Instituto Nacional de Astrofísica, Óptica y Electrónica (INAOE)

Maestría en Ciencias en Electrónica

TRNGs para generación de secuencias muy largas

Autor: Ciro Fabián Bermúdez Márquez

Asesor: Dr. Esteban Tlelo Cuautle (INAOE)
Coasesor: Dr. Cuauhtemoc Mancillas López (CINVESTAV IPN)

23 de mayo de 2023

Introducción

Números aleatorios

Los números aleatorios son valores numéricos que se generan de manera impredecible o aparentemente sin ningún patrón discernible. Tienen dos requisitos básicos.

- Buenas propiedades estadísticas.
  - Distribución de probabilidad uniforme.
  - Pasar pruebas estadísticas.
- Imprevisibilidad del número aleatorio.
  - No debe haber correlaciones o dependencias entre ellos.

Introducción

Aplicaciones

- Simuladores.
- Muestreo estadístico.
- Prueba de algoritmos.
- Análisis numérico.
- Estética en imágenes.
- Teoría de juegos.
- Criptografía.

Aplicaciones criptográficas

- Claves criptográficas.
- Números de un solo uso.
- Valores de relleno.
- Vectores de inicialización.
- Desafíos.
- Máscaras aleatorias.

Introducción

Caos

Es un comportamiento aperiódico, aparentemente impredecible, en sistemas deterministas los cuales presentan extrema sensibilidad a las condiciones iniciales, el más mínimo cambio en la condición inicial produce un resultado muy diferente [1].

Según [2] los sistemas caóticos tienen las siguientes características:

- Son aperiódicos.
- Presentan una dependencia sensible de las condiciones iniciales.
- Se rigen por uno o varios parámetros de control.
- Sus ecuaciones son no lineales.

Objetivos

Objetivo general

Diseñar e implementar en FPGA un TRNG híbrido para la generación de secuencias muy largas.

Objetivo específicos

- Investigar el estado del arte de diferentes generadores de números aleatorios.
- Estudiar los diferentes tipos de generadores de números aleatorios y analizar sus características principales.
- Estudiar la teoría de los mapas caóticos y su utilidad en generadores de números aleatorios.
- Diseñar un generador de números aleatorios híbrido utilizando un TRNG como generador de semillas y un mapa caótico para realizar un postprocesamiento que mejore sus características estadísticas y comprobar estas utilizando las pruebas NIST.
- Implementar el TRNG híbrido en una FPGA.

Generadores de números aleatorios (RNGs)

Agencias de estandarización

Agencias de estandarización como:

- NIST (National Institute of Standards and Technology) [3] en Estados Unidos.
- BSI (Federal Office for Information Security) [4] en Alemania.

han creado una serie documentaciones, recomendaciones y pruebas estadísticas para evaluar los generadores de números aleatorios.

Generadores de números aleatorios (RNGs)

Estructura de los TRNG

Diagrama de bloques de un TRNG según las especificaciones de AIS-20/31 y NIST 800-90B.

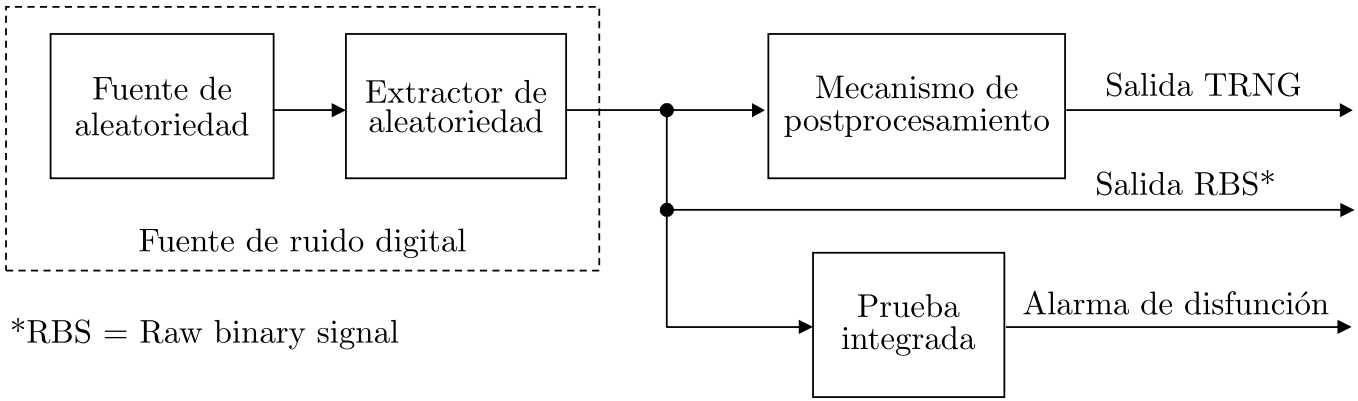


Figura 1: Estructura general de un TRNG [5].

Generadores de números aleatorios (RNGs)

Clasificación de los TRNG

Random Number Generators (RNG)

- True Random Number Generators (TRNG).
  - Físicos (PTRNG).
  - No físicos (NPTRNG).
- Deterministic Random Number Generators (DRNG).
- Hibrid Random Number Generators (HRNG).
  - HTRNG.
  - HDRNG.

Generadores de números aleatorios (RNGs)

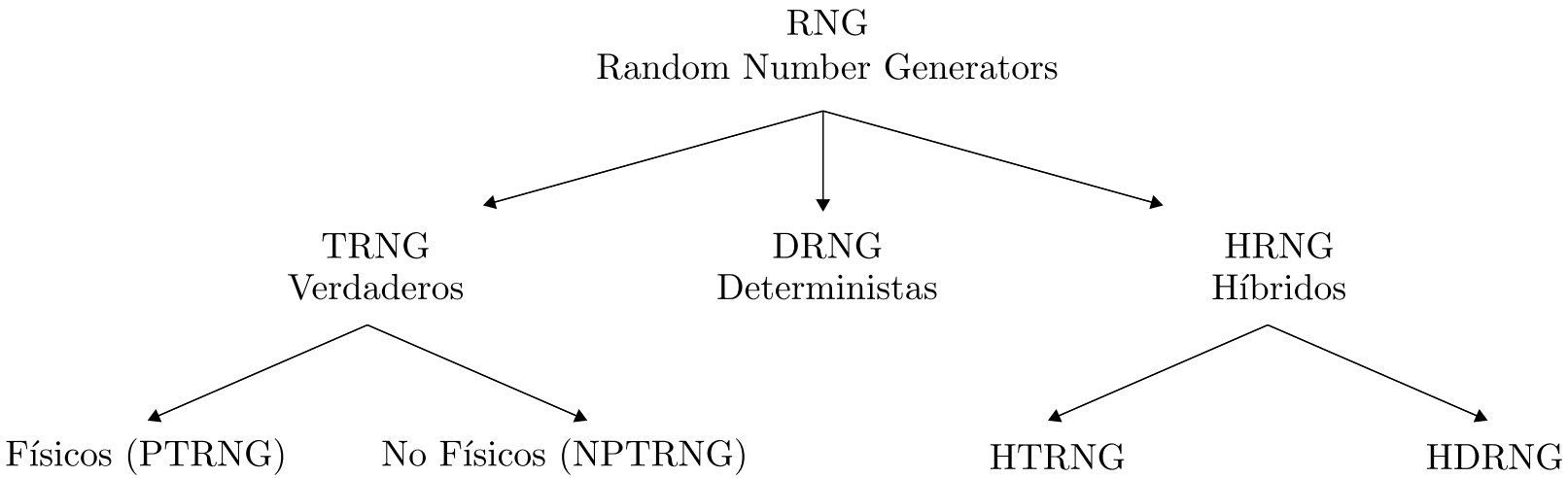


Figura 2: Clasificación de los RNG.

Generadores de números aleatorios (RNGs)

Parámetros de evaluación

- Parámetros relacionados con la calidad
  - Fuente de aleatoriedad.
  - Método de extracción de aleatoriedad y entropía del ruido digital.
  - Método de postprocesamiento (opcional).
  - Tasa de bits de salida y su estabilidad.
- Parámetros relacionados con la seguridad
  - Existencia de un modelo matemático.
  - Comprobabilidad interna.
  - Seguridad (robustez, resistencia contra ataques).
- Parámetros relacionados con el diseño
  - El uso de recursos.
  - El consumo de energía.
  - Viabilidad en dispositivos lógicos y FPGAs.
  - Automatización del diseño.

Generadores de números aleatorios (RNGs)

Fuentes de aleatoriedad

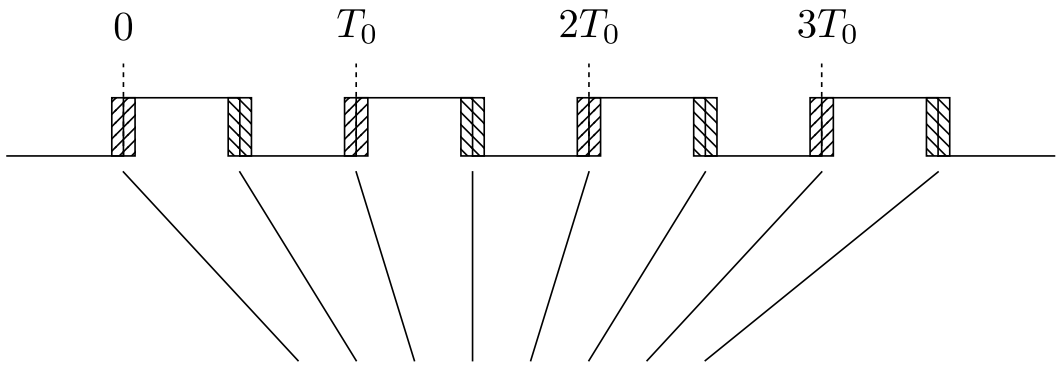
Los fenómenos físicos más utilizados para generar números aleatorios en los dispositivos lógicos son:

- Señales analógicas (ruido térmico).
- Metaestabilidad.
- Jitter del reloj.
- Caos.

Generadores de números aleatorios (RNGs)

Jitter

Es una variación del flanco del reloj desde su posición ideal o la incertidumbre en la oscilación del reloj en el dominio del tiempo.



Dependiendo del tamaño del jitter, el flanco de reloj puede llegar en cualquier punto de estas regiones

Figura 3: Jitter del reloj.

Generadores de números aleatorios (RNGs)

Jitter

Los circuitos digitales utilizan un nivel de referencia para detectar los flancos de reloj. Este nivel de referencia debe ser más estable posible. En la práctica sufre fluctuaciones debido a diversos tipos de ruido. Cuando el nivel de referencia se desplaza, los flancos del reloj se detectan antes o después de lo previsto, lo que resulta en una variación temporal conocida como jitter de reloj.

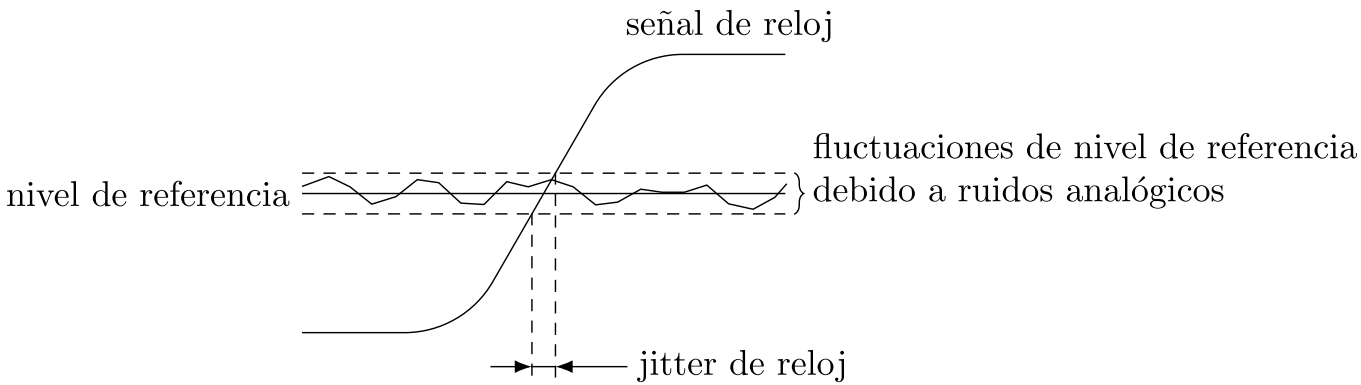


Figura 4: Fluctuaciones del nivel de referencia originadas por ruidos analógicos que provocan jitter del reloj en los circuitos digitales. [6]

Generadores de números aleatorios (RNGs)

Núcleos TRNG en FPGA

Con base en los criterios del AIS-20/31 [7], los núcleos TRNG adecuados para utilizarse en dispositivos lógicos programables (FPGA) que usan estructuras oscilantes son:

- Elementary ring oscillator based TRNG (ERO-TRNG).
- Coherent sampling ring oscillator based TRNG (COSO-TRNG).
- Multi-ring oscillator based TRNG (MURO-TRNG).
- Transient effect ring oscillator based TRNG (TERO-TRNG).
- Self-timed ring based TRNG (STR-TRNG).
- Phase-locked loop based TRNG (PLL-TRNG).

Generadores de números aleatorios (RNGs)

Núcleo ERO-TRNG, Elementary ring oscillator based TRNG

Se propuso y modeló en [8]. Dos osciladores de anillo idénticos forman la base del generador. Uno de ellos se utiliza para muestrear la salida del otro oscilador en anillo utilizando un flip-flop D. La frecuencia del oscilador de anillo de muestreo se divide por K para obtener una frecuencia más baja de la señal de muestreo, lo que permitiría acumular el jitter del oscilador de anillo muestreado.

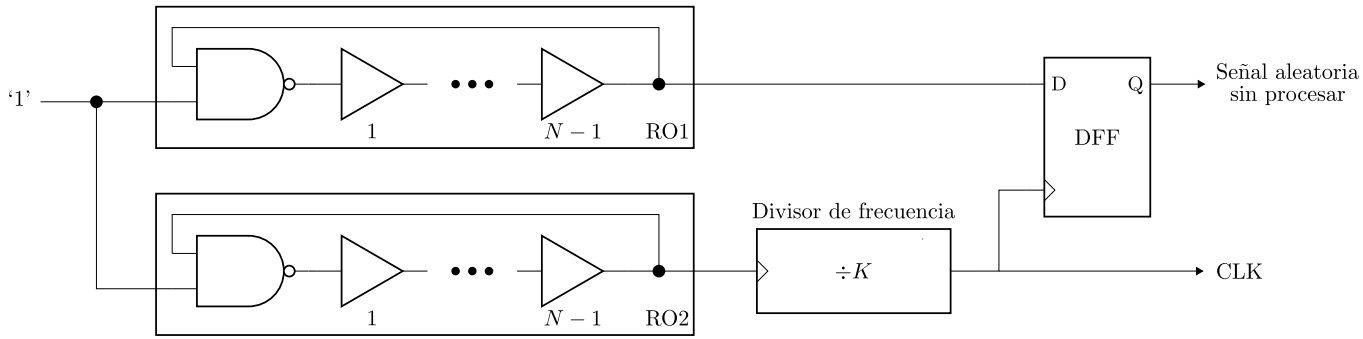


Figura 5: Diagrama de ERO-TRNG.

Generadores de números aleatorios (RNGs)

Ventajas y desventajas del núcleo ERO-TRNG

Ventajas

- Implementación sencilla.
- Repetible sin intervención manual.
- Sólido modelo estocástico.
- Bajo consumo de potencia.

Desventajas

- Tasa de bits de salida relativamente baja.

Generadores de números aleatorios (RNGs)

Núcleo COSO-TRNG, Coherent sampling ring oscillator based TRNG

Se propuso por primera vez en [9]. Utiliza dos osciladores de anillo idénticos como fuente de aleatoriedad. Las frecuencias de los osciladores varían un poco. Al muestrear la salida de uno de los osciladores mediante un flip-flop D sincronizado con la salida del otro oscilador, se obtiene una señal con un periodo variable, que corresponde al desfase relativo de los dos osciladores.

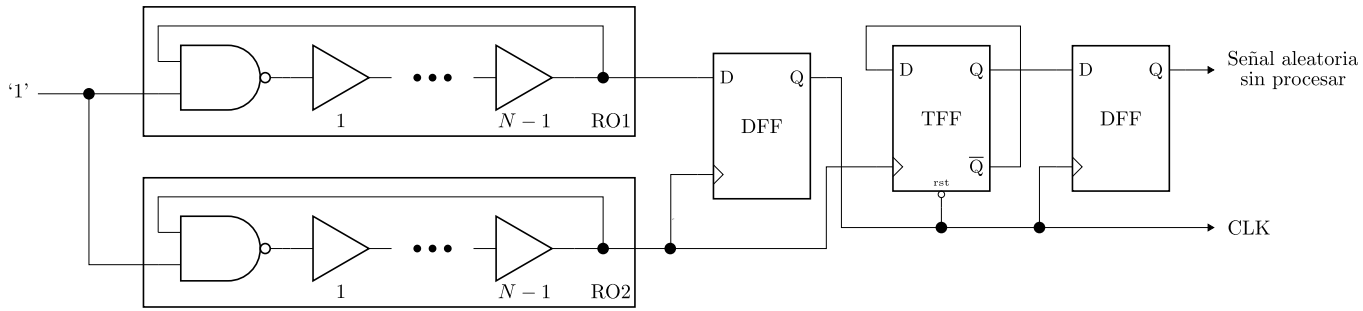


Figura 6: Diagrama de COSO-TRNG.

Generadores de números aleatorios (RNGs)

Ventajas y desventajas del núcleo COSO-TRNG

Ventajas

- Ocupa un área muy pequeña.
- Tasa de bits de salida relativamente alta.
- Bajo consumo de potencia.

Desventajas

- Implementación difícil.
- Requiere intervención manual.

Generadores de números aleatorios (RNGs)

Núcleo MURO-TRNG, Multi-ring oscillator based TRNG

Se propuso por primera vez en [10]. Utiliza m osciladores en anillo como fuente de aleatoriedad. Los osciladores deben ser independientes y su fase uniformemente distribuida. Para extraer la aleatoriedad el número de osciladores de anillo debe satisfacer $m > T/\sigma$ donde T es el valor medio del periodo de reloj y σ es la desviación típica del jitter acumulado durante el periodo de muestreo.

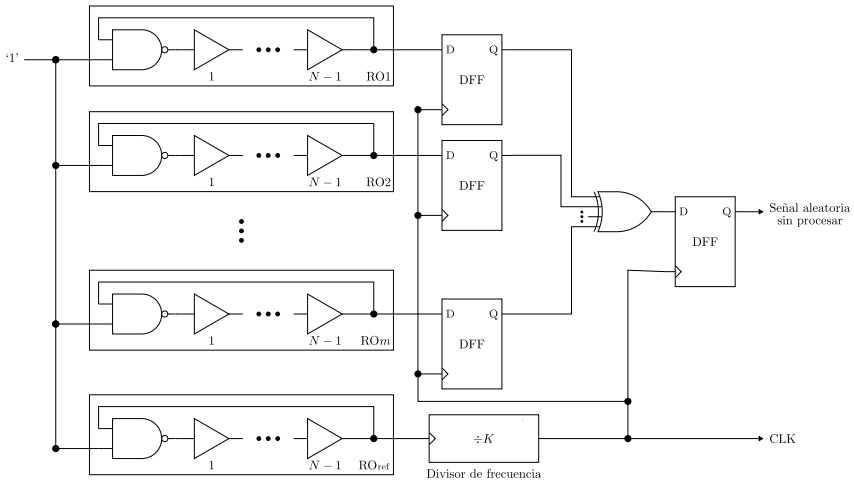


Figura 7: Diagrama de MURO-TRNG.

Generadores de números aleatorios (RNGs)

Ventajas y desventajas del núcleo MURO-TRNG

Ventajas

- No requiere intervención manual.
- Tasa de bits relativamente alta.

Desventajas

- Ocupa un área muy grande.
- Los osciladores de anillo pueden bloquearse entre sí.
- Alto consumo de potencia.

Generadores de números aleatorios (RNGs)

Núcleo TERO-TRNG, Transient effect ring oscillator based TRNG

Se propuso en [11] y el modelo estocástico en [12]. Genera bits aleatorios utilizando metaestabilidad oscilatoria. Debido a la metaestabilidad oscilatoria del TERO, el número de oscilaciones es aleatorio. Para producir un flujo de bits aleatorios, el TERO debe reiniciarse periódicamente

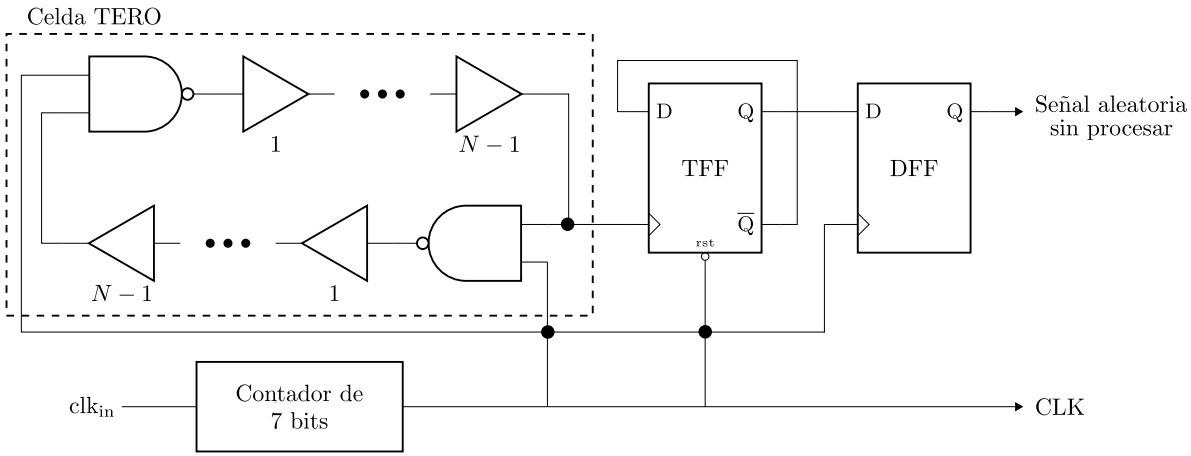


Figura 8: Diagrama de TERO-TRNG.

Generadores de números aleatorios (RNGs)

Ventajas y desventajas del núcleo TERO-TRNG

Ventajas

- Ocupa un área muy pequeña.
- Tasa de bits relativamente alta.
- Bajo consumo de potencia.

Desventajas

- Requiere intervención manual.
- No es repetible.

Generadores de números aleatorios (RNGs)

Núcleo STR-TRNG, Self-timed ring based TRNG

Se propuso en [13]. Se compone de L celdas Muller. El principio de extracción de aleatoriedad de un STR-TRNG es el mismo que el de MURO-TRNG. El reloj de muestreo se genera mediante un oscilador en anillo. La ventaja de un STR es que, si se configura correctamente, garantiza que las fases del reloj de salida estén igualmente espaciadas

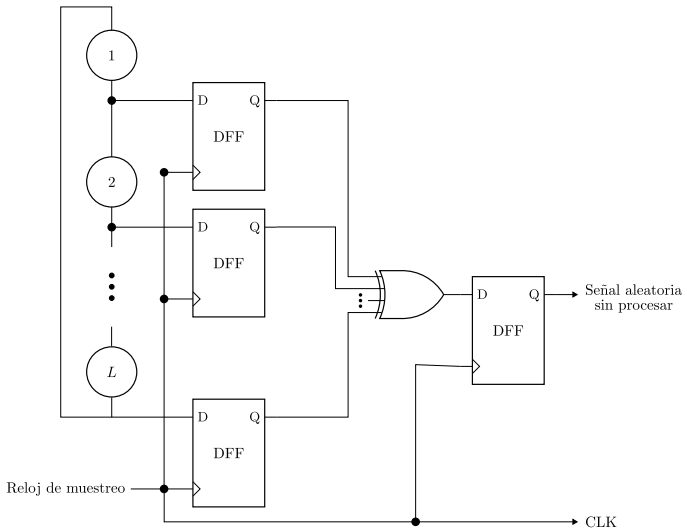


Figura 9: Diagrama de STR-TRNG.

Generadores de números aleatorios (RNGs)

Ventajas y desventajas del núcleo STR-TRNG

Ventajas

- La tasa de bits es muy alta.

Desventajas

- Ocupa un área muy grande.
- Requiere intervención manual.
- Alto consumo de potencia.

Generadores de números aleatorios (RNGs)

Núcleo PLL-TRNG, Phase-locked loop based TRNG

Se propuso en [14]. El generador se basa en el hecho de que utilizando PLLs, las frecuencias de dos relojes generados están mutuamente relacionadas. La elección de los parámetros del PLL no es trivial. Hay que respetar muchas restricciones, incluidas las físicas del fabricante del PLL y las de seguridad del TRNG.

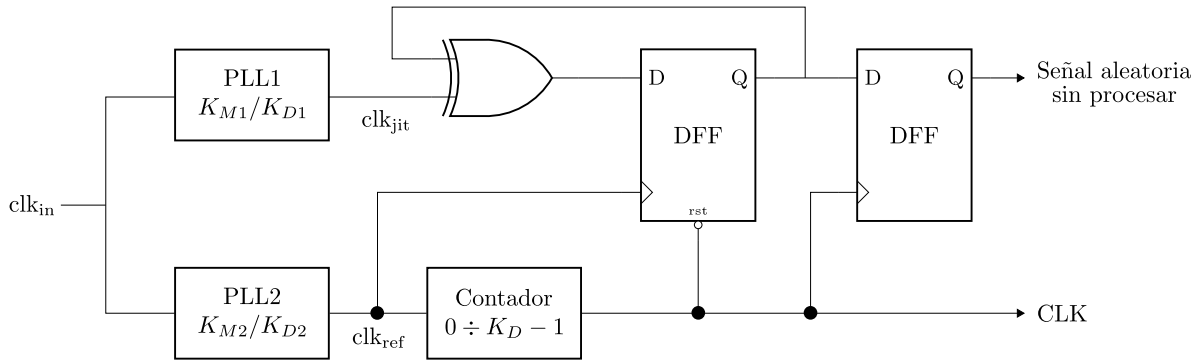


Figura 10: Diagrama de PLL-TRNG.

Generadores de números aleatorios (RNGs)

Ventajas y desventajas del núcleo PLL-TRNG

Ventajas

- No requiere intervención manual.
- Es repetible.
- Ocupa un área relativamente pequeña.
- La tasa de bits es relativamente alta.

Desventajas

- Consumo de potencia medio.
- Diferencias entre familias de FPGAs.

Generadores de números aleatorios (RNGs)

Núcleos TRNG en FPGA

Tabla 1: Resumen de los resultados núcleos TRNGs [7].

TRNG Type	FPGA device	Area (LUT/Reg)	Power cons. [mW]	Bit Rate [Mbits/s]	Efficiency [bits/μWs]	Entropy per bit	Entropy * Bit rate	Feasib. & Repeat.
ERO	Spartan 6	46/19	2.16	0.0042	1.94	0.999	0.004	5
	Cyclone V	34/20	3.24	0.0027	0.83	0.990	0.003	
	SmartFusion 2	45/19	4	0.014	3.5	0.980	0.013	
COSO	Spartan 6	18/3	1.22	0.54	442.6	0.999	0.539	1
	Cyclone V	13/3	0.9	1.44	1600	0.999	1.438	
	SmartFusion 2	23/3	1.94	0.328	169	0.999	0.327	
MURO	Spartan 6	521/131	54.72	2.57	46.9	0.999	2.567	4
	Cyclone V	525/130	34.93	2.2	62.9	0.999	2.197	
	SmartFusion 2	545/130	66.41	3.62	54.5	0.999	3.616	
PLL	Spartan 6	34/14	10.6	0.44	41.5	0.981	0.431	3
	Cyclone V	24/14	23	0.6	43.4	0.986	0.592	
	SmartFusion 2	30/15	19.7	0.37	18.7	0.921	0.340	
TERO	Spartan 6	39/12	3.312	0.625	188.7	0.999	0.624	1
	Cyclone V	46/12	9.36	1	106.8	0.987	0.985	
	SmartFusion 2	46/12	1.23	1	813	0.999	0.999	
STR	Spartan 6	346/256	65.9	154	2343.2	0.998	154.121	2
	Cyclone V	352/256	49.4	245	4959.1	0.999	244.755	
	SmartFusion 2	350/256	82.52	188	2286.7	0.999	188.522	

Mapa caótico

Definición de mapa caótico

Los mapas caóticos, mapas iterados, ecuaciones de diferencias o simplemente mapas, son sistemas dinámicos en tiempo discreto que tienen la forma general $x_{n+1} = f(x_n)$, los cuales requieren una condición inicial x_0 y se iteran continuamente para conocer su comportamiento. A la secuencia $x_0, x_1, x_2, \ldots$ se le conoce como la órbita del mapa comenzando desde x_0.

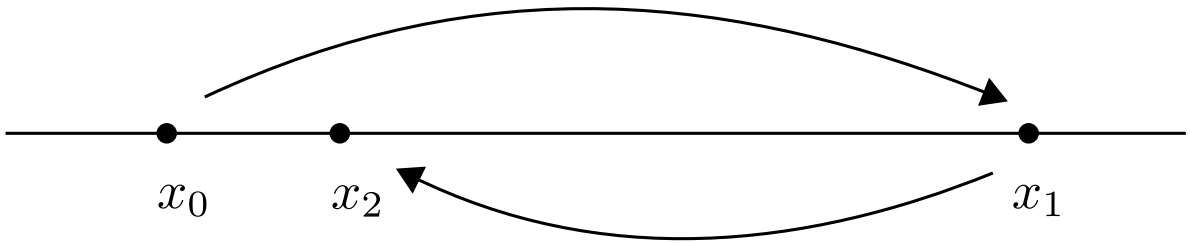


Figura 11: Ejemplo de mapeo en una dimensión.

Mapa caótico

Mapa caótico bidimensional

En el artículo [15] se analiza y estudia el mapa caótico cuadrático bidimensional más sencillo que esta dado por:

$$\begin{aligned} x_{n+1} &= a_1 + a_2 x_n + a_3 x_n^2 + a_4 x_n y_n + a_5 y_n + a_6 y_n^2 \\ y_{n+1} &= a_7 + a_8 x_n + a_9 x_n^2 + a_{10} x_n y_n + a_{11} y_n + a_{12} y_n^2 \end{aligned} \quad (1)$$

donde los parámetros $\{a_1, a_2, \ldots a_{12}\}$ y las condiciones iniciales x_0 y y_0 determinan las características de la solución. La iteraciones se representan como puntos en una superficie bidimensional. Requiere de 10 sumas y 13 multiplicaciones.

Mapa caótico

Comportamiento del mapa

La iteraciones se representan como puntos en una superficie bidimensional. Después de un cierto número de iteraciones, la solución hará una de estas cuatro cosas:

(a) convergerá a un único punto fijo.

(b) tomará una sucesión de valores que acabarán repitiéndose, produciendo un ciclo límite.

(c) será inestable y divergirá hasta el infinito.

(d) mostrará caos y rellenará gradualmente alguna región a menudo complicada pero acotada del plano $x - y$.

Mapa caótico

Pasos para encontrar atractores

Para saber qué valores de $\{a_1, a_2, \ldots a_{12}\}$ llevan al caos, [15] utilizó el siguiente procedimiento.

1. Elegir aleatoriamente los 12 coeficientes de a_1 hasta a_{12} aleatoriamente sobre algún intervalo.
2. Elegir las condiciones iniciales x_0 y y_0.
3. Iterar las ecuaciones del mapa mientras se calcula el exponente de Lyapunov y se comprueba si no hay divergencia.
4. Mantener las soluciones que están acotadas y tienen un exponente de Lyapunov positivo.

Mapa caótico

Codificación y rango de búsqueda

- Los coeficientes seleccionados en incrementos de 0.1 en el intervalo de -1.2 a 1.2, es decir, 25 valores posibles.
- Codificación con letras del alfabeto A hasta la Y.
- $A = -1.2, B = -1.1, C = -1.0, \ldots, Y = 1.2$.
- Cada atractor se identifica unívocamente con un nombre de 12 letras.
- El número de posibles casos es 25^{12} o aproximadamente 6×10^{12}.
- Aproximadamente 1.6 % son caóticos.

Mapa caótico

Codificación de parámetros a_n

Tabla 2: Conversiones para codificación de los atractores.

Letra	Codificación	Letra	Codificación
A	-1.2	N	0.1
B	-1.1	O	0.2
C	-1.0	P	0.3
D	-0.9	Q	0.4
E	-0.8	R	0.5
F	-0.7	S	0.6
G	-0.6	T	0.7
H	-0.5	U	0.8
I	-0.4	V	0.9
J	-0.3	W	1.0
K	-0.2	X	1.1
L	-0.1	Y	1.2
M	-0.0		

Mapa caótico

Ejemplos de atractores

Tabla 3: Diversos identificadores, dimensión fractal y exponente de Lyapunov para atractores del mapa caótico bidimensional.

Atractor	Identificador	Dimensión fractal	Exponente de Lyapunov
1	GLXOESFTTPSV	1.77	0.12
2	CVQKGHQTPHTE	1.79	0.14
3	UWACXDQIGKHF	1.42	0.10
4	GIIETPIQRRUL	1.50	0.13

Mapa caótico

Decodificación de parámetros de a_n para generar diferentes atractores

Después de decodificar los identificadores tenemos:

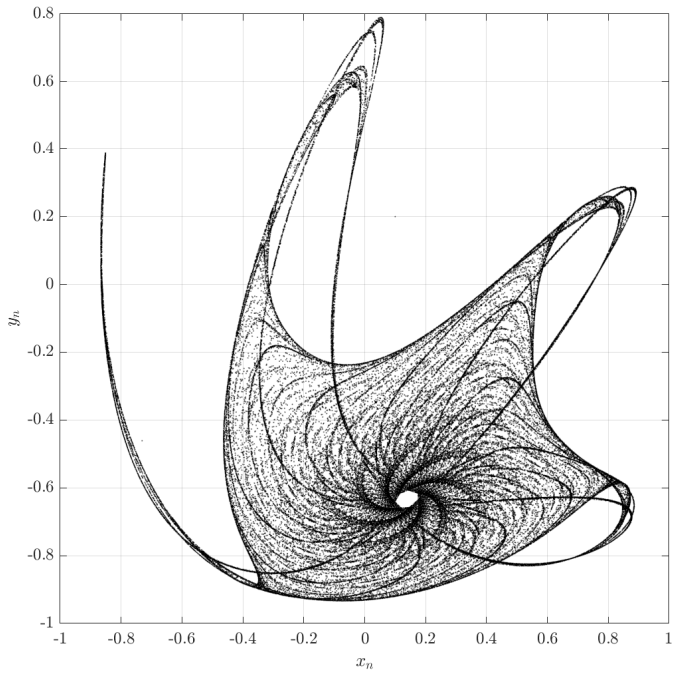
$$\begin{aligned}
A_1 &= \{-0.6, -0.1, \ 1.1, \ 0.2, -0.8, \ 0.6, -0.7, \ 0.7, \ 0.7, \ 0.3, \ 0.6, \ 0.9\} \\
A_2 &= \{-1.0, \ 0.9, \ 0.4, -0.2, -0.6, -0.5, \ 0.4, \ 0.7, \ 0.3, -0.5, \ 0.7, -0.8\} \\
A_3 &= \{\ 0.8, \ 1.0, -1.2, -1.0, \ 1.1, -0.9, \ 0.4, -0.4, -0.6, -0.2, -0.5, -0.7\} \\
A_4 &= \{-0.6, -0.4, -0.4, -0.8, \ 0.7, \ 0.3, -0.4, \ 0.4, \ 0.5, \ 0.5, \ 0.8, -0.1\}
\end{aligned}$$

utilizando las condiciones iniciales:

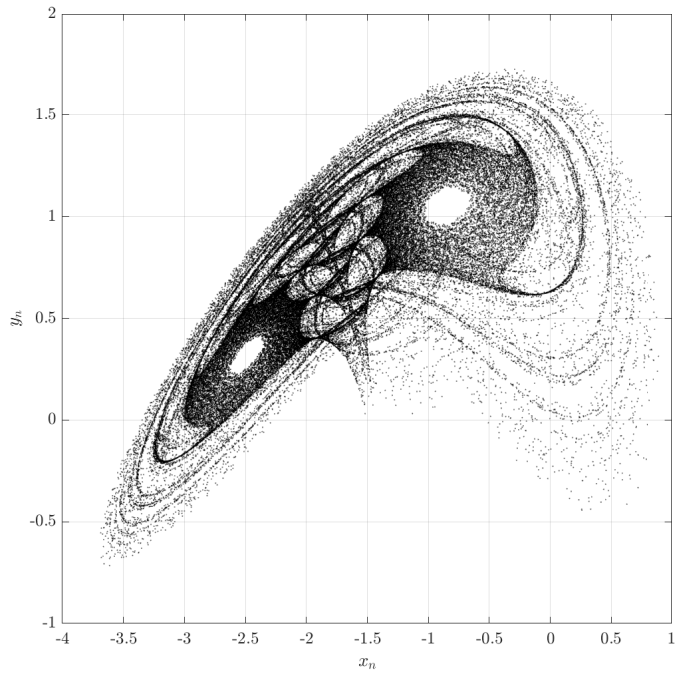
$$x_0 = 0.05, y_0 = 0.05 \tag{2}$$

Mapa caótico

Figura 12: Diferentes atractores caóticos del mapa bidimensional A_1 y A_2.



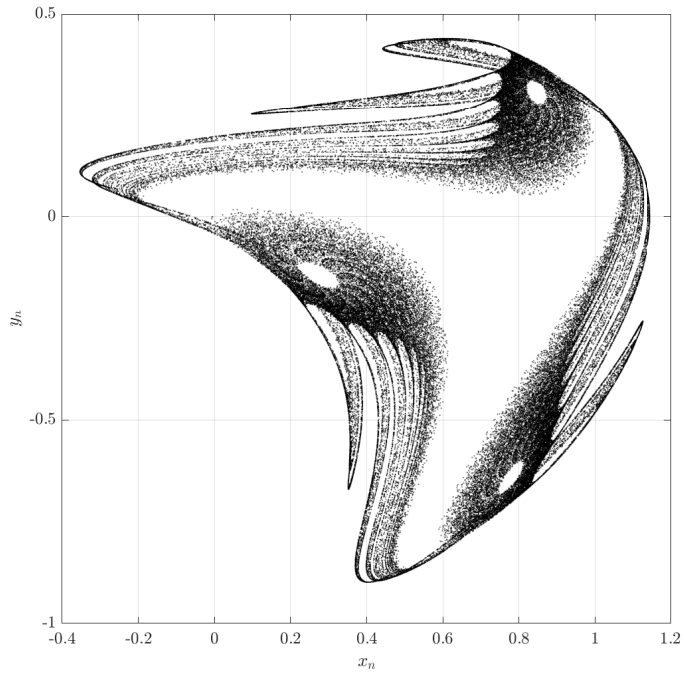
(a) Atractor 1



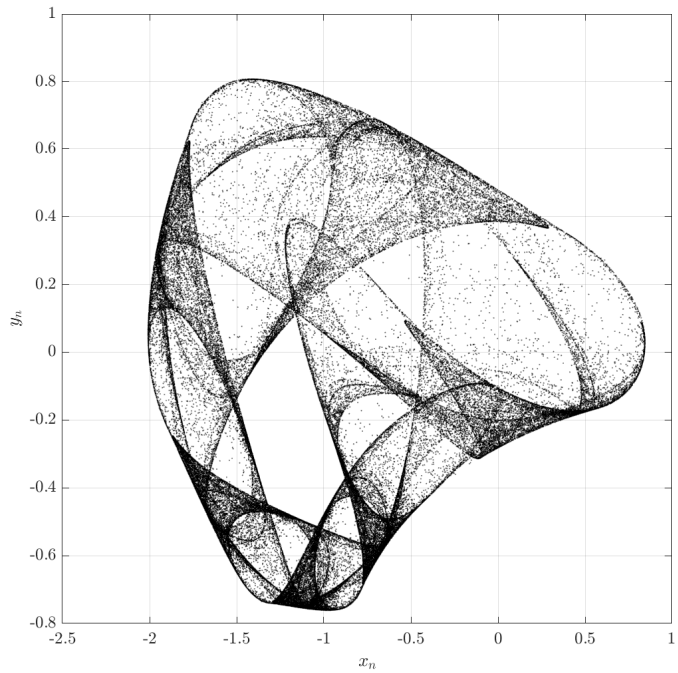
(b) Atractor 2

Mapa caótico

Figura 14: Diferentes atractores caóticos del mapa bidimensional A_3 y A_4.



(a) Atractor 3



(b) Atractor 4

Mapa caótico

¿Qué condiciones iniciales generan caos? - Dominio de atracción

- No se saben las posibles variaciones que pueden ocurrir al modificar las condición iniciales x_0 y y_0.
- No se conoce el rango que pueden tener las condiciones iniciales en el que se asegure que exista el caos.

Mapa caótico

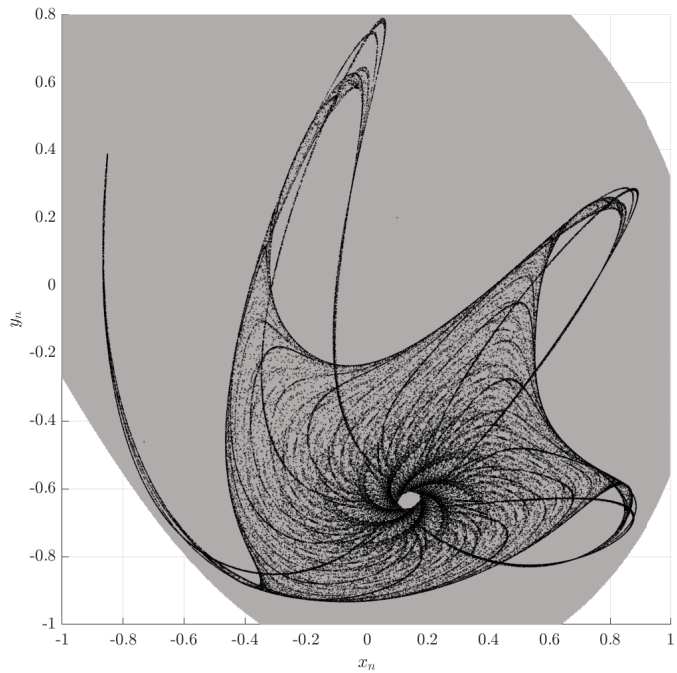
Pasos para calcular dominio de atracción

Podemos calcularlo numéricamente siguiendo los siguientes pasos:

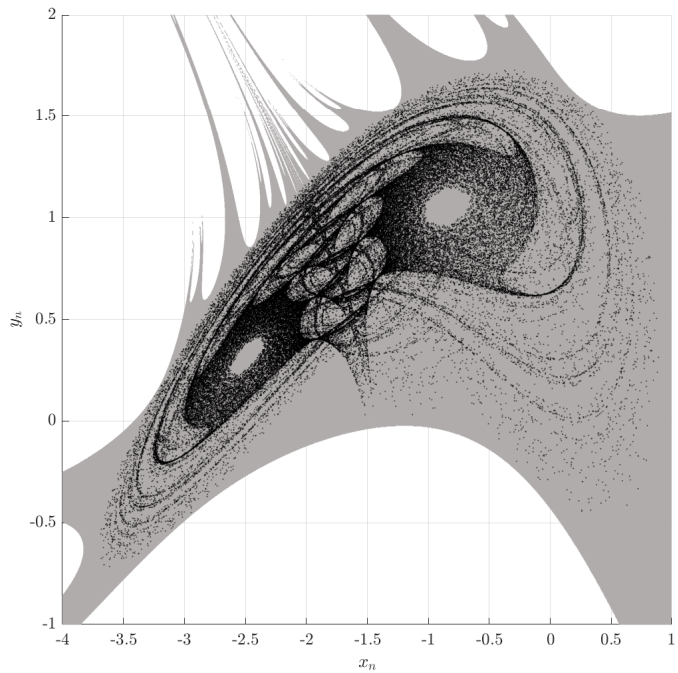
1. Elegir un rango para $x_0 \in [x_{\text{izq}}, x_{\text{der}}]$ y $y_0 \in [y_{\text{izq}}, y_{\text{der}}]$ y un tamaño de paso h donde se va a realizar el análisis.
2. Iterar el mapa unos cientos de veces para cada uno de los posibles valores de x_0 y y_0 dentro del rango y el tamaño de paso h seleccionado. Para cada combinación almacenar todas las iteraciones en un vector.
3. Comprobar cada vector en busca de puntos fijos, divergencia y de ser posible ciclos límite.
4. En una matriz de tamaño, $m \times n$, donde m es el número de elementos en el rango de y_0 y n el número de elementos del rango de x_0, escribir 1 si esta dentro del rango que produce caos o 0 si esta fuera.

Mapa caótico

Figura 16: Diferentes atractores caóticos y dominios de atracción del mapa bidimensional A_1 y A_2.



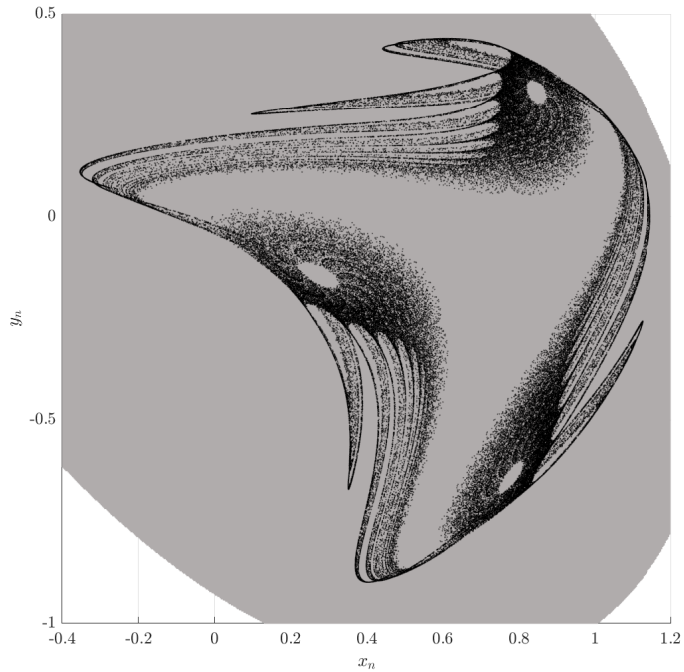
(a) Atractor 1



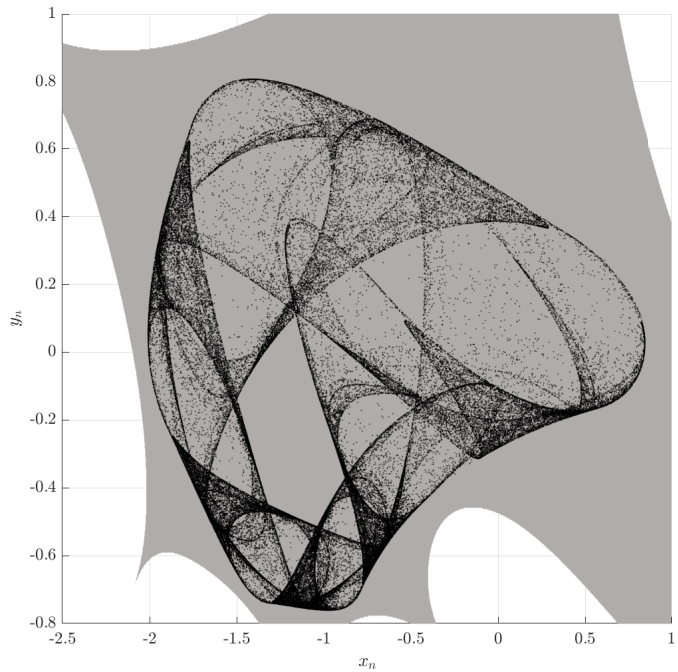
(b) Atractor 2

Mapa caótico

Figura 18: Diferentes atractores caóticos y dominios de atracción del mapa bidimensional A_3 y A_4.



(a) Atractor 3



(b) Atractor 4

Mapa caótico

Rangos de condición inicial seleccionados seleccionados

Tabla 4: Rangos usados para la condición inicial para cada uno de los atractores.

Atractor	Rango de valores para x_0 y y_0
1	$x_0 \in [-0.5, \quad 0.5]$, $y_0 \in [-0.5, \quad 0.5]$
2	$x_0 \in [-1.0, \quad 0.0]$, $y_0 \in [\quad 0.0, \quad 1.0]$
3	$x_0 \in [\quad 0.0, \quad 1.0]$, $y_0 \in [-0.6, \quad 0.4]$
4	$x_0 \in [-1.5, -0.5]$, $y_0 \in [-0.5, \quad 0.5]$

Mapa caótico

DRNG utilizando un mapa caótico bidimensional

En [16] se utiliza el mapa caótico bidimensional para diseñar un generador de números psedoaleatorios utilizando una arquitectura de punto fijo y extracción de 16 bits aleatorios haciendo uso de la operación mod 256.

$$\begin{aligned} x_{n+1} &= a_1 + a_2 x_n + a_3 x_n^2 + a_4 x_n y_n + a_5 y_n + a_6 y_n^2 \\ y_{n+1} &= a_7 + a_8 x_n + a_9 x_n^2 + a_{10} x_n y_n + a_{11} y_n + a_{12} y_n^2 \end{aligned}$$

$$s_{n+1} = \{x_{n+1} \text{ mod } 256, y_{n+1} \text{ mod } 256\} \quad (3)$$

Mapa caótico

Representaciones de números decimales en dispositivos digitales

- Punto flotante
  - Mayor rango dinámico.
  - Útil en algoritmos complejos.
- Punto fijo
  - Más rápida.
  - Menos recursos.
  - $X(a, b)$, a bits enteros, b bits fraccionarios, 1 bit de signo.
  - Rango $[-2^a, 2^a - 2^{-b}]$.
  - Precisión 2^{-b}.

Es preferible utilizar punto fijo en FPGAs.

Mapa caótico

Análisis de punto fijo

Una vez definido el sistema y los atractores a utilizar hay que seleccionar el formato de punto fijo óptimo para cada atractor. Como ejemplo utilizaremos el Atractor 1.

- Esta acotado en un rango de $x_n \in [-0.866, 0.8915]$ y $y_n \in [-0.933, 0.7896]$.
- Las operaciones que pueden generar los números con magnitud más grandes son la combinación de algunas de las sumas.
- Como $|a_n| < 1$, con la excepción de a_3, después de realizar la multiplicación por cualquiera de ellos el número se vuelve más pequeño.
- Las combinaciones de sumas que puedan generar el número más grande (β), donde a son los bits para parte entera:

$$a = \log_2(\text{abs}(\beta)) \tag{4}$$

Mapa caótico

Simulador de punto fijo en C

Diseño un simulador de punto fijo en lenguaje C para poder comprobar la factibilidad de la arquitectura antes de pasar al diseño en hardware en FPGA. ¿Cuántos bits de precisión se necesitan?.

Tabla 5: Número de bits usados en la implementación de cada uno de los atractores con aritmética de punto fijo.

Atractor	Bits parte entera	Bits parte fraccionaria	Rango	Precisión
1	3	60	$[-8.0, 8.0]$	8.6739×10^{-19}
2	4	59	$[-16.0, 16.0]$	1.7347×10^{-18}
3	4	59	$[-16.0, 16.0]$	1.7347×10^{-18}
4	3	60	$[-8.0, 8.0]$	8.6739×10^{-19}

Mapa caótico

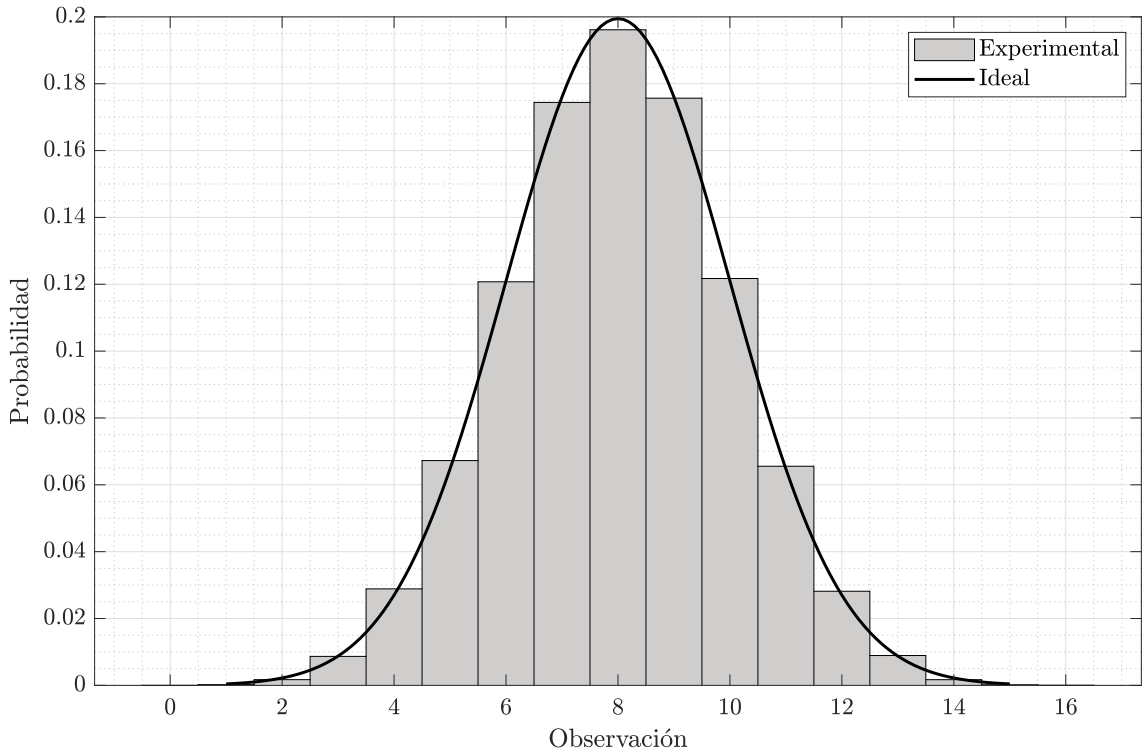


Figura 20: Distribución de 100 millones de palabras de 16 bits.

Mapa caótico

Implementación en FPGA

Si reescribimos la ecuación (1) se pueden eliminar dos multiplicadores más. Es decir la mínima cantidad de operaciones que se requieren para implementar el mapa son 10 sumadores y 11 multiplicadores.

$$\begin{aligned} x_{n+1} &= a_1 + (a_2 + a_3 x_n)x_n + a_4 x_n y_n + (a_5 + a_6 y_n)y_n \\ y_{n+1} &= a_7 + (a_8 + a_9 x_n)x_n + a_{10} x_n y_n + (a_{11} + a_{12} y_n)y_n \end{aligned} \quad (5)$$

Mapa caótico

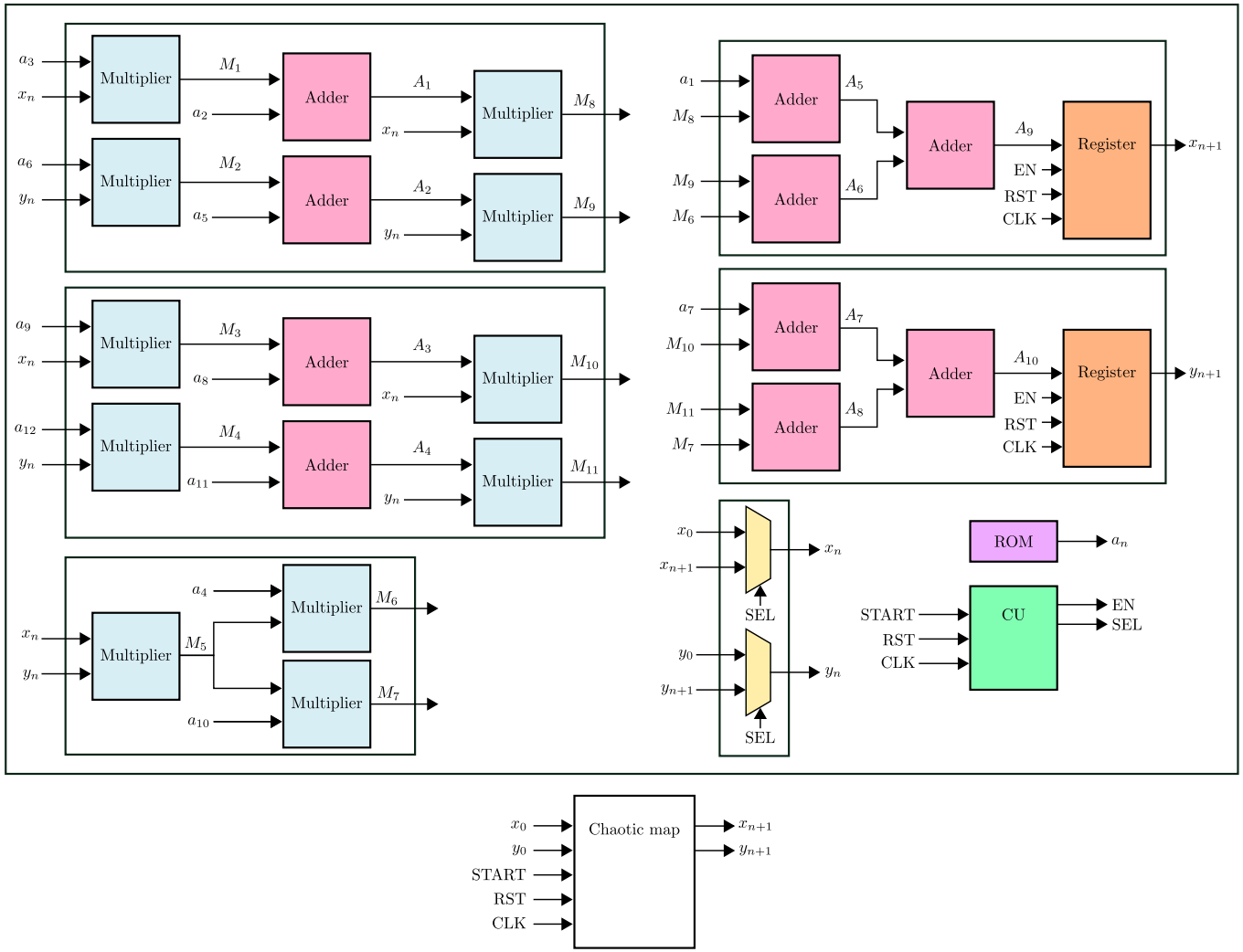


Figura 21: Diagrama de bloques del mapa caótico.

Mapa caótico

Máquina de estados de mapa caótico

- Controlar los multiplexores para seleccionar la condición inicial (SEL).
- Controlar los registros de salida que almacenan la iteración actual (EN).
- La señal START activa el sistema.

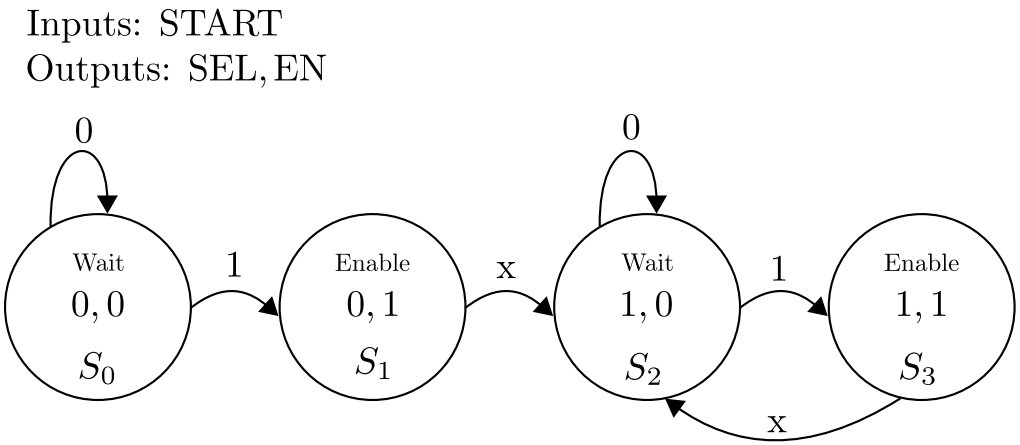


Figura 22: Máquina de estados de mapa caótico.

Mapa caótico

Multiplicadores de una sola constante (SCM)

Al multiplicar por una constante conocida, podemos explotar las propiedades de la multiplicación binaria para obtener un circuito de hardware funcionalmente equivalente con menos recursos lógicos en comparación con un multiplicador genérico. La multiplicación constante puede implementarse como un conjunto de sumas, restas y desplazamientos binarios.

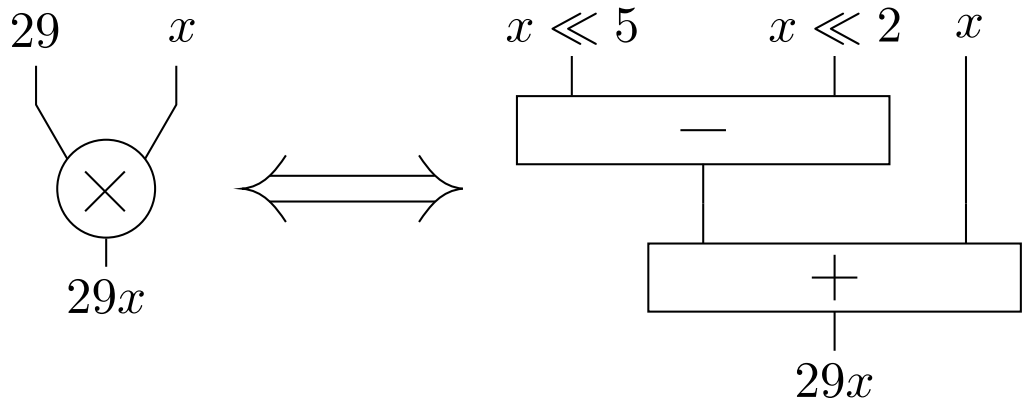


Figura 23: Ejemplo de multiplicación de una sola constante.

Generador de semillas

Descripción de generador de semillas

Utiliza un registro de corrimiento a la derecha (RSR) que muestrea el ERO-TNG y compara si 64 bits son extraídos y si se encuentran dentro del rango del dominio de atracción del mapa caótico utilizando los parámetros del Atractor 1. Un contador de 5 bits de una sola vuelta asegura que se hayan generado 64 bits y no pasé una semilla incompleta.

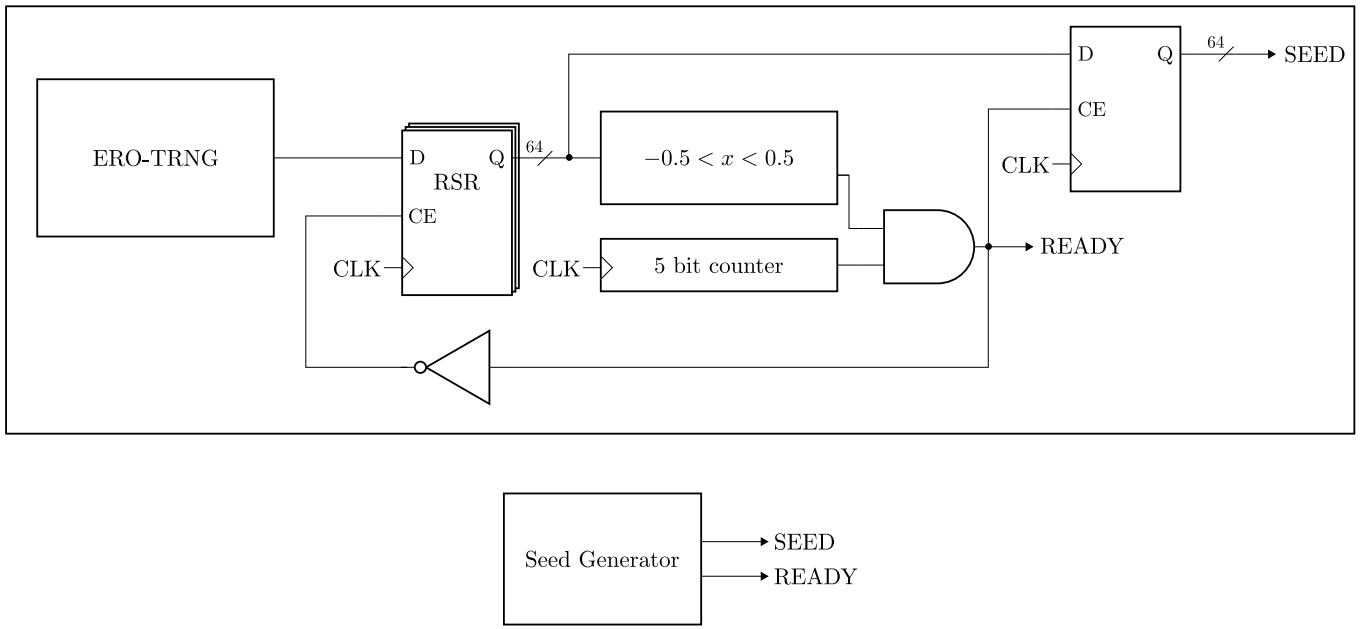


Figura 24: Generador de semilla.

Implementación de TRNG híbrido

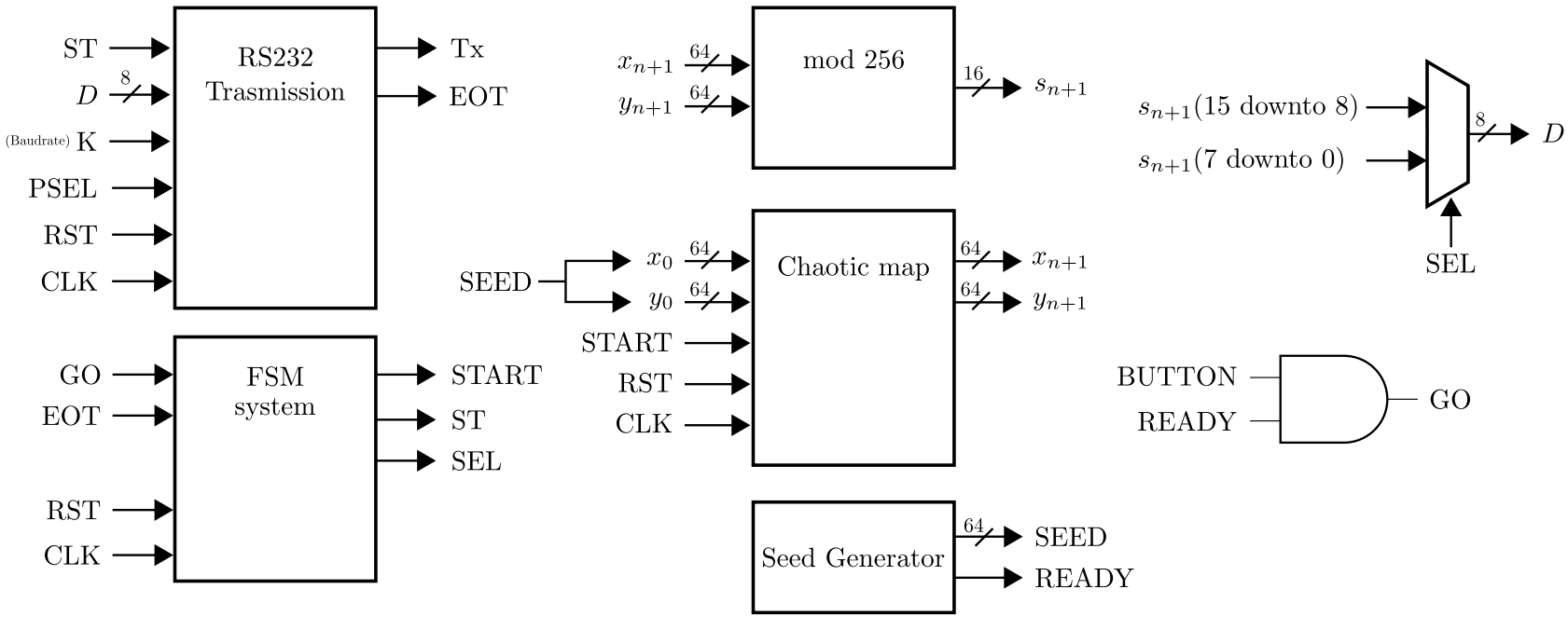


Figura 25: Diagrama de bloques de TRNG híbrido.

Implementación de TRNG híbrido

Máquina de estados de TRNG híbrido

- La señal GO activa la máquina de estados del TRNG híbrido.
- Calcula una iteración del mapa caótico (START).
- Transmite por RS232 los 16 bits aleatorios en dos paquetes de 8 bits.

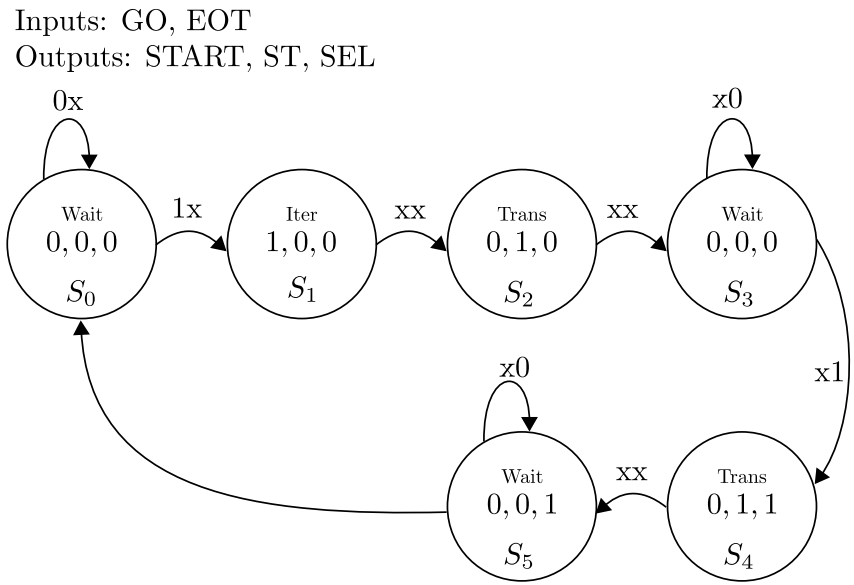


Figura 26: Máquina de estados de TRNG híbrido.

Implementación de TRNG híbrido

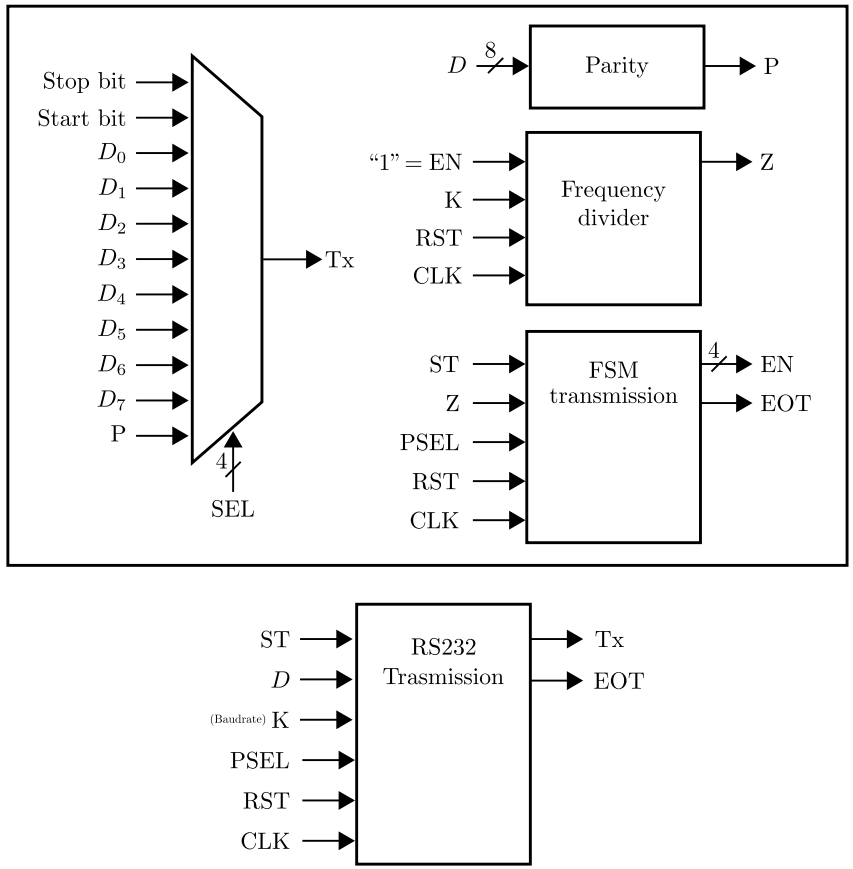


Figura 27: Diagrama de bloques de transmisión RS232.

Implementación de TRNG híbrido

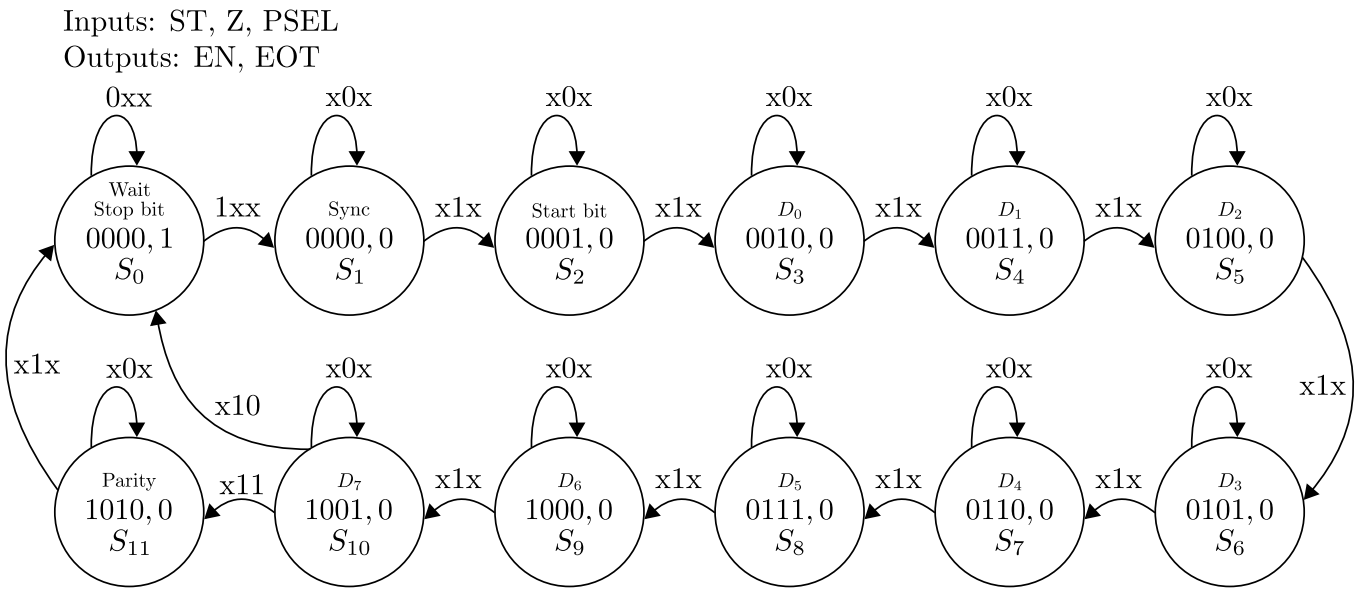


Figura 28: Máquina de estados para la transmisión RS232.

Resultados

Tabla 6: Resultados de la aplicación de las pruebas NIST al TRNG híbrido implementado en aritmética de punto fijo con 100 secuencias de un millón de datos.

Test name	p-value	%
Frequency	0.383827	0.99
Block frequency	0.108791	1.00
Cumulative sums	0.401199	1.00
Runs	0.971699	0.99
Longest Run	0.759756	1.00
Rank	0.383827	0.99
FFT	0.383827	0.99
NonOverlapping template	0.480298	0.99
Overlapping template	0.883171	0.99
Universal	0.574903	0.99
Approximate entropy	0.759756	0.98
Random excursions	0.265539	0.99
Random excursions variant	0.312463	0.99
Serial	0.595604	1.00
Linear complexity	0.574903	1.00

Resultados

Tabla 7: Resultados de la aplicación de las pruebas NIST al TRNG híbrido implementado en aritmética de punto fijo con 1000 secuencias de un millón de datos.

Test name	p-value	%
Frequency	0.587274	0.989
Block frequency	0.796268	0.998
Cumulative sums	0.848047	0.990
Runs	0.614226	0.988
Longest Run	0.660012	0.992
Rank	0.255705	0.990
FFT	0.072514	0.994
NonOverlapping template	0.481082	0.989
Overlapping template	0.143686	0.988
Universal	0.228367	0.988
Approximate entropy	0.786830	0.992
Random excursions	0.447308	0.990
Random excursions variant	0.405096	0.992
Serial	0.124135	0.993
Linear complexity	0.008816	0.989

Resultados

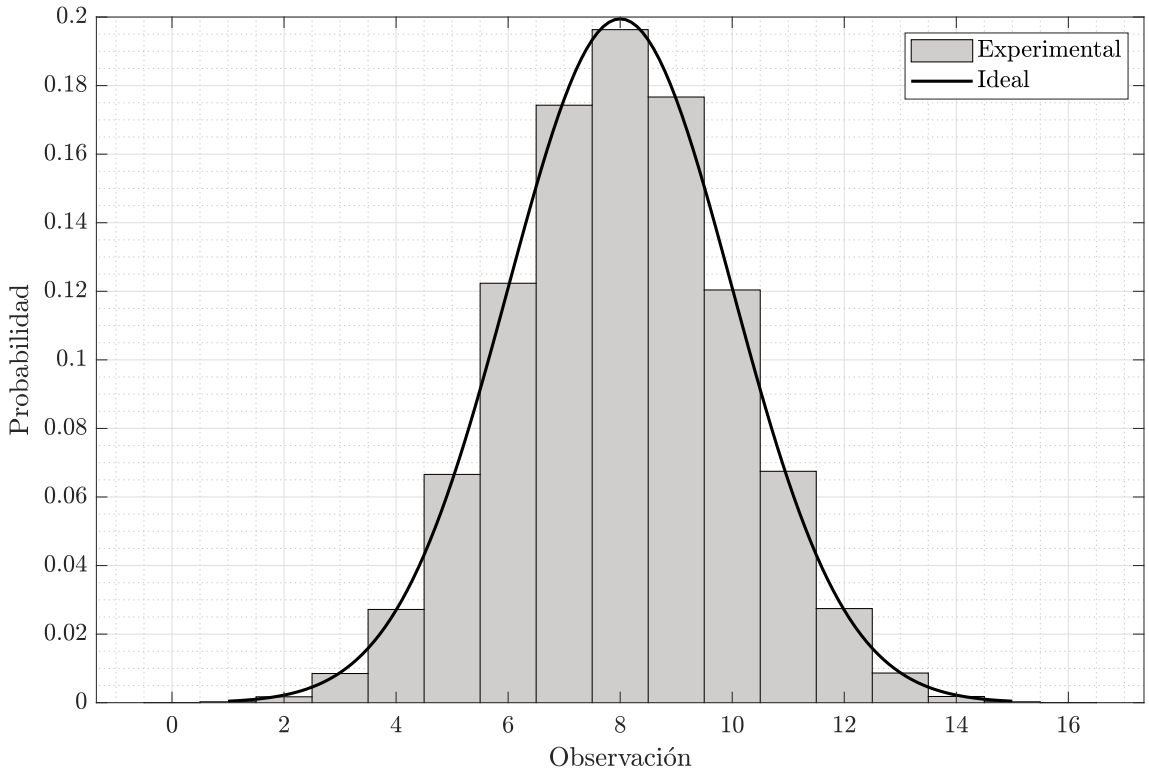


Figura 29: Distribución de 100 millones de palabras de 16 bits experimentales.

Resultados

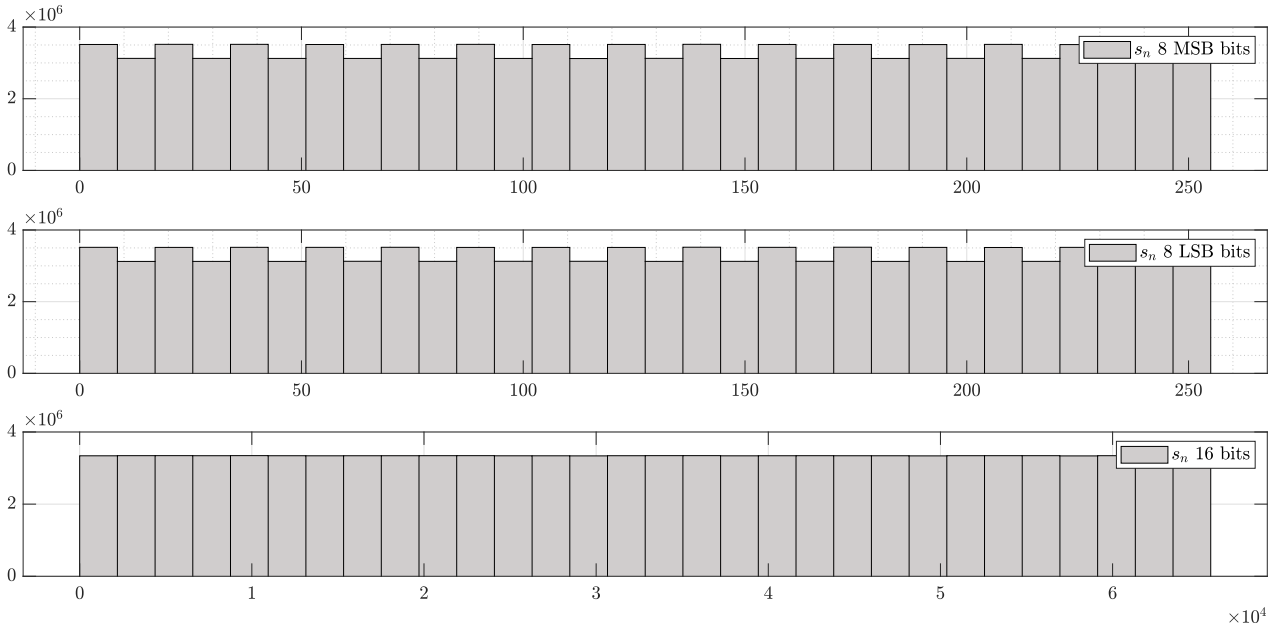


Figura 30: Histograma de resultados experimentales primeros 100 mil secuencias binarias.

Resultados

Tabla 8: Uso de recursos del TRNG híbrido con multiplicadores completos.

Recursos	Utilización	Disponibles	Utilización %
LUTs	17767	20800	85.42
FF	148	41600	0.36
DSP	90	90	100.0

Tabla 9: Uso de recursos del TRNG híbrido con multiplicadores de una sola constante en parámetros a_n.

Recursos	Utilización	Disponibles	Utilización %
LUTs	1469	20800	7.06
FF	146	41600	0.35
DSP	80	90	88.89

La velocidad obtenida por el TRNG híbrido fue de 533.33 Mbit/s, es ligeramente superior a sistemas similares las cuales rondan los 400 Mbit/s.

Conclusión

Conclusiones

- El núcleo ERO-TRNG tienen un modelo estocástico bien definido y forma parte de los núcleos aprobados por la AIS20/31.
- Se tienen 12 parámetros diferentes para configurarse lo que lo hace muy versátil (diferentes atractores).
- La implementación utiliza un 88 % de los DPS y tan solo un 7 % de los LUTs de la FPGA, no obstante considerando que la FPGA utilizada es de bajos recursos, para FPGAs más grandes las cuales tienen de 4 a 5 recursos, este sistema no representa un gran consumo de área.
- El TRNG híbrido pasó todas las pruebas NIST y el análisis estadístico demostró distribuciones uniformes.
- La velocidad obtenida por el TRNG híbrido fue de 533.33 Mbit/s, es ligeramente superior a sistemas similares las cuales rondan los 400 Mbit/s.

Conclusión

Conclusiones

- Se diseño un simulador en C para comprobar arquitecturas de punto fijo.
- Es repetible en diferentes familias de FPGAs de Xilinx.

Bibliografía I

[1] S. H. Strogatz, *Nonlinear dynamics and Chaos*.
Addison-Wesley Pub., 1994.

[2] J. C. Sprott, *Chaos and time-series analysis*.
Oxford University Press, 2003.

[3] M. S. Turan, E. Barker, J. Kelsey, K. A. McKay, M. L. Baish, and M. Boyle, "Recommendation for the entropy sources used for random bit generation," tech. rep., jan 2018.

[4] W. Killmann and W. Schindler, "A proposal for: Functionality classesfor random number generators, version 2.0," 2011.

[5] B. Badrignans, J. L. Danger, V. Fischer, G. Gogniat, and L. Torres, eds., *Security Trends for FPGAS*.
Springer Netherlands, 2011.

Bibliografía II

[6] O. Petura, *True random number generators for cryptography : Design, securing and evaluation*.
Theses, Université de Lyon, Oct. 2019.

[7] O. Petura, U. Mureddu, N. Bochard, V. Fischer, and L. Bossuet, "A survey of AIS-20/31 compliant TRNG cores suitable for FPGA devices," in *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, IEEE, aug 2016.

[8] M. Baudet, D. Lubicz, J. Micolod, and A. Tassiaux, "On the security of oscillator-based random number generators," vol. 24, pp. 398–425, oct 2010.

[9] P. Kohlbrenner and K. Gaj, "An embedded true random number generator for FPGAs," in *Proceedings of the 2004 ACM/SIGDA 12th international symposium on Field programmable gate arrays*, ACM, feb 2004.

Bibliografía III

[10] B. Sunar, W. Martin, and D. Stinson, "A provably secure true random number generator with built-in tolerance to active attacks," *IEEE Transactions on Computers*, vol. 56, pp. 109–119, jan 2007.

[11] M. Varchola and M. Drutarovsky, "New high entropy element for FPGA based true random number generators," pp. 351–365, Springer Berlin Heidelberg, 2010.

[12] P. Haddad, V. Fischer, F. Bernard, and J. Nicolai, "A physical approach for stochastic modeling of TERO-based TRNG," in *Lecture Notes in Computer Science*, pp. 357–372, Springer Berlin Heidelberg, 2015.

[13] A. Cherkaoui, V. Fischer, A. Aubert, and L. Fesquet, "A self-timed ring based true random number generator," in *2013 IEEE 19th International Symposium on Asynchronous Circuits and Systems*, IEEE, may 2013.

Bibliografía IV

[14] V. Fischer and M. Drutarovský, "True random number generator embedded in reconfigurable hardware," in *Cryptographic Hardware and Embedded Systems - CHES 2002*, pp. 415–430, Springer Berlin Heidelberg, 2003.

[15] J. Sprott, "Automatic generation of strange attractors," *Computers & Graphics*, vol. 17, pp. 325–332, may 1993.

[16] L. G. D. la Fraga, C. Mancillas-López, and E. Tlelo-Cuautle, "Designing an authenticated hash function with a 2d chaotic map," *Nonlinear Dynamics*, vol. 104, pp. 4569–4580, may 2021.