. En el segundo capítulo, se revisaron los conceptos teóricos fundamentales relacionados a los números aleatorios y tecnología CMOS, asimismo, se presentaron diferentes metodologías actuales de generación de números aleatorios en circuitos integrados. En el tercer capítulo, se analizó con detalle la topología a usar y se realizó su diseño respectivo. En el cuarto capítulo se hicieron las simulaciones necesarias para verificar el correcto funcionamiento del circuito y se analizaron las secuencias de números obtenidas usando los algoritmos propuestos por el NIST. Finalmente, se presentan las conclusiones y recomendaciones.

Índice

Índice de figuras

Índice de tablas

Introducción

El creciente uso de sistemas de identificación por radio frecuencia (RFID) genera la necesidad de garantizar la seguridad de la información involucrada. La comunicación entre lectores RFID y tarjetas inteligentes puede realizarse de manera segura gracias a la implementación de algoritmos de criptografía. Estos algoritmos utilizan llaves criptográficas secretas, las cuales deben ser irreproducibles e impredecibles. Dado que las llaves secretas son creadas por un generador de números aleatorios, la seguridad de la comunicación dependerá de la calidad de dicho generador. Existen dos grandes grupos de generadores de números aleatorios, los PRNG (Pseudo Random Number Generators) y los TRNG (True Random Number Generators), siendo estos últimos los preferidos por el grado de aleatoriedad alcanzado. Los TRNG son implementados a nivel de hardware y pueden estar basados en la amplificación directa de ruido térmico, en el muestreo de la señal de un oscilador afectado por ruido o en la generación de caos en tiempo discreto. El presente trabajo de tesis muestra el diseño, en tecnología CMOS $0.35\mu m$, de un generador de números aleatorios basado en la segunda alternativa.

En el primer capítulo se describe el problema a resolver y se plantean los objetivos que se buscan cumplir a lo largo del trabajo. También, se muestran las limitaciones que se presentarán en su realización.

En el segundo capítulo se hace una revisión de los conceptos matemáticos y físicos fundamentales necesarios para el desarrollo de la tesis. Asimismo, se presentan las tendencias actuales de generación de números aleatorios en circuitos integrados.

En el tercer capítulo se hace un análisis detallado del método a usar para la generación de números aleatorios y se encuentran los parámetros circuitales de los que dependen los distintos parámetros aleatorios. Luego se realiza el diseño del circuito completo, esto abarca el planteamiento de las especificaciones del circuito, la elección de las topologías de los distintos bloques a usar, el dimensionamiento de los transistores y el *layout* del circuito. Esto, junto con las simulaciones posteriores, será posible gracias a la herramienta CAD de diseño de circuitos integrados llamada *CADENCE*, ampliamente usada actualmente en la industria, la cual se encuentra disponible junto con la tecnología AMS $0.35\mu m$ en el laboratorio de

microelectrónica de la PUCP.

En el cuarto capítulo se realizan las simulaciones necesarias para verificar que el circuito completo cumple con las especificaciones que se plantearon inicialmente. Asimismo, se evalúa la calidad del generador aleatorio diseñado mediante los algoritmos propuestos por el *National Institute of Standards and Technology* (NIST).

Capítulo 1

Problemática

1.1 Descripción y formulación del problema

La generación de números aleatorios ha sido un problema que la tecnología ha tratado de resolver durante varias décadas. Actualmente, se acepta la existencia de dos clases de números aleatorios. Por un lado están los números pseudoaleatorios (PRN) que son generados por algoritmos matemáticos dada una condición inicial, conocida como semilla, y por otro lado están los verdaderos números aleatorios (TRN) los cuales provienen de procesos físicos aleatorios de la naturaleza. El problema con los PRN es que al venir de algoritmos matemáticos estos están siendo generados de manera determinista, si la semilla es conocida se puede conocer toda la secuencia, por lo cual no se pueden llamar números aleatorios en el sentido estricto. A pesar de esto, en muchas aplicaciones tales como el muestreo aleatorio en cadenas de información muy grandes o un simple reproductor de música en modo aleatorio estos números son más que suficientes; sin embargo, en aplicaciones orientadas a la seguridad y privacidad de la información se require de un nivel de aleatoriedad mayor [6]. Los TRN al venir de procesos físicos, considerados aleatorios, son completamente impredecibles. Dada una muestra de este proceso es imposible saber con certeza el siguiente valor. Algunos de estos procesos físicos aleatorios mencionados son el ruido atmosférico, térmico y cuántico. Un TRNG consta de tres partes [7]. La primera consiste en la fuente de aleatoriedad o entropía, que es determinada principalmente por el ruido térmico y de disparo en el caso particular de circuitos integrados. La segunda parte consiste en un mecanismo de recolección de aleatoriedad, que debe garantizar la perturbación mínima de la aleatoriedad dada por la fuente. La tercera parte consiste en una etapa de post procesamiento que corrige imperfecciones, ya sea de la fuente de entropía o mecanismo de recolección de aleatoriedad, o bien puede brindar protección frente a cambios en el entorno. Un ejemplo conocido de este bloque es el corrector de Von Neuman [6]. Un

punto importante en lo que respecta a los números aleatorios en general, es el hecho de cómo medir su aleatoriedad. Para esta tarea se pueden encontrar *tests*, que son básicamente una serie de algoritmos, que reciben como entrada las secuencias aleatorias y en base a un análisis estadístico tratan de evaluar su aleatoriedad. Algunos *tests* conocidos son los presentados por el NIST y *DieHard*. Actualmente, una aplicación específica donde se necesitan TRNG es en los procesadores criptográficos de las tarjetas inteligentes, mejor conocidas como *Smart Cards*, las cuales se están usando cada vez más como mecanismos de identificación o de pago. Las tarjetas inteligentes, al ser de dimensiones pequeñas y de carecer de una fuente de alimentación permanente, imponen fuertes restricciones de área y potencia al TRNG. La presente tesis se enfoca en desarrollar un generador de números aleatorios que pueda ser usado en el procesador criptográfico de una tarjeta inteligente como parte de la generación de semillas aleatorias para un generador PRN, su calidad aleatoria será evaluada usando los algoritmos propuestos por el NIST.

1.2 Objetivos

1.2.1 Objetivos Principales

- Diseñar un generador de números aleatorios en tecnología CMOS $0.35\mu m$.
- Evaluar la calidad aleatoria del generador diseñado usando los algoritmos propuestos por el NIST.

1.2.2 Objetivos Secundarios

- Estudiar un modelo actual usado para el transistor MOS.
- Estudiar bloques analógicos y digitales utilizados en circuitos CMOS.
- Manejar una herramienta CAD para el diseño y simulación de circuitos integrados usada actualmente en la industria.
- Elaborar el *layout* del circuito diseñado.

1.3 Importancia y justificación del estudio

Los chips de las tarjetas inteligentes poseen procesadores criptográficos, los cuales ejecutan diversas tareas, en donde la generación de llaves criptográficas desempeña un papel esencial en la seguridad. Esas llaves son generadas a partir de números pseudoaleatorios creados mediante

un algoritmo matemático. Los algoritmos para la generación de números pseudoaleatorios utilizan unas semillas que deben ser creadas a partir de algún fenómeno físico aleatorio como el ruido térmico de un resistor. La invulnerabilidad de los algoritmos criptográficos depende en gran medida de la aleatoriedad de las semillas utilizadas. Una buena criptografía requiere de números aleatorios de buena calidad y para lograrlo el TRNG debe satisfacer un requisito importante, así se conozca el diseño del TRNG no se pueden predecir los valores a su salida. Finalmente, es importante mencionar que el presente trabajo también pretende poner en evidencia parte del proceso de diseño de circuitos integrados CMOS utilizado actualmente. La razón por la cual no se completa el proceso es por la imposibilidad de mandar a fabricar un circuito integrado. La figura 1.1 muestra el proceso completo que se ha mencionado, resaltando en rojo las partes específicas que se realizarán en la tesis.

1.4 Limitaciones

- Actualmente mucha de la información sobre la generación de TRN es confidencial, por lo cual son muy pocos los métodos que se encuentran publicados y de los cuales se puedan partir en el estudio.
- Puesto que no se cuenta con la posibilidad de fabricación del chip del circuito diseñado los resultados no podrán ser verificados de manera real.
- Al utilizar un simulador para la verificación del funcionamiento del circuito, este se verá limitado por qué tan bien el simulador utilizado genere números aleatorios, con lo cual el hecho que no se logren pasar algunos tests de aleatoriedad no será un indicativo concluyente acerca de la calidad del generador diseñado.
- La generación de bits usando un simulador tarda demasiado tiempo, pues se están usando dos osciladores donde la frecuencia de uno es mucho mayor que la del otro. Por esta razón, no se podrán alcanzar cantidades de bits necesarias que tengan un suficiente significado estadístico como para evaluar la calidad del generador diseñado.

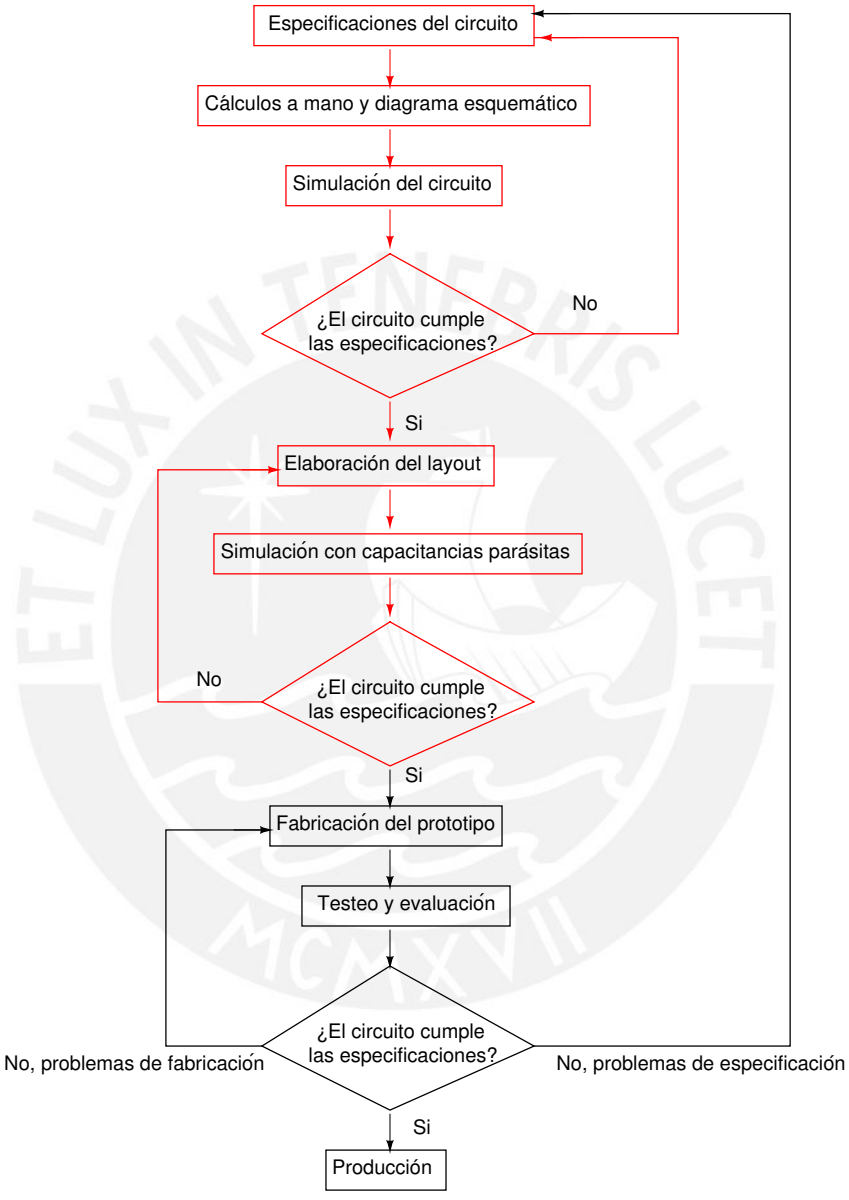

Figura 1.1: Proceso de diseño en circuitos integrados CMOS [1].

Capítulo 2

Métodos utilizados para la generación de números aleatorios en circuitos integrados

2.1 Conceptos matemáticos fundamentales

2.1.1 Variables aleatorias

Dado un experimento en el cual sus posibles resultados forman un conjunto S, una variable aleatoria es aquella que mapea este conjunto S al conjunto de los números reales, además cada posible resultado del experimento tiene asociado un número real entre 0 y 1, llamado probabilidad, que indica el nivel de certeza para la ocurrencia del resultado. De esta manera, es de suma importancia saber cómo se reparte la probabilidad en los diferentes valores que puede tomar la variable aleatoria mediante la llamada función de densidad de probabilidad, la cual será discreta o continua dependiendo del posible conjunto de valores que se puedan tomar.

2.1.1.1 Distribución de probabilidad gaussiana

Se caracteriza por tener una función de densidad de probabilidad gaussiana:

$$f_X(X = x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{x-\mu}{\sigma})^2} \tag{2.1}$$

Donde μ y σ^2 son el valor esperado y varianza, respectivamente, de la variable aleatoria.

2.1.1.2 Distribución de probabilidad uniforme

La función de probabilidad es constante en este caso. Si la variable aleatoria toma valores continuos entre x_1 y x_2, se tendrá:

$$f_X(X = x) = \frac{1}{x_2 - x_1}, x_1 \leq x \leq x_2$$

Si la variable aleatoria toma n valores discretos $x_1, x_2, ..., x_n$ se tendrá:

$$f_X(X = x_i) = \frac{1}{n}, 1 \leq i \leq n$$

2.1.2 Procesos aleatorios

Un proceso aleatorio se puede considerar como una secuencia de variables aleatorias $X_t, t \in T$, donde t puede representar al tiempo. Si T es un conjunto discreto, se tendrá un proceso aleatorio en tiempo discreto, si T es un conjunto continuo, se tendrá un proceso aleatorio en tiempo continuo. El proceso aleatorio de interés para este trabajo es el proceso gaussiano.

2.1.2.1 Proceso aleatorio gaussiano

Es un proceso en el cual el valor que se toma en cada instante de tiempo corresponde a una variable aleatoria con densidad de probabilidad gaussiana.

2.1.3 Densidad espectral de potencia

Es la herramienta que permite el análisis en frecuencia de señales aleatorias. Así como la aplicación directa de la transformada de Fourier permite conocer el contenido frecuencial de una señal determinista, con señales aleatorias se puede obtener un resultado similar. Para esto se tiene como condición que el proceso sea débilmente estacionario (WSS), luego se define la densidad espectral de potencia del proceso como:

$$S_{x(t)}(jw) = F\{r_x(t)\}$$

Donde $r_x(t)$ es la función de autocorrelación del proceso. Notar que se define una densidad espectral de potencia pero no de energía, pues para obtener el modelo probabilístico del proceso en cuestión es necesario observarlo durante periodos muy largos de tiempo, es decir, debe tener energía infinita [8].

Una relación de mucha utilidad para simplificar cálculos en el presente trabajo es la siguiente. Si se tiene un proceso aleatorio formado por una combinación lineal de varios procesos aleatorios débilmente estacionarios, los cuales son independientes (ortogonales) entre sí, el espectro resultante es la suma de los espectros con los coeficientes elevados al cuadrado. Es decir,

$$z(t) = \sum_{i=1}^{n} \alpha_i x_i(t), x_{i(t)} \perp x_{j(t)} \forall i \neq j, \forall t \Rightarrow S_{z(t)}(f) = \sum_{i=1}^{n} {\alpha_i}^2 S_{x_i(t)}(f) \tag{2.2}$$

2.2 Conceptos físicos fundamentales

2.2.1 Tecnología CMOS

Esta tecnología permite obtener los dispositivos NMOS y PMOS en el mismo chip mediante procesos físicos y químicos complejos como fotolitografía, crecimiento y deposición de óxido, etc. Los circuitos son fabricados en estructuras de silicio circulares, conocidas como *wafers*, que se encuentran dividas en varios chips o *die*, los cuales deben mantener las mismas propiedades físicas entre sí [9] [10] [1].

2.2.2 Modelamiento del transistor MOS

En el diseño de circuitos integrados analógicos es muy importante tener un conjunto de ecuaciones matemáticas que representen el funcionamiento de los dispositivos de los cuales está conformado el circuito. Tener un buen modelo le permite al diseñador predecir e intuir el comportamiento del circuito. A diferencia del diseño digital, se requiere que el modelo sea preciso a lo largo de un rango de operación y no solo en dos estados. Particularmente en tecnología CMOS, el dispositivo de interés principal a modelar es el transistor MOS. Actualmente, son populares varios modelos del transistor MOS como el modelo de carga laminar [11], EKV [12], ACM [13] y BSIM. En la presente tesis se trabajará con el modelo de carga laminar ya que una aproximación simple en este modelo conlleva a unas ecuaciones muy populares y usadas en el diseño analógico.

2.2.2.1 Modelo de carga laminar

El modelo de carga laminar ha sido rigurosamente desarrollado en [11], en esta sección solo se mencionarán puntos importantes para la tesis. El modelo para transistores de canal n, que proporciona las características corriente-voltaje ($I - V$) en el transistor MOS, en su expresión más reducida y usada, es el siguiente:

$$I_{DS} = \frac{1}{2}K_n\frac{W}{L}(V_{GS} - V_{Thn})^2, V_{DS} > V_{DSsat}, V_{GS} > V_{Thn} \tag{2.3}$$

$$I_{DS} = K_n\frac{W}{L}((V_{GS} - V_{Thn})V_{DS} - \frac{V_{DS}^2}{2}), V_{DS} < V_{DSsat}, V_{GS} > V_{Thn} \tag{2.4}$$

Y para canal p:

$$I_{SD} = \frac{1}{2}K_p\frac{W}{L}(V_{SG} - |V_{Thp}|)^2, V_{SD} > V_{SDsat}, V_{SG} > |V_{Thp}| \tag{2.5}$$

$$I_{SD} = K_p\frac{W}{L}((V_{SG} - |V_{Thp}|)V_{DS} - \frac{V_{SD}^2}{2}), V_{SD} < V_{SDsat}, V_{SG} > |V_{Thp}| \tag{2.6}$$

En donde los términos V_{Thn}, V_{Thp}, K_n y K_p están descritos en la tabla 2.1; W y L son el ancho y largo del canal del transistor MOS respectivamente; los subíndices G, D y S hacen referencia a los terminales *gate*, *drain* y *source* del transistor; y V_{DSsat} y V_{SDsat} son los voltajes de saturación de canal del dispositivo [8]. Asimismo, en estos modelos se aprecian dos regiones de operación posibles cuando $V_{GS} > V_{Thn}$, si $V_{DS} > V_{DSsat}$ se dice que el transistor opera en la zona de saturación y si $V_{DS} < V_{DSsat}$ se dice que se opera en la zona lineal óhmica o de triodo, si $V_{GS} < V_{Thn}$ se tiene $I_{DS} = 0$ y se dice que el transistor está en corte. Para el transistor PMOS las relaciones son similares. Es importante mencionar que para transistores MOS de canal largo se cumple $V_{DS} = V_{GS} - V_{Thn}$ y $V_{SD} = V_{SG} - V_{Thp}$.

En muchas ocasiones estas ecuaciones no son suficientes, pues no permiten modelar la resistencia de salida del transistor. Para esto hay que multiplicar a las ecuaciones 2.3 y 2.4 por el término $(1 + \lambda V_{DS})$ y a las ecuaciones 2.5 y 2.6 por $(1 + \lambda V_{SD})$. Hay que mencionar que el término λ no es un parámetro de proceso que los fabricantes proporcionen, al ser un parámetro que aparece en un modelo hay que realizar una extracción de parámetros para estimarlo [14]. En la tabla 2.1 se muestran los parámetros de la tecnología a usar. También, la alimentación del circuito permitida para la tecnología es 3.3V.

Parámetro	Símbolo	Mín.	Típ.	Máx.	Unidades
Voltaje Umbral NMOS	V_{Thn}	0.36	0.46	0.56	V
Voltaje Umbral PMOS	V_{Thp}	-0.58	-0.68	-0.78	V
Factor de Ganancia NMOS	K_n	150	170	190	$\frac{\mu A}{V^2}$
Factor de Ganancia PMOS	K_p	48	58	68	$\frac{\mu A}{V^2}$

Tabla 2.1: Parámetros importantes del proceso CMOS utilizado.

2.2.3 Ruido

Son perturbaciones sobre una señal y es modelado como un proceso aleatorio débilmente estacionario. En general, el ruido puede actuar de forma aditiva o multiplicativa sobre la señal de interés. En los circuitos integrados y en particular para la tecnología con la cual se está trabajando los ruidos más significativos presentes son el ruido térmico y el ruido *flicker*, los cuales actúan de forma aditiva sobre la señal de interés.

2.2.3.1 Ruido blanco

Es aquel en el que las variables aleatorias, en instantes de tiempos diferentes, no están correlacionadas, es decir:

$$E\{x(t_1)x(t_2)\} = 0, \forall t_1 \neq t_2$$

Esto se puede expresar en su función de autocorrelación como:

$$r_{x(t)}(\tau) = \sigma^2\delta(\tau)$$

Donde $\delta(\tau)$ es la función delta de Dirac y σ es la varianza del proceso en cualquier instante de tiempo.

Usando esto último, se puede encontrar el espectro de potencia del ruido blanco por medio de la transformada inversa de Fourier:

$$S_{x(t)}(jw) = \sigma^2, \forall w$$

Es decir, las frecuencias por las cuales está compuesto el ruido blanco aportan la misma cantidad de potencia.

2.2.3.2 Ruido térmico

Se debe al movimiento aleatorio de los electrones en un material debido a la temperatura. La primera fuente de ruido térmico en un circuito integrado son los resistores y se modela usualmente como una fuente de voltaje en serie con el resistor. Usando el teorema de Norton, siempre se puede transformar la fuente de voltaje en una fuente de corriente en paralelo con el respectivo resistor. En cualquier caso, para modelar el ruido térmico, los siguientes circuitos son equivalentes:

Donde $v(t)$ y $i(t)$ son procesos aleatorios. El espectro de potencia de $v(t)$ viene dado por:

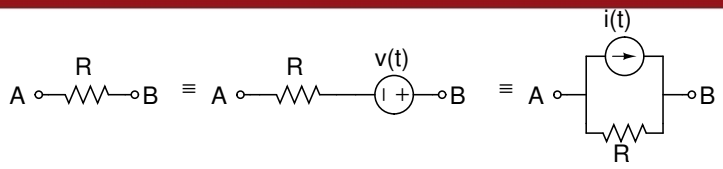

Figura 2.1: Ruido térmico de un resistor.

$$S_{v(t)}(f) = 4k_B T R, f \geq 0 \tag{2.7}$$

Y el espectro de potencia de $i(t)$ es:

$$S_{i(t)}(f) = \frac{4k_B T}{R}, f \geq 0$$

Donde $k_B = 1.38\text{x}10^{-23}$ es la constante de Boltzmann, T es la temperatura y R el valor de la resistencia.

La segunda fuente de ruido térmico en circuito integrados viene del generado en el canal de un transistor MOS operando en saturación. En el caso de transistores de canal largo, este ruido se modela usualmente como una fuente de corriente en paralelo con el canal del transistor. Usando equivalencia de circuitos se puede referir la señal de corriente a la compuerta del transistor en forma de voltaje, obteniendo los siguientes circuitos equivalentes. Los detalles se pueden encontrar en [8].

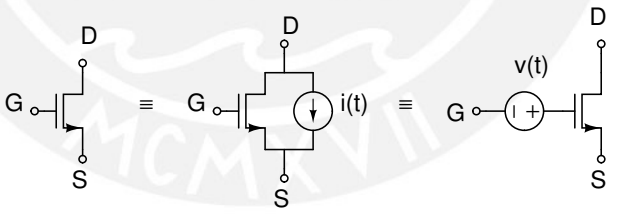

Figura 2.2: Ruido térmico en un transistor MOS.

Los espectros de potencia de los procesos aleatorios $v(t)$ y $i(t)$ mostrados son:

$$S_{i(t)}(f) = 4k_B T \gamma g_m, f \geq 0$$

$$S_{v(t)}(f) = \frac{4k_B T \gamma}{g_m}, f \geq 0$$

Donde el factor γ, que se encuentra actualmente en investigación, usualmente se considera $\gamma = \frac{2}{3}$ para transistores de canal largo [11].

Es importante mencionar, que en ambos casos este ruido térmico no es completamente blanco, es decir, el espectro de potencia de los procesos anteriores no es constante para todas las frecuencias, en la práctica, el ruido deja de ser blanco alrededor de los 100 THz [8].

2.2.3.3 Ruido *flicker*

También llamado ruido $1/f$, su origen se encuentra en la interfaz óxido - silicio de los transistores MOS. Los electrones que pasan por esta interfaz pueden quedar atrapados durante periodos de tiempo considerables, en el orden de los segundos, y luego siguen su recorrido. Por esta razón, este ruido es más fuerte a bajas frecuencias [15]. El ruido *flicker* se modela al igual que en la figura 2.2. Sin embargo, los espectros de potencia ahora son:

$$S_{v(t)}(f) = \frac{K}{C_{ox}WLf}, f \geq 0$$

$$S_{i(t)}(f) = \frac{Kg_m^2}{C_{ox}WLf}, f \geq 0$$

2.3 Topologías actuales para generación de números aleatorios

2.3.1 Amplificación directa de ruido térmico

La figura 2.3 representa la topología básica de esta metodología. Consiste en la amplificación de ruido térmico generado por una resistencia y una posterior comparación con un voltaje referencial, en el caso de la figura 2.3 se usa la tierra como referencia. Como el ruido térmico puede ser modelado como un proceso gaussiano, en cada instante de tiempo se tendrá una variable gaussiana, luego en el circuito mostrado debería ser igualmente probable tener un 1 o 0 a la salida, obteniendo una secuencia uniformemente distribuida a la salida. El principal problema de este tipo de aplicaciones es que, debido al ancho de banda limitado del amplificador, este deteriora la calidad aleatoria de la señal de salida. Además, este tipo de topología es fuertemente afectada por variaciones de la fuente de alimentación y por el ruido *flicker* de los transistores usados.

2.3.2 Muestreo de un oscilador afectado por *jitter*

En este caso se usa una señal de reloj de baja frecuencia para muestrear a una señal de alta frecuencia. Este método se basa en el hecho que cualquier oscilador no oscila a una frecuencia exacta, sino que presenta una variación entre periodo y periodo, también conocida como *jitter*,

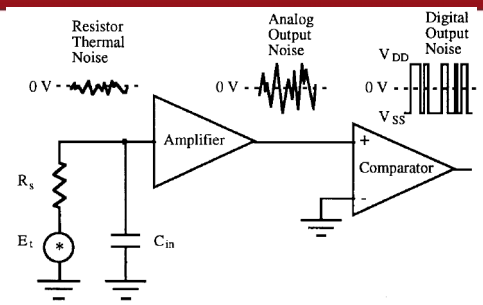

Figura 2.3: Generación de ruido térmico generado por un resistor propuesto en [2] y [3].

debido a imperfecciones en los procesos de fabricación y ruido introducido por los diferentes componentes de los cuales está formado el propio oscilador. Con esta consideración, se diseña el oscilador de baja frecuencia para que tenga una gran variación en su frecuencia de oscilación en comparación con la del oscilador de alta frecuencia. Si estos requisitos son cumplidos, la calidad de la señal aleatoria será buena y mejorará mientras mayor sea la variación del oscilador de baja frecuencia.

En la figura 2.4 se puede ver como un oscilador afectado por *jitter* es muestreado por un *flip flop* tipo T. Cumpliendo ciertas condiciones, la salida del muestreador será lo suficientemente aleatoria. En la figura mostrada, se utiliza un post procesador para hacer la secuencia muestreada más aleatoria mediante algoritmos matemáticos.

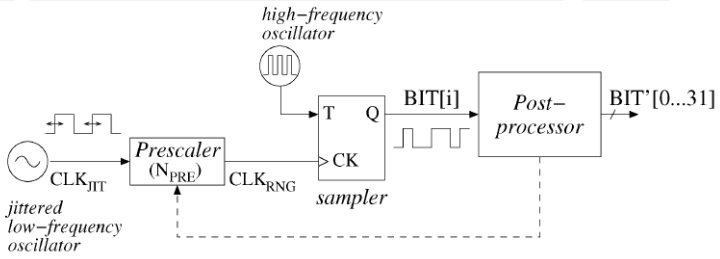

Figura 2.4: Muestreo de un oscilador afectado por *jitter* propuesto en [4].

A comparación del método anterior, este tipo de diseño no se ve tan afectado por el acoplo de señales deterministas ya que la información aleatoria se lleva en la fase del oscilador lento. Sin embargo, su problema radica en que los niveles de *jitter*, que se deben conseguir para tener una buena secuencia aleatoria, son difíciles de conseguir.

Esta topología es la elegida para diseñar en la presente tesis, su análisis se desarrollará con más énfasis en el siguiente capítulo.

2.3.3 Generación de caos en tiempo discreto

Se busca la generación de números aleatorios de manera similar a los PRNG solo que en la secuencia dada se considera la inclusión de un término aleatorio, proveniente de una señal aleatoria, que sigue cierta distribución probabilística. Este método, al igual que el anterior, es robusto frente a señales deterministas.

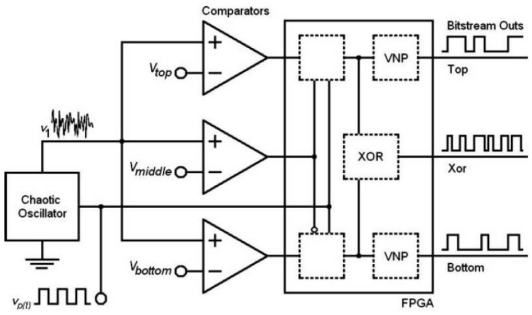

Figura 2.5: Circuito generador de caos presentado en [5].

Capítulo 3

Diseño del generador de números aleatorios

3.1 Análisis detallado de la topología propuesta

El método a usar en la presente tesis consiste en el muestreo de un oscilador afectado por *jitter*. Esta sección tiene como objetivo realizar el análisis detallado de la topología presentada en [4] con el fin de elaborar un modelo para el generador de números aleatorios y mostrar cómo sus parámetros estadísticos se relacionan con los parámetros del circuito. La generación de *jitter* se logrará amplificando el ruido térmico proveniente de una resistencia, esta señal es luego acoplada a un oscilador de baja frecuencia T_S, el cual será utilizado para muestrear a un oscilador de alta frecuencia T_F.

En este tipo de diseño, si el ciclo de trabajo (*duty cycle*) del oscilador rápido es de 50% y el oscilador lento presenta una variación entre periodo y periodo significativa frente al periodo del oscilador rápido, la probabilidad de encontrar un 1 o 0 a la entrada del circuito muestreador son iguales [4].

El problema es que, en la práctica, el *duty cycle* del oscilador rápido tendrá una variación entre dos valores d_{min} y d_{max}. Si se utilizan circuitos muestreadores como *sample and hold* o *flip flops* tipo D, el problema descrito origina que las probabilidades de obtener un 1 o 0 a la entrada del circuito muestreador ya no sean iguales, como se puede observar en la figura 3.1

Con este tipo de circuitos muestreadores, la probabilidad de obtener un 1 a la salida es exactamente igual a la de tener un 1 a la entrada del circuito de muestreo y además su valor viene dado por la siguiente expresión:

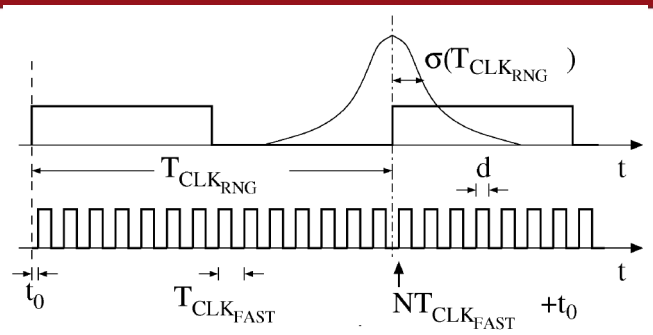

Figura 3.1: Ilustración del proceso de muestreo presentado en [4]. El muestreo se realiza con flanco de subida.

$$p(1) = \sum_{j=-\infty}^{+\infty} \int_{(N+j)T_F+t_0}^{(N+j+d)T_F+t_0} p(T_S)dT_S$$

Donde se está considerando lo siguiente:

- Un desfase $t_0 \in [0, T_F[$ permitido entre los flancos de subida (o bajada) inicial de los osciladores.
- Una variación del *duty cycle* del oscilador rápido $d_{min} \leq d \leq d_{max}$. Este rango no denota el rango de valores que puede tener el *duty cycle* por un efecto de *jitter*, sino indica el posible rango de valores que puede tener el oscilador de alta frecuencia. El efecto de *jitter* de este oscilador se desprecia frente al del oscilador de baja frecuencia en el diseño.
- El periodo del oscilador lento sigue una distribución de probabilidad gaussiana, esto es: $T_S \sim G(\mu_{T_S}, \sigma_{T_S})$.
- El valor esperado del periodo del oscilador contiene exactamente N veces al periodo del oscilador rápido, donde $N \in Z^+$. Esta condición se considera el peor caso posible, pues experimentalmente se muestra que los bits de salida generados son más aleatorios cuando este factor no es entero como se reporta en [4].

$$N = \frac{E\{T_S\}}{T_F}$$

Con estas consideraciones, ahora es más fácil ver a que se refiere cada límite de la integral. Por ejemplo, si $j = 0$ el límite inferior de la integral es $NT_F + t_0$ el cual representa el flanco de subida (o bajada) siguiente más próximo al flanco de subida esperado del oscilador lento como

se puede ver en 3.1. El límite superior $(N+d)T_F + t_0$ es simplemente el flanco de bajada (o subida) del periodo anterior.

Con $j > 0$ se consideran los flancos de subida (o bajada) siguientes al explicado y con $j < 0$, los anteriores. Como en las distribuciones gaussianas el 99% de la ocurrencia del evento se encuentra concentrada en el rango de valores $[\mu - 3\sigma, \mu + 3\sigma]$, los límites de la sumatoria se pueden limitar a $[-j_{max}, j_{max}]$, donde j_{max} es cualquier entero que cumpla:

$$(N+j)T_F + t_0 \geq E\{T_S\} + 3\sigma\{T_S\}$$

Como $E\{T_S\} = NT_F, 0 \geq t_0 < T_F$ y $T_F \ll T_S$, la desigualdad anterior se puede reducir a:

$$j_{max}T_F \geq 3\sigma T_S$$

Usando la función error complementario:

$$erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{+\infty} e^{-t^2} dt$$

Teniendo en cuenta que la distribucion gaussiana viene dada por la ecuación 2.1 se puede obtener, haciendo los cambios de variable necesarios, lo siguiente:

$$p(1) = \sum_{j=-j_{max}}^{j_{max}} \{\frac{1}{2} erfc(\frac{jT_F + t_0}{\sqrt{2}\sigma\{T_S\}}) - \frac{1}{2} erfc(\frac{(j+d)T_F + t_0}{\sqrt{2}\sigma\{T_S\}})\}$$

Lo importante que se debe notar de esta última expresión es que T_F y $\sigma\{T_S\}$ se pueden controlar mediante diseño como se verá más adelante. Las variables externas a este son precisamente d y t_0.

Notar que idealmente $p(1)_{Ideal} = 0.5$.

Asimismo, este valor será importante en el siguiente capítulo, pues permitirá verificar que el circuito se comporta de la manera explicada. Además, recibirá el nombre de probabilidad de transición, ya que si se reemplaza al FF-D por un FF-T, muestrear un 1 a la entrada significará un cambio de estado en la salida [4].

La figura 3.2 muestra el ploteo de esta función para algunos valores y rangos típicos reportados en [4]. Si el oscilador es rápido, se puede ver que mientras menor sea T_F el efecto de t_0 se puede despreciar y $p(1)$ tiende al valor del *duty cycle*. En la figura 3.3, se plotea la misma función pero con un oscilador 100 veces más lento, en este caso la probabilidad $p(1)$ depende del valor de t_0 y no se acerca al valor del *duty cycle*.

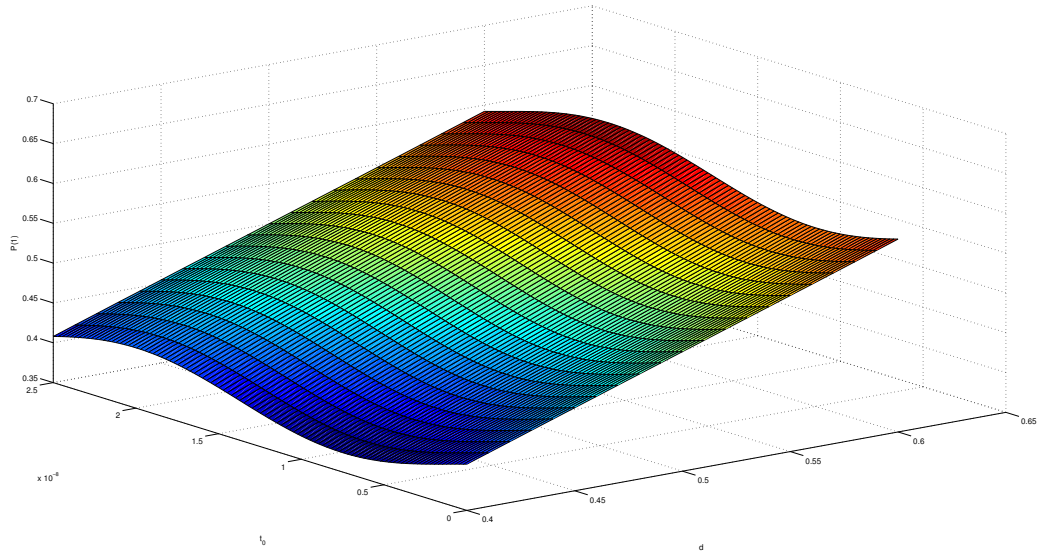

Figura 3.2: Probabilidad $p(1)$ con oscilador rápido, $T_F = 25ns$, $\sigma\{T_S\} = 10ns$.

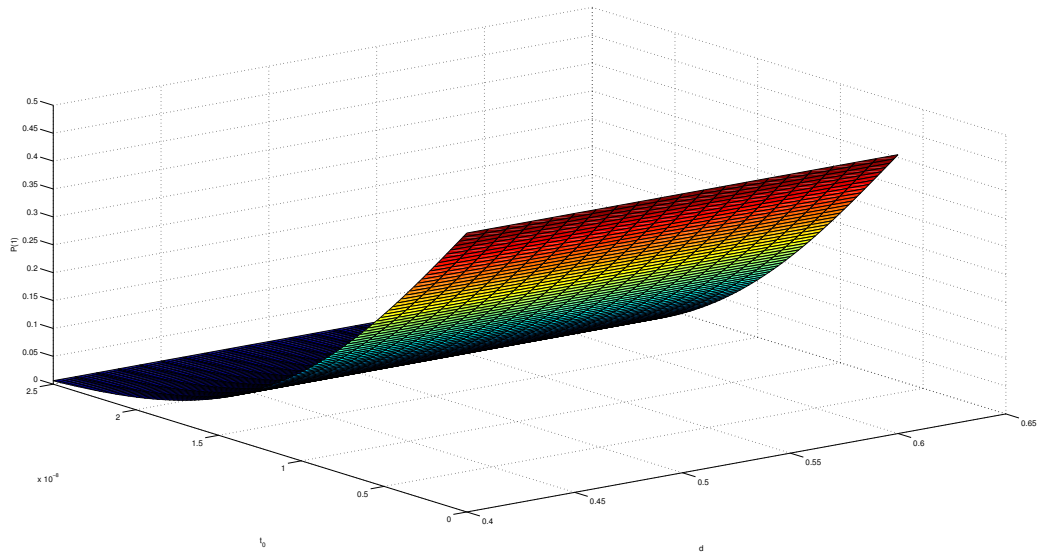

Figura 3.3: Probabilidad $p(1)$ con oscilador lento, $T_F = 2.5\mu s$, $\sigma\{T_S\} = 10ns$.

Para solucionar el problema del desbalance del *ducy cycle* es muy común utilizar filtros de paridad a la salida del circuito de muestreo, como parte de la etapa de post procesamiento digital, entre muchas otras soluciones. Sin embargo, en el presente trabajo se opta por utilizar un FF-T [16] en la etapa de muestreo, esto con el fin de disminuir el trabajo de las etapas de post procesamiento digital posteriores. El resultado de usar un FF-T es similar, pero no igual, al que se obtiene usando las técnicas mencionadas anteriormente, como se muestra a continuación:

La diferencia del muestreo con FF-T y FF-D, o similares, radica básicamente en que mientras el FF-D simplemente muestra en la salida lo que tiene en su entrada, la probabilidad de tener un 1 o 0 a la salida es igual a la de tener un 1 o 0 en la entrada, es decir, $p_{in}(1) = p(1) = p_n(1) = p_{out}(1)$. Donde $p(1)$ es fuertemente dependiente del *duty cycle* como se ha explicado anteriormente.

Por otro lado, en el FF-T la salida también es función del valor anterior, específicamente la ecuación para la probabilidad ahora es:

$$p_{n+1}(0) = p(0)p_n(0) + p(1)p_n(1)$$

$$p_{n+1}(1) = p(0)p_n(1) + p(1)p_n(0)$$

Donde:

- $p(0) = 1 - q$ y $p(1) = q$ son las probabilidades de obtener un 1 o 0 a la entrada del FF-T.
- $p_n(0)$ y $p_n(1)$ son las probabilidades de obtener un 1 o 0 a la salida del FF-T en el n-ésimo bit.

Como el espacio muestral de la variable aleatoria a la entrada y salida del FF-T es $S = \{0, 1\}$, se tiene:

$$p(0) + p(1) = 1$$
$$p_n(0) + p_n(1) = 1$$

Las expresiones anteriores ahora son:

$$p_{n+1}(0) = (1 - 2q)p_n(0) + q$$
$$p_{n+1}(1) = (1 - 2q)p_n(1) + q$$

Estas dos últimas ecuaciones de diferencia son conocidas, pues corresponden a un proceso

autoregresivo de primer orden, luego la solución viene dada por [17]:

$$p_{n+1}(0) = (1-2q)^{n+1} + \sum_{i=0}^{n} q(1-2q)^i = (1-2q)^{n+1} + \frac{1}{2}[1-(1-2q)^{n+1}]$$
$$p_{n+1}(1) = (1-2q)^{n+1} + \sum_{i=0}^{n} q(1-2q)^i = (1-2q)^{n+1} + \frac{1}{2}[1-(1-2q)^{n+1}]$$

Con esta última expresion es más sencillo ver que:

$$\lim_{n\to\infty} p_{n+1}(0) = \frac{1}{2}$$
$$\lim_{n\to\infty} p_{n+1}(1) = \frac{1}{2}$$

Con lo cual se demuestra que el FF-T elimina el problema del desbalance del *duty dycle* en el oscilador de alta frecuencia conforme se van tomando más muestras.

Asimismo, evaluando la función de autocorrelación de los procesos $p_{n+1}(0)$ y $p_{n+1}(1)$ se obtiene [17]:

$$R_{p_{n(0)}}(n, n+m) = \sigma^2_{p_0(0)}(1-2q)^m(1-2q)^{2n}$$
$$R_{p_{n(1)}}(n, n+m) = \sigma^2_{p_0(1)}(1-2q)^m(1-2q)^{2n}$$

De estas últimas expresiones se puede ver que mientras más separadas estén las muestras, menos correlacionadas estarán.

En conclusión, cuando se usa un método de muestreo tradicional, como el realizado con un FF-D, las probabilidades de obtener un 1 o 0 son básicamente funciones de valores determinísticos. Cuando se usa un método de muestreo más sofisticado, como el obtenido con un FF-T, este implementa una función de recurrencia. Específicamente, hace que las probabilidades mencionadas ahora sigan un proceso aleatorio autoregresivo de primer orden, las cuales se logran estabilizar conforme transcurre el tiempo a los valores ideales buscados.

El segundo problema que afecta la aleatoriedad del generador es el nivel de *jitter* presente en el oscilador lento. Si la desviación estándar de T_S es muy pequeña, se puede estimar fácilmente el promedio estadístico de $E\{T_S\}$, con lo cual teniendo un par de muestras se podría predecir el resto de la secuencia, desde el punto de vista matemático se diría que las muestras están correlacionadas. Para solucionar este problema, se busca hacer $\sigma\{T_S\}$ tan grande como sea posible, específicamente se toma como referencia al periodo del oscilador rápido T_F, con

lo cual la condición sería $\sigma\{T_S\} \ll T_F$, para el presente trabajo se entenderá esta desigualdad como $\sigma\{T_S\} \geq 10T_F$. Es importante notar que, como el ruido introducido por la resistencia es un ruido blanco gaussiano, conforme las muestras estén más alejadas, la correlación será menor.

Esto se puede entender de la siguiente manera. Si se modelan los instantes de muestreo como $T_M(n) = nT_S + \sum_{i=1}^{n} w(i)$, donde cada w es el ruido blanco agregado por la resistencia, se tendría que la varianza de la muestra m-ésima sería m veces mayor a la varianza de la primera muestra por una propiedad de variables gaussianas. Luego, se puede decir que en este tipo de muestreo, la varianza y, en consecuencia, la desviación estándar aumentan con el tiempo disminuyendo la correlación entre las diferentes muestras. Mientras más alejadas estén, menos correlacionadas estarán; por lo tanto, para tener una buena calidad de números aleatorios a la salida hay que asegurar que la desviación estándar del periodo del oscilador lento, conocido como *jitter*, sea tan grande como sea posible.

Como se ha mostrado, el nivel de *jitter* es importante, pues permite incrementar la calidad de los números aleatorios generados. Asimismo, es capaz de determinar la velocidad de bits a la salida del TRNG.

Por ejemplo, usando las consideraciones indicadas, se ha reportado que en procesos CMOS de $0.18\mu m$ [4], se obtienen los siguientes resultados para osciladores en anillo:

$$\frac{\sigma\{T_S\}}{E\{T_S\}} < 10^{-4}$$

Con lo cual teniendo en cuenta $\sigma\{T_S\} \geq 10T_F$ y usando un oscilador rápido de $T_F = 1\text{GHz}$, se tendría:

$$\frac{1}{E\{T_S\}} \approx \frac{1}{\overline{T_S}} < 10KHz$$

Como cada ciclo de reloj corresponde a un bit, se tendría que la velocidad a la salida estaría limitada a 10kbps.

Por lo tanto, es muy importante obtener niveles de *jitter* bastante elevados, incluso para no tener que lidiar con osciladores de tan alta frecuencia con todos los efectos que esto trae como la aparición de capacitancias e inductancias no deseadas o potencias reflejadas. En el presente trabajo se logrará esto amplificando la fuente de ruido térmico mediante un OPAMP.

Un punto relacionado a esto es saber qué tan rápido se pueden generar los bits en función del ancho de banda del OPAMP, pues siendo este el causante de limitar en gran parte la calidad aleatoria inicial, no se puede pretender generar aleatoriedad a una frecuencia mayor

a las componentes de ruido que quedan luego del filtrado. Tal límite mencionado ya ha sido revisado en [3] y muestra que la frecuencia de generación de bits aleatorios debe ser menor a una constante multiplicada por el ancho de banda del OPAMP, es decir $f_S < kBW_{OPAMP}$, donde k depende de la cantidad de bits que se necesitan, en particular como se menciona en [16], para sistemas RFID este límite resulta en $f_S < 1.66BW_{OPAMP}$.

Con estas consideraciones, se muestra el diagrama esquemático del circuito a implementar en la figura 3.4:

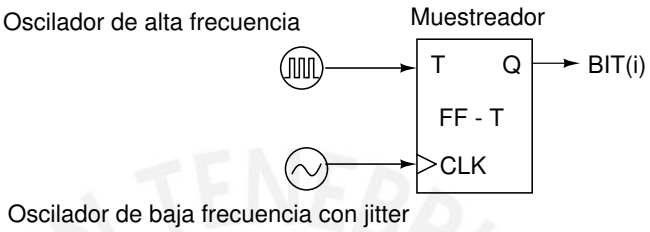

Figura 3.4: Esquemático del circuito a implementar.

De lo explicado anteriormente, es claro que la parte más crítica del diseño es el oscilador de baja frecuencia cuyo diagrama esquemático se muestra en la figura 3.5

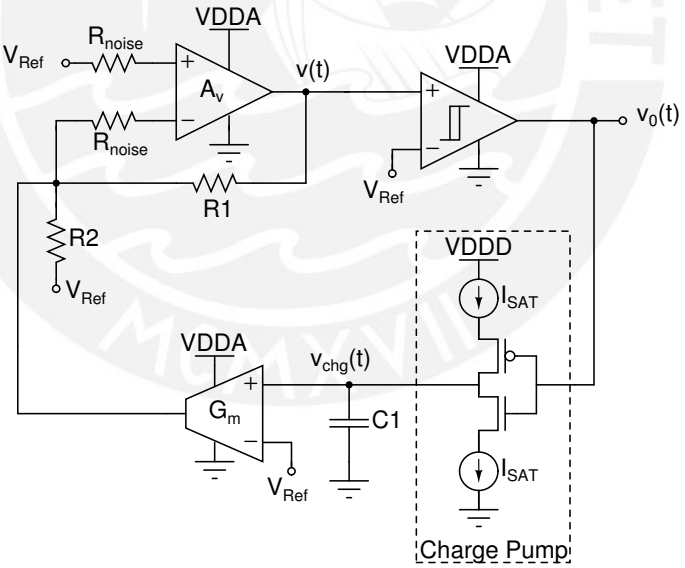

Figura 3.5: Esquemático del circuito oscilador de baja frecuencia.

El oscilador de baja frecuencia consta de:

- OPAMP: Amplifica el ruido térmico del par de resistencias R_{noise}. La salida de este será una onda triangular con el ruido térmico amplificado y acoplado.
- Comparador con histéresis: Realiza la conversión analógica - digital. Convierte la onda

triangular a una onda rectangular.

- *Charge Pump*: Realiza la conversión digital - analógica formando un lazo de realimentación.
- Condensador C_1: Carga del *charge pump*.
- Amplificador de transconductancia: Convierte la señal de voltaje proveniente del condensador en señal de corriente.
- Resistencias R_{noise}: Introducen el ruido térmico en el circuito. Se usan dos para compensar el voltaje de offset en el OPAMP.
- Resistencia R_1 y R_2: Permiten fijar la ganancia en lazo cerrado del OPAMP.

Ahora se obtendrán los parámetros para el oscilador lento:

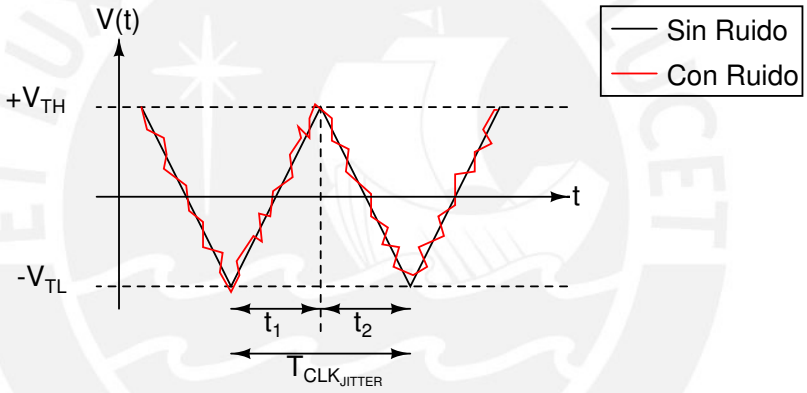

Figura 3.6: Salida $V(t)$ del oscilador lento.

Usando la figura 3.6, el periodo del oscilador lento viene dado por la suma de los intervalos t_1 y t_2:

$$T_S = t_1 + t_2$$

Donde t_1 y t_2 son también variables aleatorias que siguen una distribución gaussiana. No olvidar que el ruido agregado por las resistencias es blanco.

Se tendrá entonces:

$$E\{T_S\} = E\{t_1\} + E\{t_2\}$$
$$\sigma^2\{T_S\} = \sigma^2\{t_1\} + \sigma^2\{t_2\}$$

De la figura se puede ver que cada tramo creciente de la onda triangular se puede escribir:

$$V(t) = -V_{TL} + st + v_n(t)$$

Donde s es la pendiente en ausencia de ruido de la señal triangular.

Después de un intervalo de tiempo t_1 se tiene un cambio en la señal de salida del disparador. En este instante $v(t) = V_{TH}$. Luego:

$$-V_{TL} + st_1 + v_n(t_1) = V_{TH}$$
$$t_1 = \frac{V_{TH} + V_{TL} - v_n(t)}{s}$$

Evaluando los parámetros estadísticos de t_1 se tiene:

$$E\{t_1\} = -\frac{E\{v_n(t_1)\}}{s} = \frac{V_{TH} + V_{TL}}{s} - \frac{E\{v_n(t)\}}{s}$$
$$\sigma^2\{t_1\} = \frac{\sigma\{v_n(t_1)\}}{s^2} = \frac{\sigma\{v_n(t)\}}{s^2}$$

De manera similar para el tramo decreciente de la onda triangular se puede escribir:

$$V(t) = V_{TH} - st + v_n(t)$$

Ahora el cambio se produce en $V(t) = -V_{TL}$. Luego:

$$V_{TH} - st_2 + v_n(t_2) = -V_{TL}$$
$$t_2 = \frac{V_{TH} + V_{TL} + v_n(t_2)}{s}$$

Evaluando los parámetros estadísticos de t_2 se tiene:

$$E\{t_2\} = \frac{E\{t_1\}}{s} = \frac{V_{TH} + V_{TL}}{s} + \frac{E\{v_n(t)\}}{s}$$
$$\sigma^2\{t_2\} = \frac{\sigma^2\{v_n(t_2)\}}{s^2} = \frac{\sigma^2\{v_n(t)\}}{s^2}$$

Notar que:

$$E\{v_n(t)\} = E\{v_n(t_1)\} = E\{v_n(t_2)\}$$
$$\sigma^2\{v_n(t)\} = \sigma^2\{v_n(t_1)\} = \sigma^2\{v_n(t_2)\}$$

Pues el ruido térmico es blanco, gaussiano y fuertemente estacionario. Ahora los parámetros estadísticos de T_S se pueden obtener reemplazando en la expresión anterior.

$$E\{T_S\} = 2(\frac{V_{TH} + V_{TL}}{s}) \tag{3.1}$$

$$\sigma^2\{T_S\} = 2(\frac{\sigma^2\{v_n(t)\}}{s^2})$$

$$\sigma\{T_S\} = \sqrt{2}\frac{\sigma\{v_n(t)\}}{s} \tag{3.2}$$

Falta evaluar el parámetro s, para esto hay que considerar el modelo simplificado del circuito de la figura 3.7. Escribiendo las relaciones de entrada y salida de cada dispositivo, se tiene:

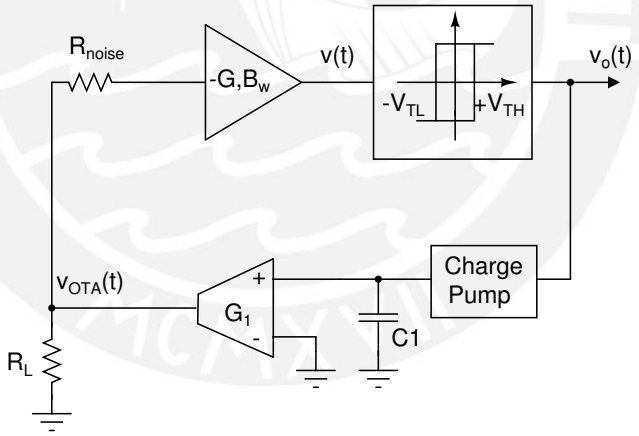

Figura 3.7: Esquemático del circuito general simplificado.

$$i_c(t) = C\frac{dv_c(t)}{dt}$$
$$i(t) = G_1 v_c(t)$$
$$v_{OTA} = R_L i(t)$$
$$V(t) = -Gv_{OTA}(t$$

Reemplazando se puede obtener:

$$V(t) = -GR_L G_1 v_c(t)$$

Derivando:

$$\frac{dV(t)}{dt} = -GR_L G_1 \frac{i_c(t)}{C} = \mp \frac{GR_L G_1 I_{SAT}}{C}$$

Como se ha definido $s > 0$, se tiene entonces $s = |\frac{dV(t)}{dt}|$

Luego:

$$s = \frac{GR_L G_1 I_{SAT}}{C} \tag{3.3}$$

Finalmente, para evaluar la desviación estándar del oscilador, se aplica la ley de corrientes de Kirchoff en el nodo X mostrado en la figura 3.8.

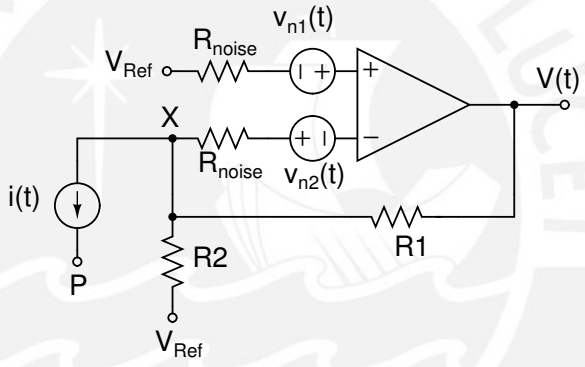

Figura 3.8: Extracción del ruido de las resistencias.

$$i(t) + \frac{v_{n1}(t) + v_{n2}(t)}{R_2} + \frac{V_{Ref} + v_{n1}(t) + v_{n2}(t) - V(t)}{R_1} = 0$$

Despejando el voltaje de salida $V(t)$:

$$V(t) = V_{Ref} + R_1 i(t) + (1 + \frac{R_1}{R_2})(v_{n1}(t) + v_{n2}(t)) \tag{3.4}$$

Definiendo $G = 1 + \frac{R_1}{R_2}$ y $v_N(t) = G(v_{n1}(t) + v_{n2}(t))$ y como $v_{n1}(t)$ y $v_{n2}(t)$ son procesos de ruido blanco independientes entre sí, usando las ecuaciones 2.2 y 2.7, se puede observar que:

$$S_{v_N(t)}(f) = G^2(S_{v_{n1}(t)}(f) + S_{v_{n2}(t)}(f))$$
$$S_{v_N(t)}(f) = 8k_BTR_{noise}G^2, f \geq 0$$

Notar que $v_N(t)$ es la componente de ruido térmico a la salida del circuito ya amplificada. Calculando su potencia promedio:

$$\overline{P}_{v_N(t)} = \sigma^2\{v_n(t)\} = 8k_BTR_{noise}G^2(B_W)$$

Y finalmente:

$$\sigma\{v_N(t)\} = \sqrt{8B_Wk_BTR_{noise}G^2} \tag{3.5}$$

Notar que las ecuaciones 3.1, 3.2 y 3.5 son las expresiones de interés, pues relacionan los parámetros estadísticos con los parámetros del circuito.

3.2 Elección de parámetros

Tomando como referencia [4] y [16], se han seleccionado las siguientes especificaciones para el generador de números aleatorios a diseñar.

Parámetro	Descripción	Valor
R_{noise}	Resistencias de ruido	30kΩ
G	Ganancia del OPAMP en lazo cerrado	48dB
Bw	Ancho de banda del OPAMP en lazo cerrado	40MHz
MF	Margen de fase del OPAMP	$\geq 30°$
R_L	Carga del OTA	400Ω
G_1	Transconductancia del OTA	$10\frac{\mu A}{V}$
C_1	Capacitor de carga del *charge pump*	10pF
I_{SAT}	Corriente de salida del *charge pump*	500nA
$V_{Histéresis}$	$V_{TH} - V_{TL}$	100mV
T_F	Periodo del oscilador rápido	100ns

Tabla 3.1: Parámetros elegidos para el generador de números aleatorios.

Con estos valores elegidos se obtienen los parámetros estadísticos para el oscilador lento indicados en la tabla 3.2, los cuales cumplen todos los requisitos presentados anteriormente. También, se usará una tensión de referencia de 1.5V($V_{Ref} = 1.5V$), y una tensión de alimentación de 3.3V ($V_{DD} = 3.3V$), como se indicó en el capítulo anterior.

A continuación, se procede con el diseño de los distintos bloques. Los esquemáticos y *layouts* se encuentran en los anexos II y III.

Parámetro	Descripción	Valor
$E\{T_S\}$	Valor esperado del periodo del oscilador lento	3.18μs
$\sigma\{T_S\}$	Varianza del periodo del oscilador lento	1.13μs
$\sigma\{v_N\}$	Ruido referido a la salida	50.1mV rms

Tabla 3.2: Parámetros elegidos para el generador de números aleatorios.

3.3 Diseño del oscilador de baja frecuencia

3.3.1 Diseño del OPAMP

La topología elegida para el amplificador operacional (OPAMP) consta de dos etapas para alcanzar valores de ganancia altos que no se pueden obtener solo con una etapa, pues la ganancia se limitaría a la del par diferencial de entrada. Como se necesita un alto valor de ancho de banda, esto implica utilizar una corriente de polarización de entrada elevada para poder tener polos a frecuencias altas, este hecho se puede verificar del modelo de carga laminar simplificado, conforme la corriente de drenador sea mayor se tendrá una resistencia asociada al transistor mayor debido al efecto de modulación de canal del transistor MOS. No se está usando un condensador de compensación entre los terminales de drenador y compuerta del surtidor común para alcanzar valores altos de ancho de banda a costa de reducir el margen de fase.

Transistor	W(μm)	L(μm)
MN0	0.7	0.7
MN1	0.7	0.7
MN2	0.7	0.7
MP0	49	0.7
MP1	49	0.7
MP2	2.8	0.7
MP3	1.4	0.7
MP4	0.7	0.7

Tabla 3.3: Dimensiones de los transistores del OPAMP.

3.3.2 Diseño del OTA

Se eligió un amplificador operacional de transconductancia (OTA) simétrico, el cual presenta un voltaje de offset bajo [18]. La transconductancia del OTA depende, en esta topología, de la transconductancia del par diferencial de entrada y las ganancias de los espejos de corriente formados por MN0-MN3 y MN1-MN2, las cuales deben ser iguales.

En este tipo de topología, para obtener una alta linealidad, se debe mantener una transcontuctancia del par diferencial de entrada baja, para un mismo nivel de corriente, ya

que, como está demostrado en [19], la linealidad de este OTA depende de manera inversamente proporcional al voltaje de overdrive $V_{GS} - V_{Th}$ del par diferencial. Notar que este requisito implica que los transistores MP0 Y MP1 cuenten con un valor relativamente alto de longitud de canal, con lo cual fue necesario utilizar la técnica de *snake*, mencionada en [9], en el *layout* de estos.

Transistor	W(μm)	L(μm)
MN0	1	1.7
MN1	1	1.7
MN2	10	0.7
MN3	10	0.7
MP0	1	35
MP1	1	35
MP2	3.5	1
MP3	3.5	1
MP4	16	0.7
MP5	16	0.7

Tabla 3.4: Dimensiones de los transistores del OTA.

3.3.3 Diseño del comparador con histéresis

Este circuito consiste en un amplificador diferencial al cual se le ha aplicado realimentación positiva mediante los espejos de corriente formados por los transistores MP0, MP2 y MP1, MP3. El parámetro importante de este circuito es el voltaje de histéresis, que se define como la diferencia entre los voltajes umbrales de conmutación del comparador, es decir: $V_{Histéresis} = V_{Thp} - V_{Thn}$. Este voltaje depende de la ganancia de corriente de los espejos de corriente formados por los transistores MP0, MP2 y MP1, MP3 con la condición de que estas ganancias sean iguales [14].

Transistor	W(μm)	L(μm)
MN0	1	1
MN1	0.7	0.7
MN2	0.7	0.7
MN3	1	1
MN4	1	1
MN5	1	1
MP0	0.95	0.7
MP1	0.95	0.7
MP2	0.7	0.7
MP3	0.7	0.7
MP4	1	1
MP5	1	1

Tabla 3.5: Dimensiones de los transistores del comparador con histéresis.

3.3.4 Diseño del *charge pump*

La topología elegida para el *charge pump* consiste en un simple inversor CMOS conectado a dos fuentes de corriente como se muestra en la figura 4.10. Cuando el voltaje de entrada es elevado se obtiene la descarga del capacitor, disminuyendo el voltaje en sus terminales. Cuando el voltaje de entrada es bajo se le entrega corriente al capacitor, aumentando el voltaje en sus terminales. Las fuentes de corriente consisten en espejos de corriente simples de ganancia igual a 1, que se los ha dimensionado para ocupar la mínima área posible. El inversor CMOS fue dimensionado tal que el factor de forma del PMOS sea mayor al del NMOS, de tal manera que se compense la baja capacidad de corriente del primero, la cual es menor por ser la mobilidad de huecos menor a la de electrones $\mu_N > \mu_P$, en este caso se eligió una razón de 2.5 con longitud de canal mínima.

Transistor	W(μm)	L(μm)
MN0	0.4	0.35
MN1	0.4	0.35
MN2	0.4	0.35
MP0	0.4	0.35
MP1	0.4	0.35
MP2	1	0.35

Tabla 3.6: Dimensiones de los transistores del *charge pump*.

3.4 Diseño del oscilador de alta frecuencia

Se ha elegido diseñar un oscilador en anillo, ya que permite obtener alcanzar las especificaciones planteadas sin sacrificar área de circuito. Este consiste en un número impar de inversores CMOS en serie, donde la salida de uno se conecta a la entrada del siguiente. El dimensionamiento de los transistores se realiza de manera similar al caso anterior. Se eligió una razón de dimensiones entre PMOS y NMOS igual a 2.

Transistor	W(μm)	L(μm)
PMOS	9.6	4.8
NMOS	4.8	4.8

Tabla 3.7: Dimensiones de los transistores del oscilador rápido.

3.5 Diseño del *flip flop* T

El FF-T consiste en un FF-D realimentado mediante una compuerta XOR como se puede ver en la figura 4.13. Tanto para el FF-D y la compuerta XOR se utilizaron las llamadas compuertas de

transmisión las cuales permiten obtener un buen desempeño en velocidad al mismo tiempo que reducen considerablemente el número de transistores a utilizar, facilitando la elaboración del *layout*, siendo su única desventaja el hecho que los niveles lógicos alcanzados no son los más altos o bajos posibles en el circuito; sin embargo, para el generador aleatorio no es de mayor importancia llegar a alcanzar niveles lógicos fuertes. El dimensionamiento de los transistores se hace igual que en el oscilador en anillo. En este caso se eligió una razón de dimensiones entre PMOS y NMOS igual a 2.5 con longitud de canal mínima. El FF-T se activa con flanco de bajada.

Transistor	W(μm)	L(μm)
PMOS	1	0.35
NMOS	0.4	0.35

Tabla 3.8: Dimensiones de los transistores del FF-T.

Capítulo 4

Simulación del circuito propuesto

4.1 Verificación de los bloques funcionales diseñados

Para verificar el funcionamiento del circuito se realizó un análisis transitorio con el circuito mostrado en la figura 4.15, cuyo *layout* se muestra en la figura 4.6. Para esta simulación, no se tomó en cuenta el ruido transitorio generado por las resistencias, este fue deshabilitado en el simulador para poder medir los parámetros de interés del circuito. Los resultados de esta simulación se muestran en la figura 4.1. De esta figura se pueden calcular todos los parámetros indicados en la tabla 3.1 a excepción de los parámetros del OPAMP, para obtener estos se hizo un análisis *ac* y *stb* con el circuito de la figura 4.16. En la figura 4.3 se puede medir la ganancia y el ancho de banda en lazo cerrado del OPAMP, como resultados de la simulación *ac*, y en la figura 4.4 se puede medir el margen de fase, como resultado del análisis *stb*. Los resultados de todas estas simulaciones están disponibles en la tabla 4.1, mientras que en la tabla 4.2 se detallan el consumo y área del circuito obtenidos. El resultado de esta primera simulación muestra que el periodo del oscilador lento es aparentemente igual entre periodo y periodo, indicando un nivel de *jitter* inicial muy bajo.

Finalmente, se realizó el análisis transitorio con el ruido transitorio del simulador, los resultados se muestran en la figura 4.5. Como parámetros del ruido se fijó una frecuencia máxima de ruido igual a 2 GHz, notar que para esta frecuencia el OPAMP atenúa las señales de entrada más de mil veces su magnitud. Es importante recordar que el ruido a introducir por el simulador es pseudoaleatorio. Como es de esperar, en este caso, el *jitter* presente en el oscilador lento es mucho mayor que en el caso sin ruido, logrando tener una variación de bits a la salida mucho mayor.

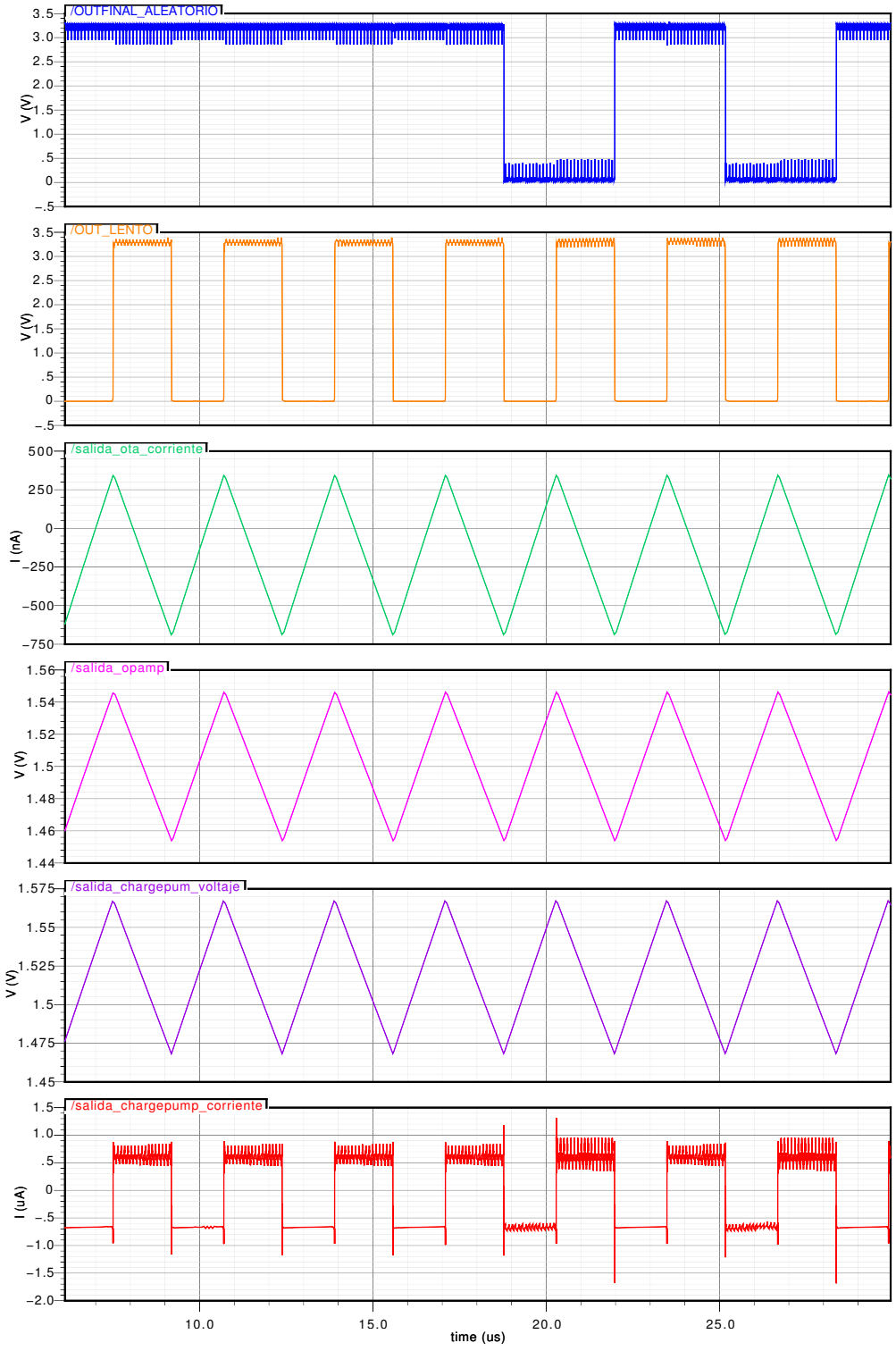

Figura 4.1: Formas de onda de interés presentes en el circuito en ausencia de ruido.

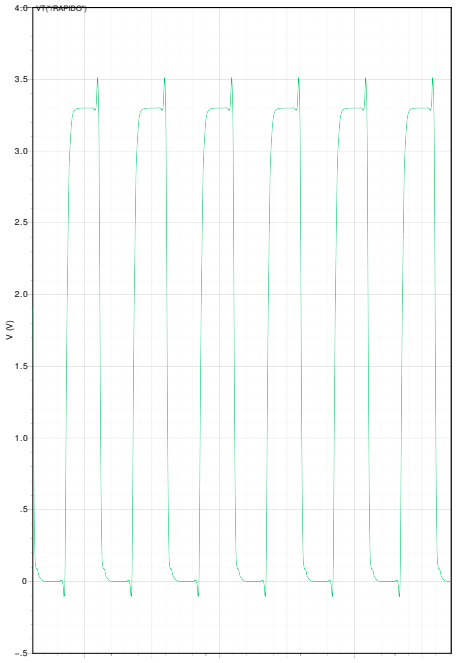

Figura 4.2: Forma de onda de salida del oscilador rápido.

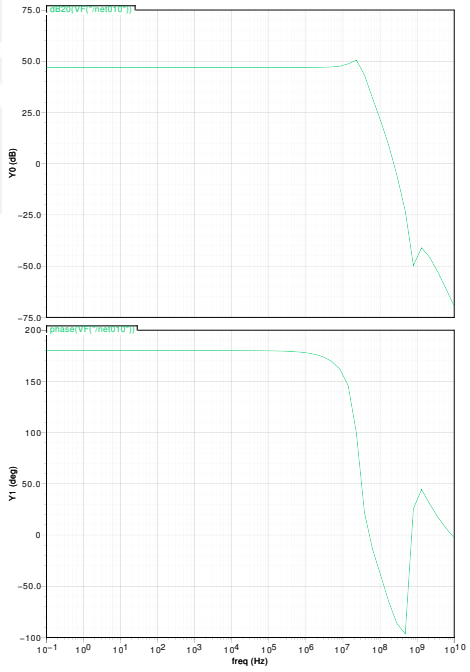

Figura 4.3: Respuesta en frecuencia del OPAMP: Lazo cerrado.

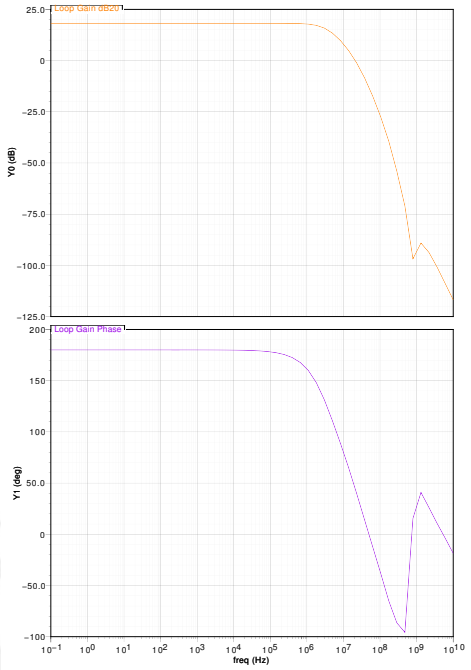

Figura 4.4: Respuesta en frecuencia del OPAMP: Ganancia de lazo.

Parámetro	Descripción	Valor
G	Ganancia del OPAMP en lazo cerrado	≈47dB
Bw	Ancho de banda del OPAMP en lazo cerrado	≈36MHz
MF	Margen de fase del OPAMP	$\approx 44°$
G_1	Transconductancia del OTA	$\approx 10\frac{\mu A}{V}$
I_{SAT}	Corriente de salida del *charge pump*	≈650nA
$V_{Histéresis}$	$V_{TH} - V_{TL}$	≈94mV
T_F	Periodo del oscilador rápido	≈99ns

Tabla 4.1: Parámetros obtenidos de la simulación.

Parámetro	Valor
Área	330μmx120μm
Consumo de potencia	3.6mW

Tabla 4.2: Área y consumo de potencia del circuito.

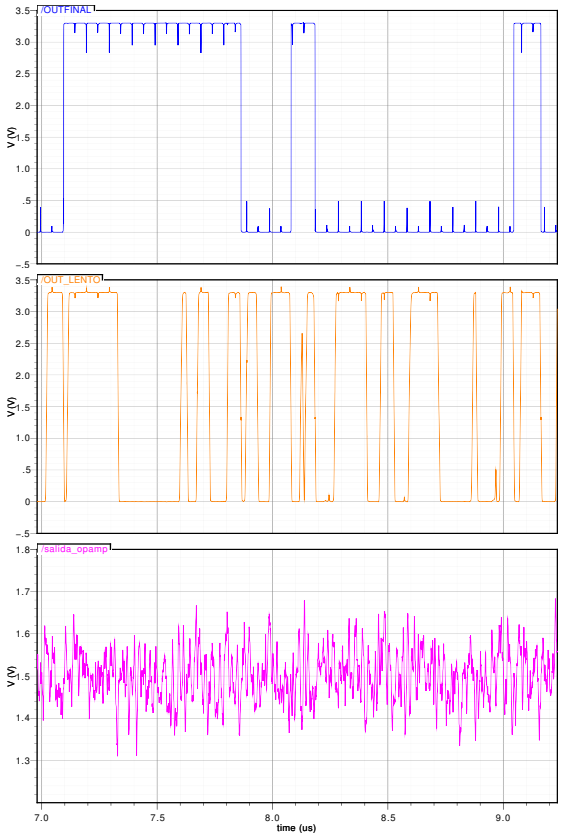

Figura 4.5: Formas de onda de salida con ruido.

4.2 Evaluación de la calidad aleatoria del generador

4.2.1 Probabilidad de transición

La probabilidad de transición, calculada en MATLAB para una secuencia generada de 2200 bits, da el siguiente resultado:

$$P_t \approx 0.4993$$

Esto verifica que el circuito se comporta como se predijo en el capítulo 3, pues la probabilidad de transición se acerca a su valor ideal $P_{t(Ideal)} = 0.5$ e incluso mejor que en [4]. Se hace incapié en que, este hecho no es un indicativo certero de la calidad aleatoria del generador, este parámetro solo indica que el circuito se está comportando, estadísticamente, como se predijo en el capítulo 3.

4.2.2 Algoritmos del NIST

Los tests propuestos por el NIST consisten en un conjunto de 15 algoritmos, los cuales tienen un parámetro de medida conocido como *P-Value*, proveniente de la estadística, que representa la probabilidad de aceptar cierta hipótesis, aleatoriedad y uniformidad en el presente caso. Los fundamentos matemáticos sobre el *P-Value* no son tema de la presente tesis, estos se pueden encontrar en [20] o de manera resumida en [21]. Asimismo, los detalles sobre cada algoritmo se pueden encontrar en la misma documentación proporcionada por el NIST [22] o en [21] y [23]. En este trabajo, los tests son aplicados sobre varias secuencias de entrada, cada uno de estos evalúa, según el algoritmo correspondiente, cuántas de las secuencias pueden ser consideradas aleatorias, por medio de este, mostrando el porcentaje de secuencias que obtienen un resultado favorable del algoritmo y el porcentaje mínimo necesario que deben pasar. Adicionalmente, se indica el *P-value* bajo la hipótesis de que las secuencias son uniformemente aleatorias [22]. Para afirmar que el generador aleatorio pasa completamente el test se deberá cumplir que el porcentaje de secuencias, que obtienen un resultado favorable por parte del algoritmo, sea mayor al mínimo indicado y además estas deben ser uniformemente aleatorias. Esto quiere decir que se puede dar el caso en que el algoritmo usado en cierto test acepte como aleatorias todas las secuencias entregadas; sin embargo, estas pueden no ser uniformemente aleatorias, con lo cual no se podría concluir que el generador aleatorio pasó por completo el test.

Para obtener los bits aleatorios se elaboró un *script* en OCEAN, disponible en el apéndice, que permite simular periodos de corta duración, donde las condiciones finales de una simulación se usan como condiciones iniciales de la siguiente. Esto, sumado con que solo

se guarda la señal de salida del FF-T, permite un gran ahorro de espacio en el disco duro. Asimismo, el *script* permite evaluar si la onda presenta, en ciertos instantes de tiempo, un valor mayor o menor a un voltaje umbral, que se eligió 1.5V, para discernir si se tiene un 1 o 0 lógico. Cada intervalo de tiempo viene dado por la ecuación 3.1. En la tabla 4.3 se muestran los resultados finales que se obtienen, solo se realizaron los tests en los cuales se tenían la cantidad de bits recomendadas en [22]. Se utilizaron 20 secuencias de 110 bits cada una.

Test	Nombre	Porcentaje	*P-Value*	Resultado
1	Frequency(Monobit)	100	0.834308	✓
2	Frequency within a Block	100	0.534146	✓
3	Runs	100	-	
4	Longest Run of Ones in a Block	95	0.834308	✓
5	Binary Matrix Rank	-	-	
6	Discrete Fourier Transform	100	-	
7	Non - Overlapping Template Matching	-	-	
8	Overlapping Template Matching	-	-	
9	Maurer's "Universal Statistical"	-	-	
10	Linear Complexity	-	-	
11	Serial	-	-	
12	Approximate Entropy	-	-	
13	Cumulative Sums(Cusum)	100	0.035174	✓
14	Random Excursions	-	-	
15	Random Excursions Variant	-	-	

Tabla 4.3: Resultados obtenidos al aplicar los algoritmos del NIST.

Conclusiones

- Se diseñó un generador de números aleatorios y se ha verificado que los parámetros de los distintos bloques que componen el circuito cumplen con las especificaciones iniciales de las que se partió.
- Se obtiene un error relativo de 0.14% para el valor de probabilidad de transición, introducido en [4], respecto a su valor ideal.
- El generador aleatorio diseñado no logra pasar todos los tests del NIST, debido a que se está utilizando el ruido pseudoaleatorio del simulador en lugar del ruido real de las resistencias. Por esta razón, no se puede descartar su uso en una aplicación criptográfica.
- Los resultados obtenidos no son concluyentes respecto a la calidad del generador aleatorio diseñado, debido a que se está simulando el ruido de las resistencias y no se están usando suficientes secuencias de bits como para que los resultados tengan suficiente significado estadístico. La principal razón de este inconveniente es que las simulaciones tardan demasiado. Como referencia, para obtener una secuencia de 1000 bits, la simulación tardó aproximadamente 26 horas en un procesador Intel Core i7 3610QM con 16GB DDR3 RAM.
- El hecho de que el generador logre pasar algunos tests del NIST indica que el circuito diseñado ha logrado introducir ruido de manera significativa.

Recomendaciones y observaciones

- Se debe realizar el análisis estadístico de las secuencias generadas por el circuito integrado fabricado, solo de esta manera se puede aceptar o rechazar su uso en aplicaciones criptográficas.
- Los valores escogidos de ancho de banda y ganancia para el OPAMP podrían ser reducidos con el fin de disminuir el consumo de potencia.
- Se puede reducir el periodo del oscilador rápido; tal que, la relación de este respecto al valor esperado del oscilador lento sea de 1 a 100, como usualmente se suele hacer en la práctica, esto aumentaría la aleatoriedad y disminuiría el área del circuito. Se usó un valor alto para que las simulaciones hechas tardaran menos.
- Implementar el presente circuito en una tecnología más reciente permitiría obtener mayores velocidades de operación a costa de trabajar con los efectos de los transistores de canal corto [11].

Bibliografía

[1] R. J. Baker, *CMOS Circuit Design, Layout and Simulation*, 3era ed. Wiley-IEEE Press, 2010.

[2] C. S. Petrie, "A noise-based ic random number generator for applications in cryptography," *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, vol. 47, pp. 615–621, Mayo 2000.

[3] C. S. Petrie y J. A. Conelly, "The sampling of noise for random number generation," *Proceedings of the 1999 IEEE Internationational Symposium on Circuits and Systems*, vol. 6, pp. 26–29, Julio 1999.

[4] M. Bucci *et al.*, "A high-speed oscillator-based truly random number source for cryptographic applications on a smart card ic," *IEEE Transactions on computers*, vol. 52, pp. 403–409, Abril 2003.

[5] S. Ergün, "A truly random number generator based on a pulse-excited cross-coupled chaotic oscillator," *Proceedings of the 25th Internationational Symposium on Circuits and Systems*, pp. 415–420, Septiembre 2010.

[6] P. K. B. Jun, "The Intel random number generator," *Cryptography Research, INC. White Paper Prepared for Intel Corporation*, Abril 1999.

[7] B. Sunar. State of the art in true random number generation. [Documento en línea]. Disponible en: helper.ipam.ucla.edu/publications/scws4/scws4_6650.pdf [Consulta: Noviembre 2013].

[8] B. Razavi, *Design of Analog CMOS Integrated Circuits*. McGraw-Hill Companies, 2003.

[9] A. Hastings, *The Art of Analog Layout*, 2da ed. Prentice Hall, 2005.

[10] C. Saint y J. Saint, *IC Mask Design*. McGraw Hill, 2002.

[11] Y. Tsividis y C. McAndrew, *Operation and Modeling of the MOS Transistor*, 3era ed. Oxford University Press, Inc., 2011.

[12] C. C. Enz y E. A. Vitoz, *Charge Based MOS Transistor Modeling*. Wiley, 2006.

[13] C. G. Montoro y M. C. Schneider, *MOSFET Modeling for Circuit Analysis and Design*. World Scientific Publishing Company, 2007.

[14] P. E. Allen y D. R. Holberg, *CMOS Analog Circuit Design*, 2da ed. Oxford University Press, 2002.

[15] H. Camenzind, *Designing Analog Chips*. Virtualbookworm.com Publishing, 2005.

[16] W. Che *et al.*, "Scheme of truly random number generator application in RFID tag," *Auto-ID Labs White Paper WP-HARDWARE-023*, 2006.

[17] M. H. Hayes, *Statistical Digital Signal Processing and Modeling*. Wiley, 1996.

[18] W. Sansen, *Analog Design Essentials*. Springer, 2006.

[19] F. Varela, "Diseño en CMOS de un filto pasa-bajo con frecuencia de corte 150Hz para la adquisición de señales del electrocardiograma," *Tesis de pregrado, Pontificia Universidad Católica del Perú*, Agosto 2011.

[20] A. Papoulis, *Probability, Random Variables and Stochastic Processes*, 4ta ed. McGraw-Hill Europe, 2002.

[21] R. G. Sadique Zaman, "Review on fifteen statistical tests proposed by NIST," *Journal of theoretical physics and cryptography*, vol. 1, Noviembre 2012.

[22] Rukhin *et al.*, "A statistical test suite for random and pseudorandom number generators for cryptographic applications," *National Institute of Standards and Technology Special Publication 800-22*, Abril 2010.

[23] F. Corporation y R. B. P. Dept, "The evaluation of randomness of RPG100 by using NIST and Diehard tests," Diciembre 2003.