Universidad Zaragoza
1542

Trabajo Fin de Grado

# Funciones físicas no-clonables para ciberseguridad

Grado en Física

Autor:

Raúl Aparicio Téllez

Directores:

Miguel García Bosque
Guillermo Díez Señorans

Facultad de Ciencias

Junio 2022

# Índice

# Índice de figuras



# Índice de tablas

## Lista de acrónimos

| | |
|---|---|
| **CDF** | Cumulative Distribution Function |
| **EER** | Equal Error Rate |
| **FAR** | False Acceptance Rate |
| **FPGA** | Field Programmable Gate Array |
| **FRR** | False Rejection Rate |
| **FSM** | Finite State Machine |
| **HD** | Hamming Distance |
| **ILA** | Integrated Logic Analyzer |
| **LUT** | Look-Up Table |
| **MUX** | Multiplexer |
| **NVM** | Non-Volatile Memory |
| **PL** | Programmable Logic |
| **PMU** | Power Management Unit |
| **PUF** | Physical Unclonable Function |
| **RO** | Ring Oscillator |
| **SCL** | Serial Clock |
| **SDA** | Serial Data |
| **SoC** | System on Chip |
| **SRAM** | Static Random Access Memory |
| **UART** | Universal Asynchronous Receiver-Transmitter |
| **VIO** | Virtual Input/Output |
| **XADC** | Xilinx Analog-to-Digital Converter |

## Resumen

En este trabajo se implementa una función física no-clonable (PUF) basada en osciladores de anillo. En primer lugar, se investigan diferentes métodos para obtener información sobre la frecuencia de los osciladores. A continuación, se examina la dependencia de la frecuencia de los osciladores con su posición en una FPGA. Se proponen cinco estrategias diferentes de selección de un pequeño conjunto de ubicaciones a partir de un gran conjunto de ubicaciones y se estudia la PUF construida en términos de unicidad, reproducibilidad e identificabilidad. Este análisis demuestra que la calidad de la PUF es altamente dependiente de la posición donde se implementan los osciladores. Finalmente, se analiza la modificación de los resultados en respuesta a cambios en el entorno de operación tales como variaciones de temperatura o del voltaje de alimentación de la placa.



**Palabras clave:** ciberseguridad, FPGA, función física no-clonable, oscilador de anillo.


# 1. Introducción

El auge de las tecnologías de la información y la comunicación ha conllevado grandes cantidades de datos confidenciales que deben estar encriptados para terceros. Este tipo de información es almacenada en bases de datos que pueden ser vulnerables frente a ciberataques. De aquí surge la necesidad de crear sistemas de seguridad que permitan reducir o resolver problemas relacionados con la identificación y seguridad de los dispositivos. En este contexto, aparecen las funciones físicas no-clonables (PUF).

Estas funciones utilizan variaciones estocásticas inherentes producidas durante el proceso de fabricación para generar claves que autentican unívocamente a cada dispositivo, actuando como si fuesen la "huella dactilar" del mismo. Las PUF generan una secuencia binaria basándose en las características del dispositivo y, por tanto, eliminan la necesidad de guardar información secreta en una memoria. De ahí su gran utilidad en el ámbito de la ciberseguridad, ya que pueden utilizarse para la identificación y autenticación de dispositivos, así como para la generación de claves con fines criptográficos [1].

Por estas razones, las funciones físicas no-clonables han experimentado una creciente popularidad a lo largo de los últimos años. Desde que se acuñó el término PUF en 2001, se han propuesto más de 40 tipos de funciones físicas no-clonables. Algunas de ellas son las *Arbiter PUF*, que comparan el retraso de dos caminos a modo de "carrera digital"; las *SRAM PUF*, que utilizan los bloques SRAM existentes para generar datos específicos del chip o las *Ring-Oscillator PUF*, en las que se centra este trabajo. En él, se han estudiado las frecuencias de varios osciladores en anillo (RO) de cinco inversores situados en 400 ubicaciones diferentes en 35 FPGAs idénticas. En base a ello, se han propuesto cinco estrategias distintas de selección de 101 osciladores para implementar una PUF y se ha analizado la calidad de la PUF en términos de unicidad, reproducibilidad e identificabilidad. Los resultados obtenidos ponen de manifiesto que la interdistancia promedio puede experimentar grandes cambios en función de la posición escogida.

A partir de este trabajo, se ha presentado un artículo en el 17º Congreso Internacional de Investigación de Doctorado en Microelectrónica y Electrónica (PRIME 2022) [2].

# 2. Objetivos

El objetivo principal de este trabajo es analizar el comportamiento de una PUF de oscilador en anillo en función de la posición de los osciladores en una FPGA, proponer diversas estrategias de selección de osciladores y estudiar con cuál se obtiene una PUF de mayor calidad.

Para ello, se han establecido los siguientes objetivos específicos:

- Estudiar qué es una PUF y sus principales propiedades: analizar sus ventajas, desventajas e importancia en el ámbito de la ciberseguridad.
- Implementar osciladores en anillo en una FPGA: estudiar el lenguaje de descripción de hardware Verilog así como diseñar e implementar osciladores en anillo en una FPGA.
- Obtener información sobre la frecuencia de los osciladores: proponer diversas estrategias que permitan caracterizar los osciladores en anillo implementados, estudiar los bloques *Integrated Logic Analyzer* (ILA) y *Virtual Input/Output* (VIO) así como la lectura de datos vía puerto serie.
- Analizar la calidad de la respuesta de la PUF: estudiar las palabras binarias generadas haciendo uso de herramientas estadísticas.
- Examinar la robustez de la PUF analizando su variabilidad frente a cambios ambientales tales como variaciones de temperatura o cambios en el voltaje de alimentación de la FPGA.

# 3. Fundamentos teóricos

## 3.1. ¿Qué son las PUF?

En términos generales, una función física no-clonable es una entidad que utiliza variaciones en el proceso de fabricación de los dispositivos con el fin de generar una respuesta a modo de "huella digital" del dispositivo [3], [4]. Esta respuesta suele ser una palabra binaria. El hecho de que estas variaciones no puedan ser controladas externamente, les aporta este carácter no-clonable. Se puede definir qué es una PUF haciendo una analogía con una huella dactilar:

- Una huella dactilar presenta individualismo: es una característica propia que define a cada ser humano. Del mismo modo, una PUF es una característica específica de cada dispositivo.
- Una huella dactilar es inherente: todo ser humano nace con ella. Análogamente, una PUF está presente en un objeto desde que es creado, como consecuencia de variaciones producidas durante el proceso de fabricación.
- Una huella dactilar es inclonable. Es decir, no es posible crear dos seres humanos con la misma huella dactilar. Incluso dos clones biológicos tendrían distinta huella dactilar. Esta inclonabilidad es la principal propiedad de una PUF.

Dado que una PUF está siempre asociada a un objeto físico, se puede concluir:

*Una PUF es una entidad física que presenta una funcionalidad desafío-respuesta que depende de estructuras físicas difíciles de clonar.*

## 3.2. Ventajas e inconvenientes

En comparación con otras técnicas utilizadas en ciberseguridad, tanto en el ámbito de la autenticación como en el de la generación y almacenamiento seguro de claves, las PUF presentan dos importantes ventajas: la reducción de costes y el incremento de la seguridad [3].

Una gran parte de los sistemas de seguridad utilizados actualmente almacenan sus datos o contraseñas en memorias no volátiles (NVM). Sin embargo, los chips que cuentan con NVM resultan más caros de producir que aquellos que no tienen este tipo de memorias, ya que requieren de procesos adicionales. Además, estas claves deben ser generadas fuera del chip y luego transferirlas al mismo, suponiendo un coste económico adicional y, sobre todo, un problema de seguridad. En cambio, en una PUF esta información se genera directamente en el chip de tal forma que no es necesario almacenar ningún dato en NVM. El hecho de que no se almacena ninguna clave en NVM y solo se regenera cuando es necesaria para un aplicación, aumenta la seguridad de los dispositivos. Las PUF se basan en variaciones incontrolables ocurridas durante el proceso de fabricación. De esta manera, incluso con todos los detalles y controles del proceso de fabricación, los fabricantes no pueden eliminar ni controlar la naturaleza aleatoria de las PUF, alcanzando un alto nivel de seguridad [4].

No obstante, las PUF no son perfectas, en vista de que no es posible obtener en todo momento la misma respuesta para un mismo desafío [3]. Estos errores son aleatorios y se deben a efectos difíciles de controlar tales como: variaciones de temperatura, cambios en el voltaje de alimentación del chip o efectos de envejecimiento del dispositivo. Esta es la principal desventaja de las PUF. Sin embargo, para propósitos de identificación de dispositivos, estos errores no resultan un problema si la palabra de respuesta es lo suficientemente larga.

## 3.3. Métricas de la calidad de una PUF

### 3.3.1. Distancia Hamming

Dadas dos palabras binarias $Y$,$Y'$ con $n$ bits de longitud, se define la distancia Hamming ($HD$) como el número de bits diferentes entre las dos palabras:

$$HD\ (Y;Y') = \sum_{i=1}^{n} Y_i \oplus Y_i' \tag{3.1}$$

donde $Y_i$,$Y_i'$ son los bits i-ésimos de las palabras $Y$,$Y'$ respectivamente y $\oplus$ indica la operación XOR, que proporciona un '1' en la salida siempre que las entradas $Y_i$,$Y_i'$ no coincidan:

$$Y_i \oplus Y_i' = Y_i \text{ XOR } Y_i' = (Y_i \cdot \overline{Y_i'}) + (\overline{Y_i} \cdot Y_i') \tag{3.2}$$

Desde un punto de vista teórico [3], [4], teniendo en cuenta que los elementos de la cadena de bits son variables aleatorias, las distancias Hamming entre las distintas palabras se distribuyen siguiendo una distribución binomial:

$$f_{\text{binomial}}(k;n,p) = \binom{n}{k} p^k\,(1-p)^{n-k} \tag{3.3}$$

donde $n$ se corresponde con el número de bits de la cadena, $k$ con el número de bits no coincidentes y $p$ con la probabilidad de que dos bits que estén en la misma posición sean diferentes.

También resulta útil definir el concepto de distribución de probabilidad acumulada (CDF), que determina la probabilidad de que la variable aleatoria tome un valor menor o igual que $k$:

$$F_{\text{binomial}}(k; n, p) = \sum_{i=0}^{k} f_{\text{binomial}}(i; n, p) \tag{3.4}$$

### 3.3.2. Unicidad y reproducibilidad

La unicidad de una PUF viene descrita por el concepto de interdistancia [4]. Esta variable aleatoria describe la distancia entre dos palabras de bits $Y(x)$,$Y'(x)$ generadas por la misma PUF implementada en FPGAs diferentes ante un mismo desafío $x$:

$$D^{\text{inter}}(x) = \text{dist}\left[Y(x); Y'(x)\right] \tag{3.5}$$

En este trabajo se utiliza como métrica la distancia Hamming y se hablará de Inter-$HD$. Idealmente, la Inter-$HD$ debe ser del 50 %. En un caso real, seguirá una distribución binomial.

Análogamente, la reproducibilidad de una PUF viene descrita por el concepto de intradistancia [4]. Esta variable aleatoria describe la distancia entre dos palabras de bits $Y^j(x)$,$Y'^j(x)$ de una PUF en una misma FPGA $j$ ante un mismo desafío $x$:

$$D^{\text{intra}}_{\text{FPGA }j}(x) = \text{dist}\left[Y^j(x); Y'^j(x)\right] \tag{3.6}$$

Ya que se utiliza como métrica la distancia Hamming, se hablará de Intra-$HD$. Idealmente, la Intra-$HD$ debe ser del 0 %. En un caso real, suele seguir una distribución binomial.

### 3.3.3. Identificación

En un sistema de identificación basado en una PUF, la respuesta generada se compara con una lista de respuestas ya registradas. En el momento en el que el sistema encuentra una respuesta en la lista con una distancia menor o igual a un cierto umbral $t_{id}$ previamente establecido, el dispositivo es identificado con la entrada coincidente en la lista [4]. En este proceso de comparación pueden darse varios escenarios:

- Verdadera aceptación: La entidad candidata es la misma que produjo la respuesta de la lista y se puede reproducir la respuesta con una distancia menor o igual que el umbral.
- Verdadero rechazo: La entidad candidata no es la misma que produjo la respuesta de la lista y se observa una distancia mayor que el umbral de aceptación.
- Falsa aceptación: La entidad candidata no es la misma que produjo la respuesta de la lista, pero la distancia es menor o igual que el umbral.
- Falso rechazo: La entidad candidata es la misma que produjo la respuesta de la lista, pero no se puede obtener con una distancia menor o igual que el umbral.

En estos sistemas no interesa que se produzcan falsas aceptaciones o falsos rechazos. La probabilidad de que un intento de identificación tenga como resultado cualquiera de estos dos casos viene determinada por la *tasa de falso rechazo* ($FRR$) y la *tasa de falsa aceptación* ($FAR$).

Por lo tanto, ambos deben ser lo más pequeños posibles, pero resulta evidente que no se pueden minimizar al mismo tiempo. Ambos errores dependen de dónde se coloca el umbral, de

tal forma que si se aumenta el valor de $t_{id}$, la $FRR$ disminuye y la $FAR$ aumenta. Análogamente, si se disminuye el valor de $t_{id}$, la $FRR$ aumenta y la $FAR$ disminuye. Resulta necesario alcanzar un compromiso entre que el sistema sea robusto y seguro al mismo tiempo. Ambas variables se definen a partir del umbral $t_{id}$ según las Ecuaciones 3.7 y 3.8.

$$FAR = F_{\text{binomial}}(t_{id}; n, \hat{p}_{\mathcal{P}}^{\text{inter}}) \tag{3.7}$$

$$FRR = 1 - F_{\text{binomial}}(t_{id}; n, \hat{p}_{\mathcal{P}}^{\text{intra}}) \tag{3.8}$$

donde $\hat{p}_{\mathcal{P}}^{\text{inter}}$ y $\hat{p}_{\mathcal{P}}^{\text{inter}}$ son los estimadores binomiales de la PUF.

El umbral para el cual se produce la intersección de ambas curvas se denomina *equal error threshold* ($t_{EER}$) y la tasa de error correspondiente *equal error rate* ($EER$). El $EER$ proporciona información sobre la calidad del sistema de identificación. Para distribuciones discretas, se escoge el valor del umbral de tal forma que $FAR$ y $FRR$ sean lo más parecido posible (Ecuación 3.9) y se define $EER$ como el mayor de los dos errores (Ecuación 3.10).

$$t_{EER} = argmin_t\{\max\{FAR(t_{id}), FRR(t_{id})\}\} \tag{3.9}$$

$$EER = max\{FAR(t_{EER}), FRR(t_{EER})\} \tag{3.10}$$

Este parámetro de identificabilidad ($EER$) resulta el más importante de todos, más incluso que la reproducibilidad y la unicidad, ya que al fin y al cabo la aplicación práctica para la cual se va a utilizar una PUF es para la identificación y autenticación de dispositivos.

## 3.4. PUF de oscilador en anillo

Un oscilador en anillo (Figura 3.1) es un sistema formado por un número impar de inversores lógicos colocados en forma de anillo, de tal forma que la salida del último inversor está conectado con la entrada del primero. De este modo, la salida oscila entre dos valores de voltaje.

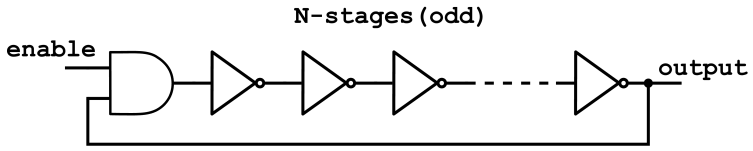



**Figura 3.1:** Esquema de un oscilador en anillo.

A la salida se obtiene una señal periódica con una frecuencia inversamente proporcional al tiempo de propagación medio de un inversor $t$ y al número de inversores $n$:

$$f = \frac{1}{2tn} \tag{3.11}$$

Típicamente se añade una puerta AND actuando como "interruptor" del circuito, de tal forma que si `enable=0` es estable (no se producen oscilaciones) y si `enable=1` es "astable" (sí oscila).

Una PUF de oscilador en anillo (RO-PUF) es un tipo de función física no-clonable que aprovecha la diferencia en las frecuencias de los osciladores producidas durante el proceso de fabricación para crear una huella digital única que identifique a cada dispositivo [3], [4]. Ya que utiliza únicamente puertas lógicas, puede ser implementada directamente en una FPGA. A este tipo de PUF que no requiere de pasos de procesamiento adicionales se le conoce como *Intrinsic*

*PUF*. Además, no es necesario que el cableado de los inversores sea simétrico, como por ejemplo en la *Arbiter PUF*.

Una RO-PUF consta básicamente de dos componentes: osciladores en anillo idénticos y contadores de frecuencia. La señal periódica se lleva a un contador que se encarga de contar el número de ciclos en un cierto intervalo de tiempo. El valor del contador será una medida directa de la frecuencia de la señal. Debido a los cambios producidos durante el proceso de fabricación, aunque los osciladores sean idénticos, tendrán frecuencias algo distintas. Por tanto, se podrían utilizar estas frecuencias como "respuesta" de la PUF. Este sistema es sensible frente a cambios en el entorno de operación. Para conseguir una respuesta más "estable", lo que se hace habitualmente es comparar las frecuencias de dos osciladores idénticos (técnica de medida compensada) [3], [5].

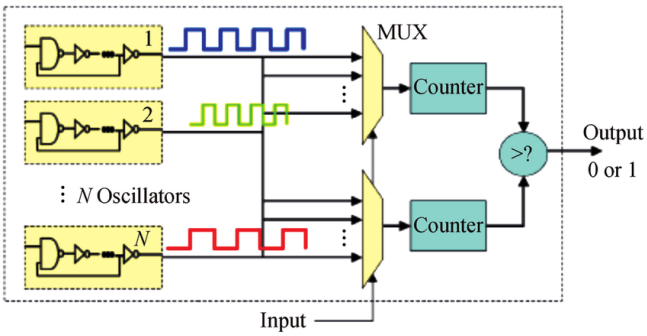



**Figura 3.2:** Arquitectura de una PUF de oscilador en anillo [6].

Con el objetivo de minimizar la influencia de estas alteraciones, Suh y Devadas [5] proponen un modelo de RO-PUF basado en un conjunto de $n$ osciladores idénticos que se llevan a dos multiplexores de $n$ entradas cada uno. A la salida de cada multiplexor se coloca un contador y se comparan las frecuencias de los osciladores. En este caso, se realizarán comparaciones por parejas (Figura 3.2), de tal forma que si $f_i$ es la frecuencia del oscilador $i$ y $f_j$ la del oscilador $j$, se producirá un bit $r_{ij}$ de salida tal que:

$$r_{ij} = \begin{cases} 1 \text{ si } f_i > f_j \\ 0 \text{ si } f_i < f_j \end{cases} \tag{3.12}$$

Ya que las frecuencias están influenciadas por las variaciones del proceso, el bit resultante será aleatorio. Si disponemos de $n$ osciladores, el número de parejas que se podrán formar será:

$$\binom{n}{2} = \frac{n(n-1)}{2} \tag{3.13}$$

Si se realizan todas las comparaciones posibles se obtiene un conjunto de comparaciones que no son linealmente independientes. Para ejemplificar esto, se imagina la comparación de las frecuencias de tres osciladores $a, b, c$ tal que $f_a < f_b$ y $f_b < f_c$. Por consiguiente, la frecuencia del oscilador $a$ será menor que la del $c$, $f_a < f_c$. Habitualmente no se utilizan todas las comparaciones para definir la PUF. Existen varias arquitecturas posibles en función de la estrategia de comparación utilizada [7]: *1-out-of-2* (se realizan comparaciones por parejas sin repetición: el 1 con el 2, el 3 con el 4 ...), *all pairs* (todos con todos) o $n-1$. La topología $n-1$ consiste en comprar los osciladores de tal forma que se repita un oscilador en cada comparación: el oscilador 1 con el 2, el 2 con el 3, el 3 con el 4 etc. En este trabajo se ha optado por esta topología.

# 4. Obtención de la frecuencia de los osciladores

El propósito de este apartado es estudiar diversas estrategias que permitan caracterizar frecuencialmente un conjunto de osciladores en anillo implementados en una FPGA.

## 4.1. Descripción del sistema de medida

Para la realización de este trabajo se han utilizado 35 placas PYNQ-Z2 que se identificarán con una letra y un número (por ejemplo, FPGA-B5). Cada una de ellas dispone de una SoC zynq-7000 formada por una FPGA Artix7 y un microprocesador ARM Cortex-A9 [8]. Esta FPGA dispone de 13 300 *slices*. Cada *slice* cuenta con 4 LUTs de 6 entradas y con 8 flip-flops. Cada una de las porciones o *slices* se expresa mediante una coordenada $X_iY_j$ $i, j = 0, 1, \ldots$ Además, las cuatro LUTs de cada *slice* se numeran con las letras A, B, C y D tal y como se muestra en la Figura 4.1.b. Por simplicidad, se identificará a cada oscilador a partir de la primera coordenada de la *slice* donde se ha implementado.

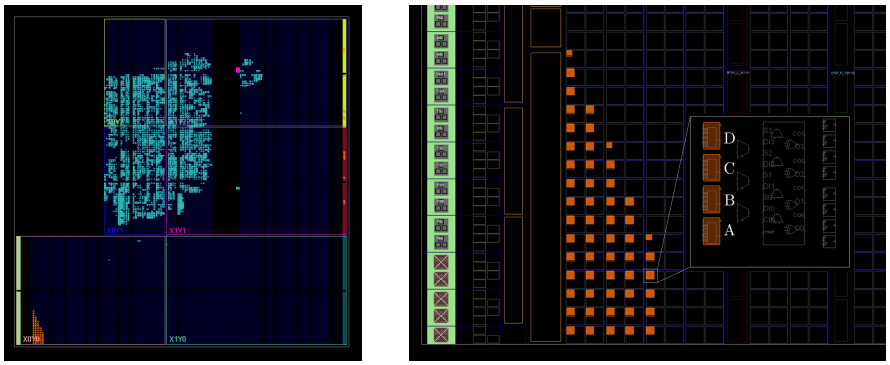
Para hallar la frecuencia de un oscilador, se cuenta el número de ciclos de la señal ($N_{\text{osc}}$) existentes para un cierto intervalo de tiempo y se compara con el del reloj interno de la FPGA ($N_{\text{clock}}$). En este caso, el intervalo de tiempo seleccionado es el equivalente a $N_{\text{clock}} = 10^6$ ciclos de reloj (8 ms). Por tanto, conocido que el reloj interno opera a 125 MHz, es posible hallar la frecuencia de cada oscilador según la Ecuación 4.1.

$$\text{Frecuencia oscilador (MHz)} = \frac{N_{\text{osc}}}{N_{\text{clock}}} \cdot 125 \text{ MHz} \tag{4.1}$$

A continuación, se proponen diversas formas de obtener información frecuencial de osciladores en anillo implementados en la FPGA.

## 4.2. Monitorización de señales internas con bloques ILA

El bloque ILA (*Integrated Logic Analyzer*) es un núcleo de propiedad intelectual (*IP Core*) que permite monitorizar señales internas del diseño. Cada señal monitorizada supone un requisito de almacenamiento, de tal forma que la implementación de este bloque supone un elevado consumo de memoria (Figura 4.1.a). Esta herramienta permite visualizar múltiples señales que pueden ser combinadas a través de una condición única de disparo o *trigger*.




(a) (b)

**Figura 4.1:** a) Implementación del bloque ILA en la FPGA. b) Posición de los osciladores.

Para estudiar su funcionamiento, se ha medido el número de ciclos de las señales de cinco osciladores de 61, 51, 41, 31 y 21 inversores situados en las posiciones $X_iY_0$ $i = 0, 1, 2, 3, 4$ respectivamente (Figura 4.1.b) de la FPGA-B9. El bloque ILA permite leer el número de ciclos del oscilador almacenados en la variable $N_{\text{osc}}$ y así hallar su frecuencia. Se ha medido la frecuencia de cada oscilador diez veces y calculado la media junto con su error (Ecuación 4.2 y 4.3).

$$\text{Media muestral:} \quad \bar{f} = \frac{1}{N}\sum_{i=0}^{N} f_i \tag{4.2}$$

$$\text{Error estándar de la media:} \quad \sigma_{\bar{f}} = \frac{\sigma}{\sqrt{N}} \tag{4.3}$$

donde $\sigma$ es la desviación estándar muestral de la media y $N$ el número de medidas (en este caso diez). Los resultados obtenidos se muestran en la Tabla 4.1.

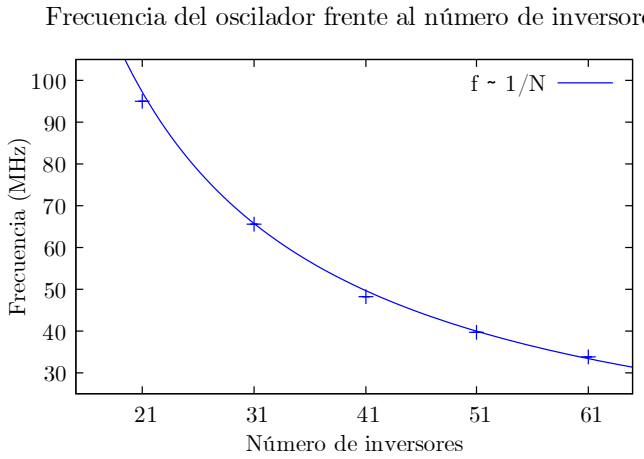



**Figura 4.2:** Frecuencia frente al número de inversores.

| $N_{\text{inv}}$ | $\bar{f}$ (MHz) |
|---|---|
| 21 | $94{,}979 \pm 0{,}010$ |
| 31 | $65{,}689 \pm 0{,}004$ |
| 41 | $48{,}269 \pm 0{,}053$ |
| 51 | $39{,}822 \pm 0{,}003$ |
| 61 | $33{,}829 \pm 0{,}002$ |

**Tabla 4.1:** Frecuencia media de los osciladores según el número de inversores.

Las frecuencias medias obtenidas tienen unos errores relativos inferiores al 0,2 %. Además, en la Figura 4.2 se observa que la frecuencia aumenta de forma inversamente proporcional al número de inversores, corroborando así lo esperado según la Ecuación 3.11. Este bloque resulta útil como sistema de depuración de errores, ya que permite conocer el valor de algunas de las variables del sistema en tiempo real. Sin embargo, resulta ineficiente para medir la frecuencia de un gran número de osciladores, ya que habría que definir tantas variables como osciladores tuviese el sistema.

## 4.3. Control de señales internas con VIO

En una RO-PUF, solo es necesario conocer simultáneamente la frecuencia de dos osciladores para obtener el bit de salida. Una solución al problema anterior consiste en llevar cada una de las señales de los osciladores a dos multiplexores A y B que permitan seleccionar qué dos osciladores se van a comparar. Esta es la estructura común de una PUF de oscilador en anillo (Figura 3.2).

El bloque VIO (*Virtual Input/Output*) es un IP Core personalizable que permite monitorizar y controlar señales internas de la FPGA en tiempo real. El bloque observa el valor de las señales entrantes y permite establecer el valor de la señales salientes. Este núcleo está sincronizado con el diseño y, por tanto, las componentes del bloque VIO están sometidas a las mismas restricciones que el reloj de la FPGA. Se crearán dos multiplexores A y B que permitan seleccionar

el oscilador que se va a comparar. Además, un registro almacena la variable con el selector del multiplexor (MUX). Esta señal es interceptada por el bloque VIO y modificada antes de entrar a su correspondiente MUX. Hay que destacar que también es necesario implementar el bloque ILA con el objetivo de conocer el valor de cada uno de los ciclos de reloj.

Para verificar su funcionamiento se han definido cinco osciladores con el mismo número de inversores, situados en las posiciones $X_iY_0$ $i = 0, 1, 2, 3, 4$ y se han obtenido las frecuencias de parejas de RO $X_iY_0$,$X_{i+1}Y_0$. A continuación, se ha obtenido el bit de salida resultado de la comparación de la frecuencia de ambos osciladores según la Ecuación 3.12. En este caso se ha utilizado la FPGA-B9. Los resultados obtenidos se muestran en la Tabla 4.2.

| RO-A | RO-B | $f_A$ (MHz) | $f_B$ (MHz) | $r_{AB}$ |
|---|---|---|---|---|
| $X_0Y_0$ | $X_1Y_0$ | 96,15 | 97,90 | 0 |
| $X_1Y_0$ | $X_2Y_0$ | 97,91 | 95,96 | 1 |
| $X_2Y_0$ | $X_3Y_0$ | 95,95 | 100,61 | 0 |
| $X_3Y_0$ | $X_4Y_0$ | 100,64 | 97,86 | 1 |

**Tabla 4.2:** Comparaciones realizadas con el bloque VIO y bit de salida $r_{AB}$.

Los resultados obtenidos indican que para osciladores con el mismo número de inversores, la frecuencia es distinta en función de la posición donde se ha implementado. Se ha repetido este experimento en distintas FPGAs y se ha observado que la respuesta de la PUF ("0101") es recurrente en todos los casos. Es decir, osciladores situados en *slices* $X_iY_j$ con $i$ par tienen una frecuencia distinta que aquellos situados en *slices* con $i$ impar.

El bloque VIO ha permitido reducir el número de señales de salida, ya que únicamente han sido necesarias dos, correspondientes a los dos multiplexores y la señal de reloj. Sin embargo, la selección de los osciladores que se van a comparar se realiza de forma manual, retrasando significativamente el proceso de medida. Nuevamente, este bloque solo resultará útil como sistema de depuración de errores. Además, en los diseños definitivos, este bloque no se utiliza.

## 4.4. Lectura de la frecuencia vía puerto serie

A continuación, se propone la implementación de una máquina de estados (FSM) sobre una FPGA. Para ello, se han definido los siguientes elementos: una señal de reloj, un bus de entrada que le da las instrucciones a la máquina y dos buses de salida (uno de ellos da información sobre el estado en el que se encuentra la máquina y el otro permite extraer los datos para leerlos).

Una vez definida la FSM, se ha medido el número de ciclos de cada oscilador. Dicho valor se saca vía puerto serie UART, para lo cual se ha utilizado el cliente PuTTY. De tal forma que si hay algún *byte* de entrada, PuTTY lo escribe en el puerto serie y si hay algún *byte* de salida lo escribe por pantalla. Asimismo, se han definido dos instrucciones diferentes con las letras 'c' y 'r' respectivamente para controlar la máquina de estados programada en el procesador del SoC. Esta FSM se comunica con el diseño implementado en la lógica programable (PL) de tal forma que cada vez que se pulsa la tecla 'r' el programa selecciona el oscilador situado en $X_0Y_0$ y cada vez que se pulsa la letra 'c' el programa selecciona el siguiente oscilador, hasta que se han seleccionado todos y vuelve al primero de ellos ($X_0Y_0$). De este modo, es posible extraer las frecuencias de

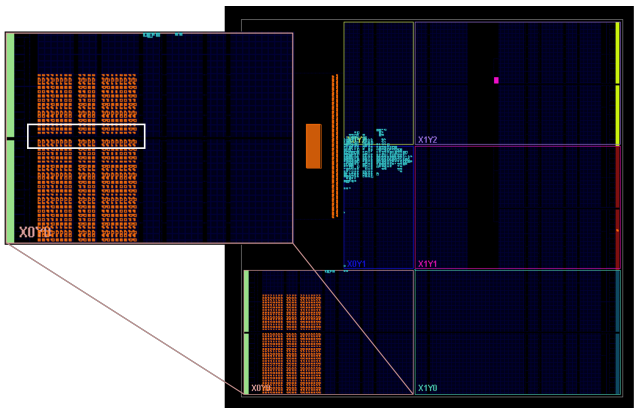
múltiples osciladores sin tener que seleccionarlos manualmente. De aquí en adelante, ésta será la forma a partir de la cual se leerán las frecuencias de los osciladores implementados en la FPGA.

# 5. Resultados

El propósito de esta sección es estudiar el comportamiento de una PUF basada en un oscilador en anillo en función de la posición de los osciladores en la FPGA. Para ello, se implementarán varios osciladores en distintas posiciones de la placa y se analizará la respuesta de la PUF haciendo uso de herramientas estadísticas. Se propondrán diversas estrategias de selección de osciladores y se discutirá la calidad de la PUF en términos de unicidad, reproducibilidad e identificabilidad para cada uno de los casos.

## 5.1. Descripción del experimento

Se han definido 400 osciladores formados por una puerta AND y 5 inversores cada uno. Por tanto, se han necesitado 6 LUT para definir cada uno de ellos: las A,B,C,D de la *slice* $X_iY_j$ y las A,B de la *slice* $X_iY_{j+1}$ $i = 0, 1, \ldots, 19$ $j = 0, 2, \ldots, 38$. Los osciladores se han dispuesto en forma de matriz 20×20 tal y como se muestra en la Figura 5.1.a. Además, se ha asignado un índice a partir del cual se identificará cada oscilador, comenzando por el 1 en la esquina inferior izquierda hasta el 400 en la esquina superior derecha. El esquema de la Figura 5.1.b refleja el índice asignado a cada oscilador y su ubicación aproximada en la FPGA.




(a)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 381 | 382 | 383 | 384 | 385 | ··· | 399 | 400 |
| 361 | 362 | 363 | 364 | 365 | ··· | 379 | 380 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋱ | ⋮ | ⋮ |
| 81 | 82 | 83 | 84 | 85 | ··· | 99 | 100 |
| 61 | 62 | 63 | 64 | 65 | ··· | 79 | 80 |
| 41 | 42 | 43 | 44 | 45 | ··· | 59 | 60 |
| 21 | 22 | 23 | 24 | 25 | ··· | 39 | 40 |
| 1 | 2 | 3 | 4 | 5 | ··· | 19 | 20 |

(b)

**Figura 5.1:** a) Implementación del diseño en la FPGA. En la zona donde se han implementado los osciladores se observa una separación física entre las *slices* $X_iY_{23}$ y $X_iY_{24}$. b) Definición de los índices en función de su posición en la FPGA.

## 5.2. Análisis de la frecuencia de los osciladores

A continuación, se han tomado cinco medidas de la frecuencia de cada uno de los osciladores implementados en la FPGA-B5 y se ha calculado la frecuencia promedio de cada oscilador. Seguidamente, se ha representado la frecuencia media de cada uno de ellos junto con su error frente al índice asignado (Figura 5.2.a). Además, se ha obtenido un mapa de frecuencias de la placa en función de la posición espacial de las *slices* donde se han implementado cada uno de los 400 osciladores (Figura 5.2.b). Los resultados obtenidos ponen de manifiesto una cierta

variabilidad en la frecuencia del oscilador en función de su posición en la FPGA tal y como se predecía en la Sección 4.3.

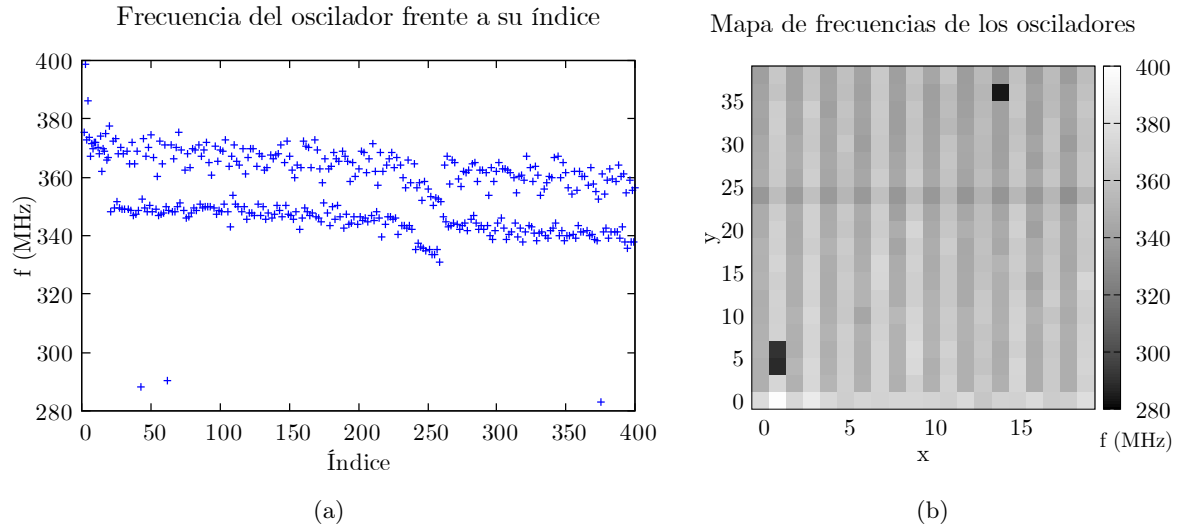



**Figura 5.2:** a) Frecuencia del oscilador frente a su índice. b) Mapa de frecuencias.

Para determinar si este efecto es común al resto de FPGAs, se realizado esta medida en cada una de las 35 FPGAs y se han observado varios efectos comunes a todas las placas:

- En general, los osciladores con índice par presentan una frecuencia superior a los que tienen índice impar. En las FPGAs de Xilinx las coordenadas $X_iY_j$ con $i$ par utilizan *slices* de tipo M, mientras que las que tienen $i$ impar utilizan *slices* de tipo L. Por tanto, este fenómeno podría deberse a que los osciladores con índice impar están implementados en *slices* de tipo M, mientras que los osciladores con índice par están implementados en *slices* de tipo L. La diferencia principal entre ambos tipos de *slices* es que las tipo L (*logic*) solo pueden implementar funcionalidades lógicas. Además de utilizarse para funcionalidades lógicas, las tipo M (*memory*) pueden ser configuradas para implementar memoria distribuida o registros de desplazamiento. Sin embargo, el fabricante no proporciona ningún tipo de información que permita explicar esta diferencia de frecuencias entre osciladores pares e impares. De hecho, en ambos tipos de *slices* se utiliza la misma notación. Esto haría que un diseñador que quisiese implementar una PUF basada en oscilador en anillo comparando osciladores de índice par con aquellos de índice impar obtuviese resultados muy predecibles, obteniendo así una PUF de baja calidad.

- Los osciladores con los índices 42, 62 y 375 presentan una frecuencia claramente inferior al resto, del orden de 100 MHz menos. Este efecto se produce en las 35 placas estudiadas para exactamente las mismas ubicaciones. En el diseño implementado se han observado algunas diferencias en el *routeado* de los osciladores con dichos índices en comparación con el resto de osciladores implementados en la FPGA que podrían explicar este hecho. En la Figura 5.3 se han coloreado manualmente las conexiones de los osciladores con frecuencias inferiores (rosa) con el fin de observar las diferencias con las conexiones de osciladores adyacentes (azul). Zulfikar et al. [9] ya demuestran en uno de sus artículos la importancia de realizar un *routeado* uniforme con el objetivo de obtener una PUF de calidad en términos de unicidad y reproducibilidad. En él se comenta que, ya que el ruido influye en el rendimiento de los

osciladores, si el *routeado* no es uniforme algunos osciladores estarán expuestos a un mayor ruido que otros. Este *routeado* puede realizarse manualmente pero es difícil y requiere de una gran cantidad de tiempo por parte del diseñador.

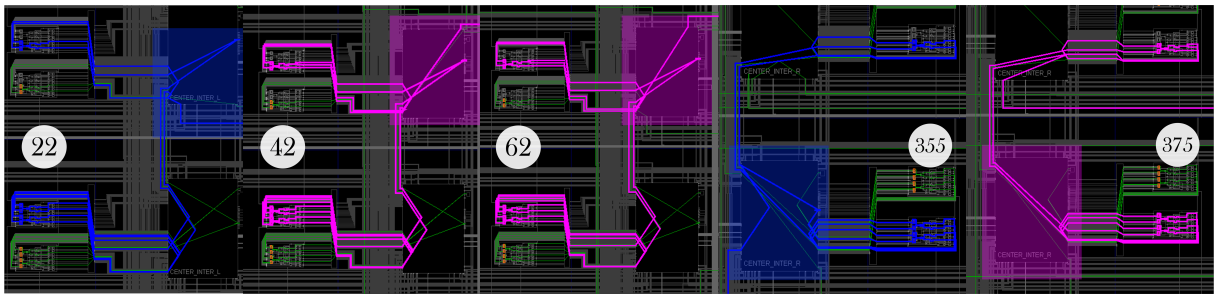



**Figura 5.3:** Diferencias entre las conexiones de los osciladores con una frecuencia inferior 42, 62 y 375 (rosa) y los osciladores adyacentes 22 y 355 (azul).

- A medida que aumenta el índice del oscilador, la frecuencia de los osciladores tiende a disminuir. Esto supone la existencia de una correlación negativa entre la frecuencia de un oscilador y el índice asignado. De este modo, los osciladores más cercanos a la posición $X_0Y_0$ tienen una frecuencia mayor que aquellos más cercanos a la posición $X_{19}Y_{38}$. Esto podría deberse a las variaciones *intra-die* producidas durante el proceso de fabricación. Las variaciones *intra-die* se refieren a cambios en las propiedades del dispositivo que ocurren en un mismo chip, de tal forma que una determinada característica (en este caso, la frecuencia) varía en función de la ubicación en el chip. A pesar de que este efecto ocurre en las FPGAs estudiadas, no es posible asegurar que esto se produzca en todas las FPGAs.
- Los osciladores situados en $X_iY_{j=24}$ $i = 0, 1, \ldots, 19$ tienen una frecuencia menor que aquellos con la misma $i$ y distinta $j$. Esto se explica debido a la existencia de una separación física entre las *slices* $X_iY_{23}$ y $X_iY_{24}$ tal y como se observa en la Figura 5.1.
- Un fenómeno similar se observa en los osciladores situados en $X_iY_0$ $i = 0, 1, \ldots, 19$, que tienden a presentar una frecuencia superior al resto de osciladores de la FPGA. En algunos casos se llegan a observar diferencias del orden de 50 MHz. Esto puede interpretarse como un efecto de borde, ya que dichos osciladores se corresponden con la parte inferior de la placa. Algo similar se observa en varios artículos como el de Maiti et al. [10], que estudiaron la diferencia frecuencial de varios osciladores en anillo en una FPGA Spartan XC3S500E. En él se observa que osciladores situados en unos de los laterales de la FPGA tienden a tener una frecuencia distinta al resto.

En la PUF implementada se ha comparado las frecuencias de dos osciladores $i$ y $j$ siguiendo la topología $n - 1$, de tal forma que si la frecuencia del primer oscilador es mayor que la del segundo ($f_i > f_j$), el bit de salida será un '1'. En caso contrario ($f_i < f_j$), el bit de salida será un '0'. A lo largo de todo el trabajo se ha utilizado la misma arquitectura de RO-PUF. Sin embargo, este análisis es extrapolable a otro tipo de arquitecturas que comparen las frecuencias de osciladores en anillo idénticos.

Frecuentemente, las PUF de oscilador en anillo presentes en la literatura obtienen resultados basados en el hecho de comparar osciladores idénticos. Este tipo de comparaciones demuestra la importancia de seleccionar parejas de osciladores con frecuencias similares, ya que al comparar

un oscilador de índice impar con uno de índice par, la frecuencia del primero será menor que la del segundo y se obtendrá un bit '0' en todos estos casos. Uno de los desafíos a los que se enfrenta un diseñador que pretende implementar una PUF de este tipo, es elegir las localizaciones de los osciladores para que sus frecuencias sean lo más similares posibles. Para resolver este problema, en este trabajo se han estudiado diversas estrategias para seleccionar las localizaciones de un pequeño conjunto de osciladores de anillo a partir de un "gran conjunto" de posibles localizaciones.

## 5.3. Estrategias de selección de los osciladores

Supongamos que un diseñador quisiera implementar una PUF con $n = 101$ osciladores utilizando la estrategia $n - 1$ y pudiera elegir cualquiera de las 400 localizaciones medidas en el apartado anterior para ubicar sus osciladores. ¿Qué localizaciones debería elegir? A continuación, se proponen cinco formas diferentes de selección de dichas localizaciones. Al realizar las comparaciones por parejas, se obtendrá como respuesta una palabra de $n - 1 = 100$ bits. La elección de $n$ se basa en el compromiso que es necesario alcanzar entre el consumo en recursos (área y potencia) de la PUF y la seguridad de la misma. La PUF como solución de seguridad está pensada para sistemas distribuidos (redes de sensores, IoT, ...) que previsiblemente serán pequeños y funcionarán con baterías. Por tanto, resulta crítico el hecho de mantener un bajo consumo y un área reducida. Las cinco estrategias propuestas de selección de osciladores son:

i) *101 primeros:* Consiste en seleccionar los osciladores con los 101 primeros índices (oscilador 1, oscilador 2, ... oscilador 101).

ii) *101 aleatorios:* En esta estrategia se han seleccionado 101 osciladores de forma aleatoria. Por tanto, teniendo en cuenta que el índice del oscilador es una variable aleatoria, desde un punto de vista estadístico, aproximadamente la mitad de los osciladores tendrán índice par y la otra mitad tendrán índice impar.

iii) *101 impares aleatorios:* Se trata de seleccionar 101 osciladores de forma aleatoria entre aquellos que tienen índice impar, es decir, de entre los que tienen una frecuencia menor.

iv) *101 primeros impares:* Consiste en escoger los 101 primeros osciladores que tienen índice impar (oscilador 1, oscilador 3, ... oscilador 201).

v) *101 mejores:* En esta estrategia se busca seleccionar aquellos osciladores con frecuencias similares con el fin de evitar comparaciones de osciladores con índice par con aquellos de índice impar, aumentando así la aleatoriedad de la respuesta de la PUF. Para ello, se ha calculado el promedio (Ecuación 4.2) de las frecuencias de los $N = 200$ osciladores que tienen índice impar. Además, se ha calculado el error de la media muestral de acuerdo a la Ecuación 4.3. A continuación, para diez de las 35 FPGAs, se han seleccionado aquellos osciladores cuya frecuencia estuviese en el intervalo:

$$\left(\bar{f} - \alpha\, \sigma_{\bar{f}}\ ,\ \bar{f} + \alpha\, \sigma_{\bar{f}}\right) \tag{5.1}$$

donde $\bar{f}$ es la frecuencia promedio de los osciladores impares en cada una de las placas y $\sigma_{\bar{f}}$ el error de la media muestral. De este modo, se han obtenido los "mejores osciladores" de cada FPGA. Seguidamente, de entre los osciladores seleccionados, se han escogido aquellos índices comunes a las diez FPGAs. Así, se ha utilizado un pequeño número de FPGAs para determinar cuáles son las mejores localizaciones y, una vez elegidas, se han utilizado esas

localizaciones en todas las FPGAs. El parámetro $\alpha$ de la Ecuación 5.1 es un número real generado mediante prueba y error con el objetivo de obtener al final del proceso exactamente 101 osciladores. En este experimento, el parámetro obtenido ha sido $\alpha = 8{,}86$.

El hecho de haber utilizado en los casos iii), iv) y v) los osciladores con índice impar, busca evitar las comparaciones entre osciladores con índices pares (implementados en *slices* de tipo L) e impares (implementadas en *slices* tipo M) descritas anteriormente. Resulta destacable el hecho de que se obtienen resultados similares si en lugar de trabajar con el dominio de osciladores de índice impar, se utiliza el dominio de osciladores con índice par.

## 5.4. Estudio de la unicidad de la PUF

### 5.4.1. Cálculo de la interdistancia

Tal y como se ha estudiado en la Sección 3.3.2, para determinar la unicidad de una RO-PUF se utiliza como métrica la Inter-$HD$. Con el fin de calcular la Inter-$HD$ en cada una de las cinco estrategias descritas anteriormente, se ha realizado una medida de las frecuencias de los 400 osciladores en las 35 FPGAs. Por consiguiente, se dispondrá de $N(N-1)/2 = 35(35-1)/2 = 595$ parejas.

Aunque la respuesta consta únicamente de 100 bits, y por tanto son necesarios 101 osciladores, se ha medido la frecuencia de todos los osciladores implementados y luego se han realizado las comparaciones utilizando un programa en C. Mediante este sistema, es posible comparar las estrategias descritas entre sí, ya que todas las frecuencias se corresponden con una misma medida. En la Tabla 5.1 se muestra la Inter-$HD$ promedio obtenida en cada uno de los cinco casos junto con el error de la media muestral.

| Estrategia | Inter-$HD$ (%) |
|---|---|
| *101 primeros* | $3{,}29 \pm 0{,}06$ |
| *101 aleatorios* | $19{,}22 \pm 0{,}16$ |
| *101 impares aleatorios* | $42{,}02 \pm 0{,}25$ |
| *101 primeros impares* | $45{,}01 \pm 0{,}25$ |
| *101 mejores* | $45{,}93 \pm 0{,}25$ |

**Tabla 5.1:** Inter-$HD$ promedio para cada estrategia.

Considerando la distancia Hamming como métrica de la distancia, las respuestas aleatorias perfectamente uniformes deberían tener una interdistancia promedio del 50 %. En concreto, la Inter-$HD$ debería ser una distribución binomial con $p = 0{,}5$, pero este hecho será abordado en la Sección 5.4.2. Sin embargo, tanto en la Figura 5.7 como en la Tabla 5.1 se observa que todas las interdistancias experimentales promedio se encuentran por debajo del 50 %, incluso si se considera el error estándar de la media. Además, la interdistancia cambia drásticamente en función de la estrategia que se ha utilizado:

- *101 primeros:* Utilizando esta estrategia se ha obtenido una interdistancia muy baja, cercana al 3 %. Esto se debe a que se están comparando sistemáticamente osciladores con índice impar (y por tanto, que tienen una frecuencia menor) con aquellos de índice par (que tienen una mayor frecuencia). Por consiguiente, al ser un fenómeno que ocurre sistemáticamente en las 35 FPGAs, tienden a obtenerse los mismos bits de salida.

- *101 aleatorios:* En este caso se sigue obteniendo una interdistancia baja, cercana al 20 %, comparada con la Inter-$HD$ esperada. Esto se debe a que al seleccionar 101 osciladores de forma aleatoria, en aproximadamente el 50 % de los casos se estará comparando un oscilador de índice impar con uno de índice par. Por tanto, la mitad de los bits de la palabra de respuesta serán predecibles.
- *101 impares aleatorios:* Al seleccionar 101 osciladores impares de forma aleatoria, la interdistancia mejora considerablemente (hasta un 42 % aproximadamente) acercándose más al valor esperado. Esto se debe a que se están seleccionando osciladores con frecuencias similares y por tanto aumenta la aleatoriedad en cada una de las comparaciones. En cambio, ya que a medida que aumenta el índice del oscilador su frecuencia disminuye, debido a la existencia de una correlación negativa tal y como se ha comentado en la Sección 5.2, los osciladores con índice impar más alto tenderán a tener una frecuencia menor en comparación con los de índice impar más bajo. Esto hace que varias comparaciones den como resultado un '1'. En esta estrategia se ha observado que un 55,0 % de los bits de la respuesta de la PUF son un '1'. Esta propiedad se conoce como "sesgo" de la respuesta.
- *101 primeros impares:* Utilizando esta estrategia se están seleccionando osciladores con frecuencias similares entre sí, ya que todos ellos pertenecen al dominio de frecuencias bajas. Además, al escoger los 101 primeros osciladores impares en lugar de realizar una selección aleatoria, se reduce significativamente el efecto de la correlación negativa. De hecho, en esta estrategia un 52,8 % de los bits de la respuesta de la PUF son un '1' frente al 55,0 % que se obtenía con el método anterior. Estos dos fenómenos hacen que aumente el grado de aleatoriedad y aumente la interdistancia. Hay que destacar que, en general, al utilizar esta estrategia podrían estar escogiéndose algunos osciladores que tuviesen una frecuencia muy diferente al resto (tal y como ocurría con los osciladores de índice 42, 62 y 375). En este hipotético caso, sería tarea del diseñador asegurarse de que esto no ocurra. En este trabajo, al estar seleccionando únicamente los 101 primeros impares, estos tres osciladores no tienen ningún efecto sobre esta estrategia.
- *101 mejores:* Con este método se obtiene la interdistancia más elevada y cercana al valor teórico esperado. Por tanto, en términos de unicidad, se trata de la mejor de las cinco estrategias analizadas. Este resultado no resulta extraño, pues se debe a que al seleccionar los osciladores comunes con frecuencia más cercana a la media, se están escogiendo aquellos con las frecuencias más similares entre sí. Por consiguiente, esto aumenta el grado de aleatoriedad y proporciona una clave más segura desde el punto de vista de la unicidad. El principal inconveniente de este método es que requiere un estudio previo de las frecuencias de los osciladores en función de su posición en la placa. Sin embargo, hay que destacar que el propio uso de una PUF en un escenario real pasa por una fase de "enrollment", en la que se crea una base de datos de desafíos-respuestas para cada instancia PUF antes de ser lanzadas. Por tanto, podría aprovecharse esta fase para realizar un estudio de las frecuencias de los osciladores y aumentar así la calidad de la PUF.

### 5.4.2. Ajuste a una distribución binomial

Con el fin de verificar que las distancias Hamming de las distintas respuestas dan como resultado una distribución binomial (Sección 3.3.1), se ha ajustado la Inter-$HD$ a una función con la forma de la Ecuación 3.3, teniendo en cuenta la longitud de la palabra $n = 100$. Idealmente,

se esperaría que la interdistancia siguiese una distribución binomial de parámetro $p = 0{,}5$; ya que se están comparando respuestas de una misma PUF implementada en dos FPGAs distintas y, en principio, debería existir la misma probabilidad de obtener un bit '0' que un '1' en la respuesta.

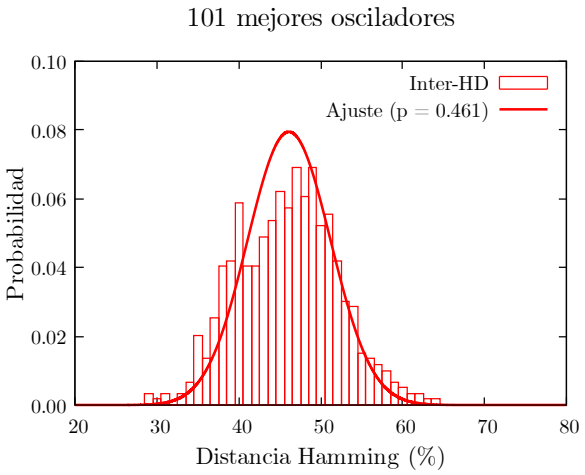



**Figura 5.4:** Ajuste de la Inter-$HD$ a una binomial (caso *101 mejores*).

| Estrategia | $p$ |
|---|---|
| *101 primeros* | 0,035 |
| *101 aleatorios* | 0,194 |
| *101 impares aleatorios* | 0,422 |
| *101 primeros impares* | 0,446 |
| *101 mejores* | 0,461 |

**Tabla 5.2:** Parámetro $p$ del ajuste de la Inter-$HD$ a una binomial para cada estrategia.

En la Figura 5.4 se observa que la interdistancia experimental obtenida para el caso *101 mejores* se ajusta correctamente a una distribución binomial con $n = 100$ y $p = 0{,}461$. Análogamente, la interdistancia en el resto de estrategias también se corresponde con una binomial, aunque con distinto parámetro $p$ (Anexo A). En la Tabla 5.2 se muestra el parámetro $p$ obtenido en el ajuste en la binomial para cada caso estudiado. En ella se observa que las estrategias *101 primeros* y *101 aleatorios* presentan una interdistancia promedio claramente más alejada del valor esperado.

Una causa de que la interdistancia promedio sea más baja, es que las respuestas tienen un mayor número de '1' que de '0'. Además, de los tres casos restantes, el mejor resultado se obtiene para el caso *101 mejores*. Esto es consistente con las explicaciones aportadas anteriormente donde osciladores con frecuencias similares suponían una mejor respuesta. Esta comparación entre las distribuciones binomiales obtenidas se muestra en la Figura 5.5.

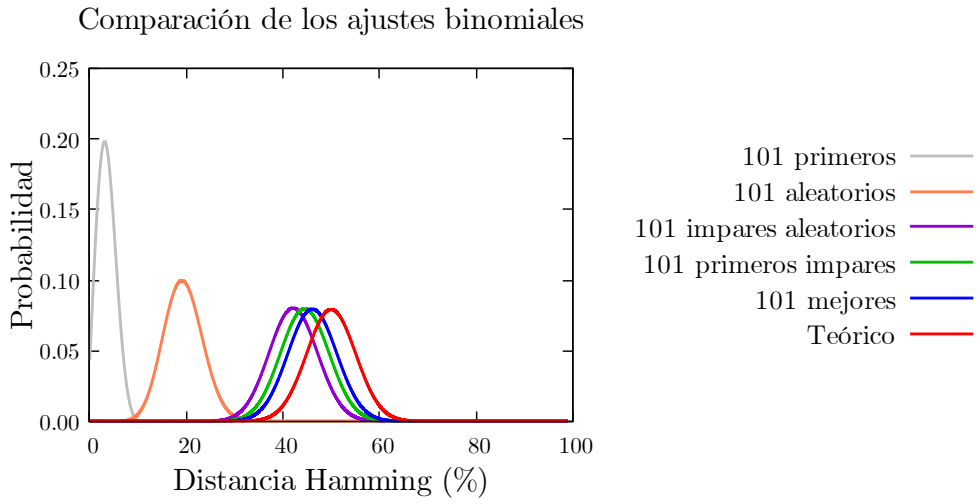



**Figura 5.5:** Comparación de los ajustes binomiales de la Inter-$HD$.

### 5.4.3. Distancia de Kolmogorov-Smirnov

El test de Kolmogorov-Smirnov [11] es una prueba estadística que permite determinar la bondad del ajuste de una determinada distribución experimental a otra distribución teórica esperada. En este trabajo se utiliza el estadístico de Kolmogorov-Smirnov ($D_{\text{K-S}}$) para estudiar cuál es la mejor estrategia en términos de unicidad. Las mejores estrategias tendrán una Inter-$HD$ promedio más cercana al 50 % y una menor $D_{\text{K-S}}$. Para ello, se ha calculado la distribución de probabilidad acumulada de las binomiales en cada uno de los cinco casos y se ha comparado con la CDF de la distribución binomial teórica (Figura 5.6). Esto permitirá realizar una descripción cuantitativa a diferencia de la descripción cualitativa realizada en los dos apartados anteriores.

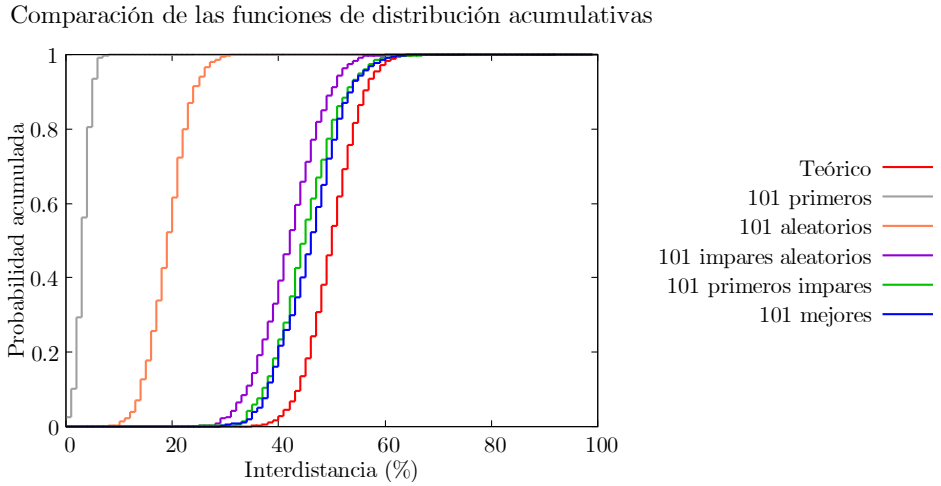



**Figura 5.6:** Comparación de las funciones de distribución acumuladas.

La distancia de Kolmogorov-Smirnov (Ecuación 5.2) se define como la máxima distancia vertical entre la función de distribución acumulada experimental $\hat{F}_n(x_i)$ y la distribución teórica $F_0(x)$. Esta variable proporciona información sobre la diferencia entre las dos distribuciones y permite cuantificar la similitud de la ambas (experimental y teórica).

$$D_{\text{K-S}} = \max_{0 \leq i \leq 100} |\hat{F}_n(x_i) - F_0(x)| \tag{5.2}$$

En la Figura 5.6 se observa que las estrategias *101 primeros* y *101 aleatorios* tienen una distancia de Kolmogorov-Smirnov máxima, lo que implica que ambas distribuciones son completamente diferentes en comparación con la distribución teórica. En los tres casos restantes la distancia K-S es claramente inferior. De hecho, llega a ser mínima para la estrategia *101 mejores*. Esto verifica que las mejores estrategias presentan interdistancias promedio más cercanas al 50 % y por tanto, distancias K-S más bajas.

| Estrategia | $D_{\text{K-S}}$ |
|---|---|
| *101 primeros* | 1,0000 |
| *101 aleatorios* | 0,9999 |
| *101 impares aleatorios* | 0,5294 |
| *101 primeros impares* | 0,3722 |
| *101 mejores* | 0,2813 |

**Tabla 5.3:** Distancias de Kolmogorov-Smirnov según la estrategia utilizada.

## 5.5. Estudio de la reproducibilidad de la PUF

Para determinar la reproducibilidad de una RO-PUF se utiliza como métrica la Intra-$HD$. Para realizar el cálculo de la intradistancia en cada uno de los casos anteriores, se han realizado 100 medidas de la frecuencia de los 400 osciladores en 5 FPGAs distintas con identificadores A5, B5, B9, C1 y C2. Por tanto, se dispondrá de $100(100-1)/2 = 4950$ parejas por cada placa. En la Tabla 5.4 se observan los resultados obtenidos para las distintas placas en cada una de las cinco estrategias junto con el promedio y su error.

| Estrategia | A5 | B5 | B9 | C1 | C2 | Intra-$HD$ (%) |
|---|---|---|---|---|---|---|
| *101 primeros* | 0,00 | 0,60 | 0,00 | 0,52 | 1,31 | $0{,}49 \pm 0{,}54$ |
| *101 aleatorios* | 0,53 | 2,00 | 0,48 | 0,15 | 1,04 | $0{,}84 \pm 0{,}72$ |
| *101 impares aleatorios* | 0,33 | 1,18 | 0,98 | 1,28 | 0,94 | $0{,}94 \pm 0{,}37$ |
| *101 primeros impares* | 2,09 | 1,57 | 1,61 | 0,83 | 2,53 | $1{,}72 \pm 0{,}64$ |
| *101 mejores* | 1,61 | 1,03 | 0,52 | 1,43 | 2,34 | $1{,}39 \pm 0{,}68$ |

**Tabla 5.4:** Intra-$HD$ promedio en función del caso estudiado.

En una PUF con reproducibilidad perfecta se debería obtener una Intra-$HD$ promedio del 0 %, ya que estaríamos comparando las respuestas de una misma PUF ante el mismo desafío. En este caso, se ha observado que la intradistancia varía considerablemente en función de la FPGA seleccionada. Por este motivo, se ha calculado la Intra-$HD$ promedio en 5 FPGAs diferentes. En ninguno de los casos se observa una intradistancia del 0 %, lo que supondría que la PUF tendría una reproducibilidad perfecta, aunque en todos ellos se obtiene una intradistancia baja. Esto se debe al ruido térmico. Al estar seleccionando osciladores con frecuencias similares, pequeñas variaciones en el entorno de operación, como cambios de temperatura o en el voltaje de alimentación, producirán que algunos bits cambien de '0' a '1' (y viceversa), aumentando así el valor de la Intra-$HD$.

En las estrategias *101 primeros*, *101 aleatorios* y *101 impares aleatorios* la intradistancia promedio toma un valor menor, inferior al 1 %. Por tanto, son los casos que presentan una mejor reproducibilidad. Sin embargo, tal y como se ha estudiado en el análisis anterior, son las tres estrategias con una peor unicidad, lo que hace que no sean las mejores formas de implementar una RO-PUF. En las estrategias *101 primeros impares* y *101 mejores* se obtienen intradistancias mayores. Comparando estas dos estrategias, se observa que el caso *101 mejores* presenta una mayor interdistancia (mejor unicidad) y una menor intradistancia (mejor reproducibilidad).

## 5.6. Identificabilidad

En la Figura 5.7 se han representado las dos métricas (Intra-$HD$ e Inter-$HD$) de forma conjunta junto con su ajuste a curvas binomiales. En todos los casos observamos un cierto grado de identificabilidad, ya que la intradistancia obtenida es más pequeña que la interdistancia medida. Sin embargo, para el caso *101 primeros* (Figura 5.7.a) ambas curvas se superponen claramente. Esto supone un problema de identificación ya que una medida que cae en la superposición de ambas curvas puede ser resultado de comparar una misma FPGA u otra diferente, dando lugar a posibles errores de falso rechazo y falsa aceptación. En el resto de estrategias no se observa este problema "a simple vista".

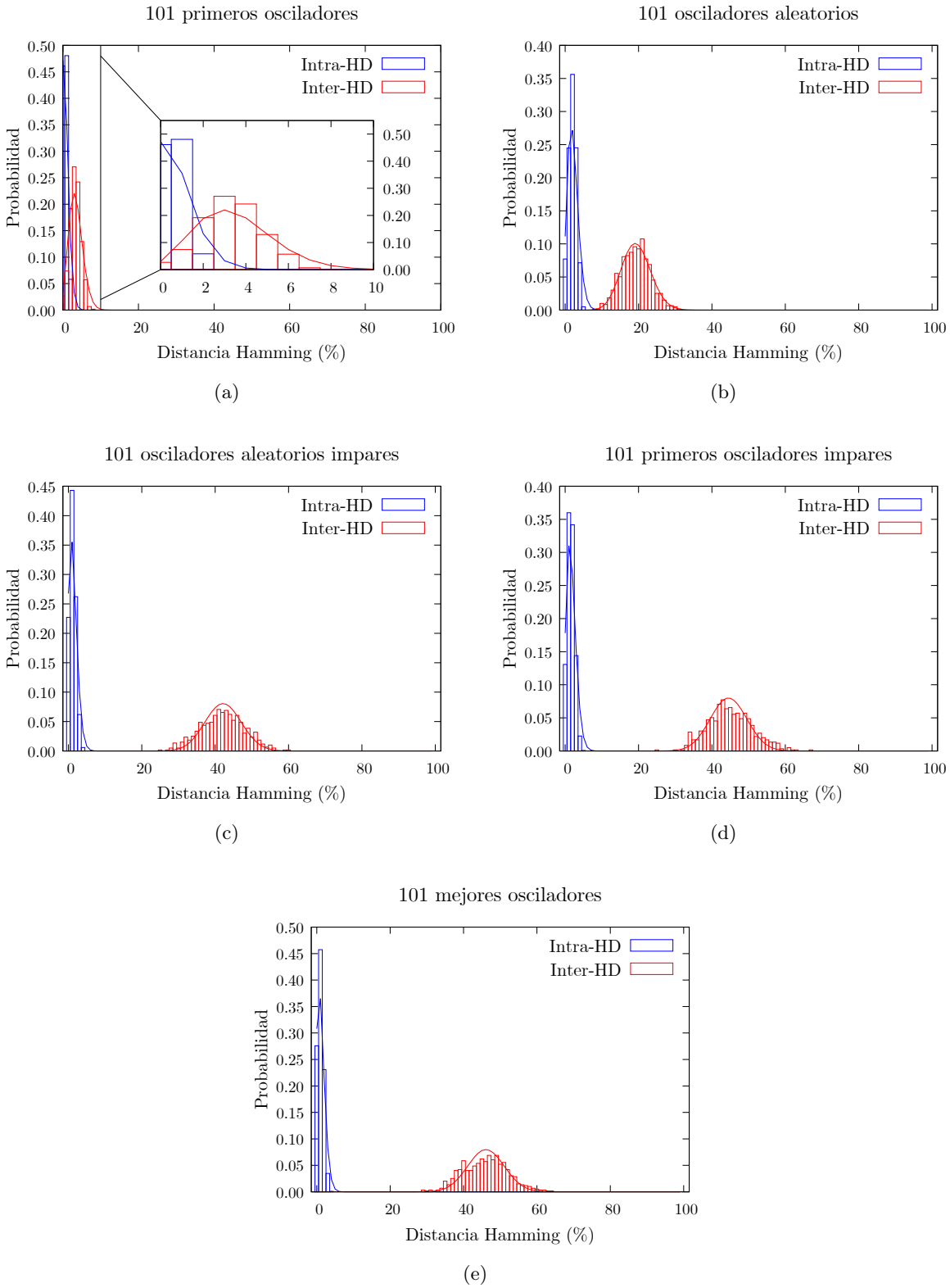



**Figura 5.7:** Histograma de la Inter-$HD$ e Intra-$HD$ en función de la estrategia estudiada para la FPGA-B5 y sus ajustes a una distribución binomial. a) *101 primeros.* b) *101 aleatorios.* c) *101 impares aleatorios.* d) *101 primeros impares.* e) *101 mejores.*

Para cuantificar el grado de identificabilidad, se han calculado las curvas $FAR$ y $FRR$ en cada una de las cinco estrategias (Anexo B) a partir de los ajustes a las binomiales de la interdistancia e intradistancia. Además, se ha obtenido el $EER$ y el $t_{EER}$ así como las tasas $FAR$ y $FRR$ para dicho umbral (Tabla 5.5).

| Estrategia | $t_{EER}$ | $FRR\ (t_{EER})$ | $FAR\ (t_{EER})$ | $EER$ |
|---|---|---|---|---|
| *101 primeros* | 2 | $3{,}971 \cdot 10^{-2}$ | $3{,}25 \cdot 10^{-1}$ | $3{,}25 \cdot 10^{-1}$ |
| *101 aleatorios* | 8 | $3{,}45 \cdot 10^{-4}$ | $1{,}40 \cdot 10^{-3}$ | $1{,}40 \cdot 10^{-3}$ |
| *101 primeros impares* | 15 | $1{,}86 \cdot 10^{-11}$ | $2{,}93 \cdot 10^{-10}$ | $2{,}93 \cdot 10^{-10}$ |
| *101 impares aleatorios* | 13 | $6{,}70 \cdot 10^{-11}$ | $2{,}32 \cdot 10^{-10}$ | $2{,}32 \cdot 10^{-10}$ |
| *101 mejores* | 14 | $1{,}05 \cdot 10^{-12}$ | $8{,}42 \cdot 10^{-12}$ | $8{,}42 \cdot 10^{-12}$ |

**Tabla 5.5:** Valores $t_{EER}$, $FAR\ (t_{EER})$, $FRR\ (t_{EER})$ y $EER$ para cada estrategia.

En este trabajo se ha utilizado como figura de mérito el $EER$. Éste es el indicador que se utiliza habitualmente para determinar la identificabilidad. En términos prácticos, éste es el parámetro más importante de todos, ya que la aplicación final de una PUF es la identificación y la autenticación. Cuanto menor sea el $EER$, mejor será la PUF en términos de identificabilidad.

El mayor $EER$ se obtiene para la estrategia *101 primeros*, por tanto se trata de la peor de las cinco estrategias desde el punto de vista de la identificabilidad. Esto ya se había sospechado en la sección anterior. En el caso *101 aleatorios*, el $EER$ mejora tres órdenes de magnitud respecto a la selección de los primeros osciladores. Sin embargo, para la estrategia *101 primeros impares* se produce una mejora considerable del $EER$, ya que aumenta nueve órdenes de magnitud respecto al caso *101 primeros* y siete órdenes de magnitud respecto a la estrategia *101 aleatorios*.

Respecto a la estrategia *101 impares aleatorios*, se ha obtenido un menor umbral ($t_{EER} = 13$) que en el caso *101 mejores* ($t_{EER} = 14$). Esto indica que la probabilidad de producirse un falso rechazo o una falsa aceptación se minimiza con un menor número de bits no coincidentes para el caso *101 impares aleatorios*. A pesar de esto, para la estrategia *101 mejores* (Figura 5.8) se obtiene un menor $EER$, con un valor del orden de $10^{-12}$, es decir, dos órdenes de magnitud de diferencia respecto al caso *101 impares aleatorios*. Se concluye que *101 mejores* es la mejor estrategia para seleccionar los osciladores, tal y como se esperaba.

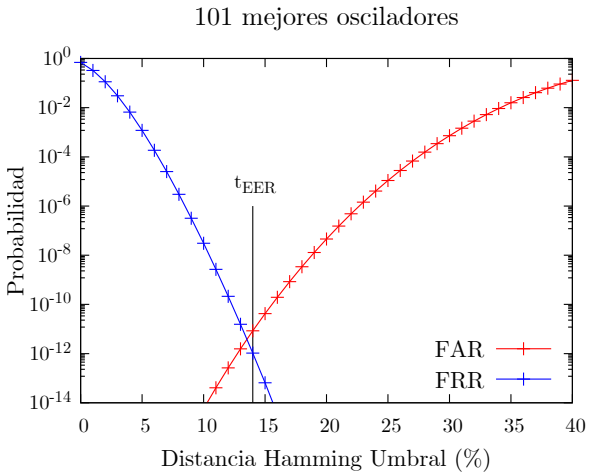



**Figura 5.8:** Curvas $FAR$ y $FRR$ para la estrategia *101 mejores*.

# 6. Modificación de los resultados frente a cambios ambientales

Modificaciones en el entorno de operación pueden causar que la misma instancia de PUF produzca respuestas distintas ante un mismo desafío. Algunas de estas variaciones están relacionadas con cambios en la temperatura o en el voltaje de alimentación de la FPGA. A continuación, se estudia el efecto de cambiar estos dos parámetros en las respuestas de las RO-PUF implementadas en la FPGA y se analiza la reproducibilidad de la PUF.

## 6.1. Cambios de temperatura

Para estudiar cómo cambia la reproducibilidad de la PUF frente a cambios de temperatura, se ha utilizado una cámara térmica Aralab FitoTerm 22E que dispone de un sensor de temperatura Pt 100 Jumo Techno 80 pc1.2005.1, permite variar la temperatura de -40 °C a +160 °C, ±0,5 °C y se alimenta eléctricamente con 230 V, 50 Hz, 16 A. Se ha estudiado la variación de la intradistancia en la FPGA-B9. Para ello se han seleccionado distintas temperaturas en la cámara térmica $T_{\text{chamber}}$ desde -20 °C hasta 60 °C en intervalos de 10 °C.

Además, la FPGA dispone de un Convertidor Analógico-Digital Xilinx (XADC) *on-chip*. Dicho XADC contiene un sensor de temperatura que se encarga de producir una salida de voltaje que es proporcional a la temperatura de la pastilla [12]. Este voltaje de salida es digitalizado por el ADC que produce un código de salida de 12 bits, de tal forma que se obtiene la temperatura de la FPGA ($T_{\text{FPGA}}$) a partir de la función de transferencia de salida y se almacena en un registro. Típicamente este sensor se utiliza para evitar que la FPGA alcance temperaturas críticas que impidan su correcto funcionamiento. Además de un sensor de temperatura, el XADC también permite visualizar otros parámetros como el voltaje de alimentación del núlceo o el voltaje de alimentación de los bloques de memoria RAM.

### 6.1.1. Frecuencias de los osciladores

En primer lugar, se ha estudiado la influencia de la temperatura en la frecuencia de los osciladores. Para ello se han seleccionado cuatro RO de índices pares e impares situados en varias zonas separadas de la FPGA. Los resultados obtenidos se muestran en el Anexo C. En todas las curvas se observa una tendencia clara: la frecuencia de los osciladores tiende a disminuir a medida que aumenta la temperatura. Algunos artículos como el de Shu-Min et al. [13] ya muestran que la frecuencia de un oscilador en anillo se reduce a medida que aumenta de temperatura y proponen este fenómeno como mecanismo de medición de la temperatura ambiente.

### 6.1.2. Reproducibilidad

Una vez estabilizada tanto la temperatura ambiente medida con el sensor de la cámara ($T_c$) como la temperatura interna de la FPGA medida con el XADC ($T_{\text{FPGA}}$), se han tomado 100 medidas en cada temperatura para cada uno de los 400 osciladores para la FPGA-B5. En la Tabla 6.1 se muestran ambas temperaturas. Seguidamente, se ha calculado la Intra-$HD$. Una vez más, se han tomado medidas de los 400 osciladores y no únicamente de los 101 necesarios para obtener la palabra de salida con el fin de comparar estrategias entre sí con frecuencias de los osciladores que correspondan a una misma medida.

Ya determinada con cuál de las cinco estrategias se obtiene un mejor resultado, en la Figura 6.1 se muestra la variación de la Intra-$HD$ promedio en función de la temperatura ambiente ($T_c$) seleccionada en la cámara térmica para el caso *101 mejores*. En la Tabla 6.1 se observa que la FPGA se encuentra sistemáticamente unos 15 °C por encima de la temperatura del entorno. Utilizando la estrategia *101 mejores* todas las intradistancias promedio se encuentran por debajo del 2 %, incluso considerando el error de la media. Por tanto, se puede considerar que para esta estrategia la RO-PUF implementada es estable frente a variaciones en el rango de temperaturas estudiado. La Intra-$HD$ máxima se obtiene cuando la temperatura de la cámara es de 20 °C y la mínima para 40 °C. Además, no se observa una tendencia clara de la Intra-$HD$ media con la temperatura: a temperaturas inferiores a 0 °C la intradistancia aumenta con la temperatura, pero luego disminuye exhibiendo un pico en 20 °C y vuelve a aumentar para 60 °C.

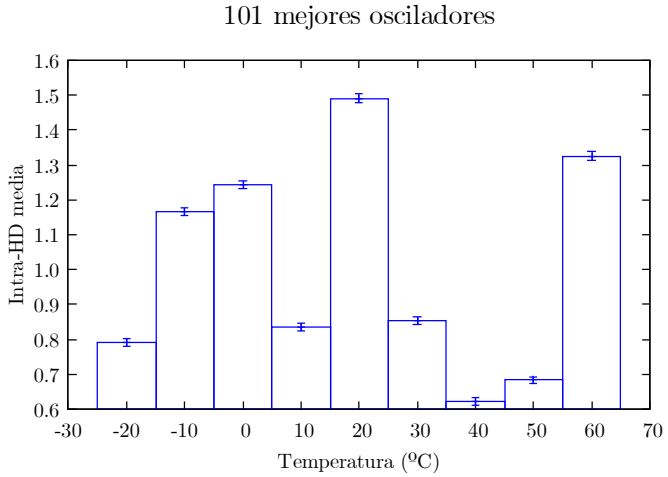



**Figura 6.1:** Evolución de la Intra-$HD$ promedio en función de la temperatura (*101 mejores*).

| $T_c$ (°C) | $T_{FPGA}$ (°C) |
|---|---|
| -20 | -5,4 |
| -10 | 5,9 |
| 0 | 14,5 |
| 10 | 24,6 |
| 20 | 34,8 |
| 30 | 44,8 |
| 40 | 55,0 |
| 50 | 65,3 |
| 60 | 75,3 |

**Tabla 6.1:** Temperaturas medidas con la cámara térmica y con el XADC.

Este estudio también se ha realizado para el resto de estrategias. Al igual que con la estrategia *101 mejores*, no se observa una tendencia clara. Las gráficas obtenidas se muestran en el Anexo D. En los casos *101 primeros* y *101 aleatorios* también se obtiene que para todas las temperaturas la Intra-$HD$ es inferior al 2 %, presentando así una buena reproducibilidad estable frente a cambios de temperatura. Sin embargo, a pesar se su buena reproducibilidad, estos casos presentaban una unicidad muy pobre en comparación con el resto de estrategias. Por último, para los otros dos casos (*101 impares aleatorios* y *101 primeros impares*) se obtiene que para algunas temperaturas la Intra-$HD$ promedio supera el 2 %. Esto permite concluir que, de entre las tres estrategias que presentaban mejor unicidad, seleccionando los *101 mejores* osciladores se obtiene una mejor reproducibilidad en el intervalo de temperaturas estudiado.

## 6.2. Cambios en el voltaje de alimentación

El objetivo de esta sección es estudiar el efecto de los cambios en el voltaje de alimentación de la FPGA ($V_{CCINT}$) en la reproducibilidad de la PUF. Esta señal es generada por la unidad de gestión de energía (PMU) integrada TPS65400 dentro de la placa y tiene un valor nominal de 1,0 V. Además de tener un sensor de temperatura, el XADC tiene un sensor de voltaje a través del cual se ha observado que el valor real de la tensión de alimentación es $V_{CCINT}$ = 1,022 V. Una forma de modificar este voltaje de alimentación es a través del puerto I2C.

El puerto I2C del TPS va al pin JP2 de la FPGA y por tanto, puede ser *hackeado*. El bus I2C utiliza dos líneas para transmitir la información: SDA (para los datos) y SCL (para el reloj). Además, tiene una línea de tierra GND. Para cambiar el valor de la señal se ha utilizado una placa Arduino UNO y se han conectado SDA, SCL y GND. Mediante una máquina de estados es posible cambiar el voltaje de referencia ($V_{\text{REF}}$) en intervalos de 10 mV y esto hará que $V_{\text{CCINT}}$ varíe 12,5 mV. El voltaje $V_{\text{REF}}$ tiene un valor nominal de 1,25 V y puede tomar un valor máximo de 2,0 V. Para evitar dañar la FPGA, solo se ha modificado hasta que $V_{\text{CCINT}}$ varíe un máximo de un 10 %, evitando llegar hasta 1,1 V.

### 6.2.1. Frecuencias de los osciladores

En primer lugar, se ha representando la frecuencia de varios osciladores situados en distintas zonas de la FPGA. Los resultados se muestran en el Anexo C. En este caso se observa una tendencia clara: a medida que aumenta la tensión de alimentación de la placa, la frecuencia de los osciladores también tiende a aumentar, llegando a variar más de 60 MHz en 100 mV. En otros artículos [13] ya se explora esta dependencia, obteniendo unos resultados similares.

### 6.2.2. Reproducibilidad

A continuación se ha estudiado la Intra-$HD$ en las cinco estrategias propuestas para nueve valores distintos de $V_{\text{CCINT}}$. Se han tomado 100 medidas en cada voltaje para cada uno de los 400 osciladores de la FPGA-A1 y se ha calculado la Intra-$HD$ promedio. Los resultados obtenidos para el caso *101 mejores* se muestran en la Figura 6.2.

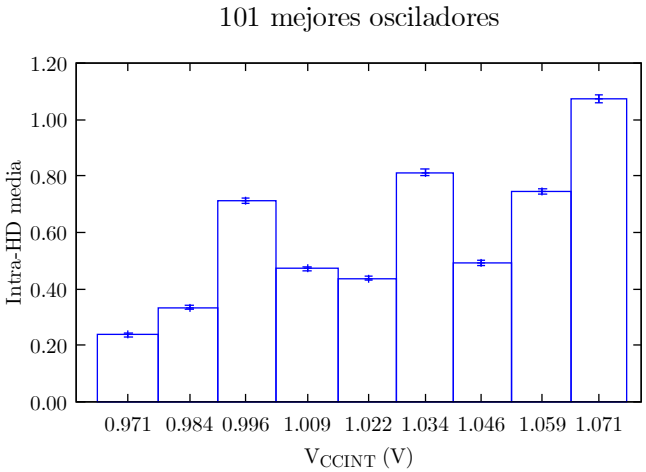



**Figura 6.2:** Evolución de la Intra-$HD$ media en función de la $V_{\text{CCINT}}$ (*101 mejores*)

Utilizando la estrategia *101 mejores*, todas las Intra-$HD$ promedio se encuentran por debajo del 1,1 %, incluso si se considera el error de la media. Por tanto, para esta estrategia la RO-PUF implementada resulta estable frente a variaciones del voltaje de alimentación en el rango estudiado. La Intra-$HD$ máxima se obtiene para $V_{\text{CCINT}} = 1{,}071$ V y la mínima para $V_{\text{CCINT}} = 0{,}971$ V. Además, no se observa una tendencia clara de la Intra-$HD$ media a medida que aumenta el voltaje: podría parecer que la Intra-$HD$ media aumenta con $V_{\text{CCINT}}$, pero hay algunos valores que claramente no siguen esta tendencia (0,996 V, 1,034 V ...). Este fenómeno también se observaba en el estudio previo de la reproducibilidad frente a cambios de temperatura.

Estos resultados ponen de manifiesto las bondades de la técnica de medida compensada (comparación de las frecuencias de los osciladores por parejas) que nace precisamente para que la respuesta de la PUF sea robusta frente a cambios ambientales. En este trabajo se ha observado que incluso variando las frecuencias con la temperatura o con el voltaje de alimentación de la FPGA, la respuesta se mantiene aproximadamente estable.

Con el resto de estrategias tampoco se observa una tendencia clara. En el caso *101 primeros*, la Intra-$HD$ prácticamente no varía para cada uno de los voltajes seleccionados, excepto para un caso en el que disminuye prácticamente a cero y otro en el que aumenta. En las estrategias *101 aleatorios* y *101 impares aleatorios* son en las que se observan mayores diferencias en las Intra-$HD$ para cada uno de los voltajes. Por último, en el caso *101 primeros impares* se observa una tendencia similar al caso *101 mejores*. Sin embargo, hay que destacar que en todos los casos la Intra-$HD$ se encuentra por debajo del 1,2 % y por tanto presentan también una reproducibilidad estable frente a cambios de $V_{CCINT}$. Las gráficas obtenidas se muestran en el Anexo E.

# 7. Conclusiones y futuras líneas de investigación

En este trabajo se ha estudiado el concepto de PUF, sus principales ventajas e inconvenientes así como algunas métricas para determinar la calidad de la PUF. Además, se han analizado diversas formas de obtener la frecuencia de varios osciladores en anillo implementados en una FPGA y se han comparado diversas estrategias para seleccionar la posición de los osciladores para construir una RO-PUF. Por último, se ha implementado una PUF basada en la mejor de las estrategias y se ha analizado la calidad de la PUF así como su estabilidad frente a cambios en la temperatura y en el voltaje de alimentación.

El análisis principal realizado en este trabajo pone de manifiesto que la calidad de la PUF depende en gran medida de la posición en la que se implementan los osciladores en la FPGA. De este estudio se concluye la importancia de evitar comparaciones entre osciladores implementados en *slices* de tipo L y *slices* de tipo M, ya que este tipo de comparaciones disminuyen la aleatoriedad de la respuesta de la PUF. De las cinco estrategias observadas, hay dos que destacan debido a su alta unicidad, reproducibilidad e identificabilidad: *101 primeros impares*, donde se han seleccionado osciladores próximos implementados en un mismo tipo de *slice*; y *101 mejores*, donde se han seleccionado aquellos osciladores con frecuencias más cercanas entre sí. Con el segundo sistema, se han obtenido unos resultados ligeramente mejores en términos de unicidad, reproducibilidad e identificabilidad en comparación con el primer sistema de selección de osciladores. Sin embargo, la estrategia *101 mejores* requiere de un análisis previo de las frecuencias de los osciladores lo que resultaría un inconveniente en algunas aplicaciones, aunque podría utilizarse la fase de "enrollment" para hacer este estudio de las frecuencias. Por último, se ha observado que al aumentar la temperatura, la frecuencia de los osciladores disminuye ligeramente y al aumentar el voltaje de alimentación de la FPGA, la frecuencia aumenta considerablemente. Además, la respuesta de la PUF se mantiene aproximadamente estable frente a ambos tipos de variaciones. Ésta es una de las bondades de la técnica de medida compensada.

Futuras líneas de investigación pasarían por aumentar el número de osciladores de la FPGA, implementar otros tipos de arquitectura PUF o bien analizar si comparando las frecuencias de los osciladores de otra forma también se consigue minimizar el efecto de la posición de los osciladores en la FPGA. También se podría extender este análisis a otros dispositivos y fabricantes

o bien estudiar el consumo de cada uno de los osciladores individualmente con el fin de obtener estructuras de RO-PUF que además minimicen dicho consumo.

A partir de este trabajo, se ha presentado un artículo en el 17º Congreso Internacional de Investigación de Doctorado en Microelectrónica y Electrónica (PRIME 2022) [2].

# Bibliografía


[1] Scott A. Vanstone, Paul C. van Oorschot, and Alfred J. Menezes. *Handbook of Applied Cryptography*. CRC Press, Dec 7, 2018.

[2] M. Garcia-Bosque, R. Aparicio, G. Díez-Señorans, C. Sánchez-Azqueta, and S. Celma. An analysis of the behaviour of a puf based on ring oscillators depending on their locations. In *Proceedings of the 2022 17th Conference on Ph.D. Research in Microelectronics and Electronics (PRIME)*, 2022.

[3] Maximilian Hofer and Christoph Böhm. *Physical Unclonable Functions in Theory and Practice*. Springer, New York, NY, 1. aufl. edition, 2013.

[4] Roel Maes. *Physically Unclonable Functions : Constructions, Properties and Applications*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013 edition, 2013.

[5] G. Suh and Srinivas Devadas. Physical unclonable functions for device authentication and secret key generation. In *2007 44th ACM/IEEE Design Automation Conference*, DAC '07, pages 9–14. ACM, Jun 4, 2007.

[6] Jiliang Zhang. 1-out-of-n ring oscillator puf architecture, 2014. [Online; accedido el 16 de abril, 2022]. Disponible: https://www.researchgate.net/figure/1-out-of-N-ring-oscillator-PUF-architecture-11_fig6_271997658.

[7] G. Díez-Señorans, M. García-Bosque, C. Sánchez-Azqueta, and S. Celma. Extracción de entropía en pufs de medida compensada. *Jornada de Jóvenes Investigadores del I3A*, 2021.

[8] Tul. *PYNQ-Z2 Reference Manual v1.1*. [Online; accedido el 15 de abril, 2022]. Disponible: https://dpoauwgwqsy2x.cloudfront.net/Download/PYNQ_Z2_User_Manual_v1.1.pdf.

[9] Zulfikar Zulfikar, Norhayati Soin, Sharifah Fatmadiana Wan Muhamad Hatta, and Mohamad Sofian Abu Talip. Runtime analysis of area-efficient uniform ro-puf for uniqueness and reliability balancing. *Electronics (Basel)*, 10(20):2504, Oct 14, 2021.

[10] A. Maiti and P. Schaumont. Improving the quality of a physical unclonable function using configurable ring oscillators. In *2009 International Conference on Field Programmable Logic and Applications*, pages 703–707. IEEE, Aug 2009.

[11] N. Smirnov. Table for estimating the goodness of fit of empirical distributions. *The Annals of mathematical statistics*, 19(2):279–281, Jun 1, 1948.

[12] Xilinx. *7 Series FPGAs and Zynq-7000 SoC XADC Dual 12-Bit 1 MSPS Analog-to-Digital Converter. User Guide v1.10.1.* [Online; accedido el 15 de abril, 2022]. Disponible: https://www.xilinx.com/content/dam/xilinx/support/documents/user_guides/ug480_7Series_XADC.pdf.

[13] Katherine Shu-Min LI, Yingchieh HO, Yu-Wei YANG, and Liang-Bi CHEN. An oscillation-based on-chip temperature-aware dynamic voltage and frequency scaling scheme in system-on-a-chip. *IEICE transactions on information and systems*, E97.D(9):2320–2329, 2014.

# Anexos

## A. Ajuste de la Inter-*HD* a una distribución binomial

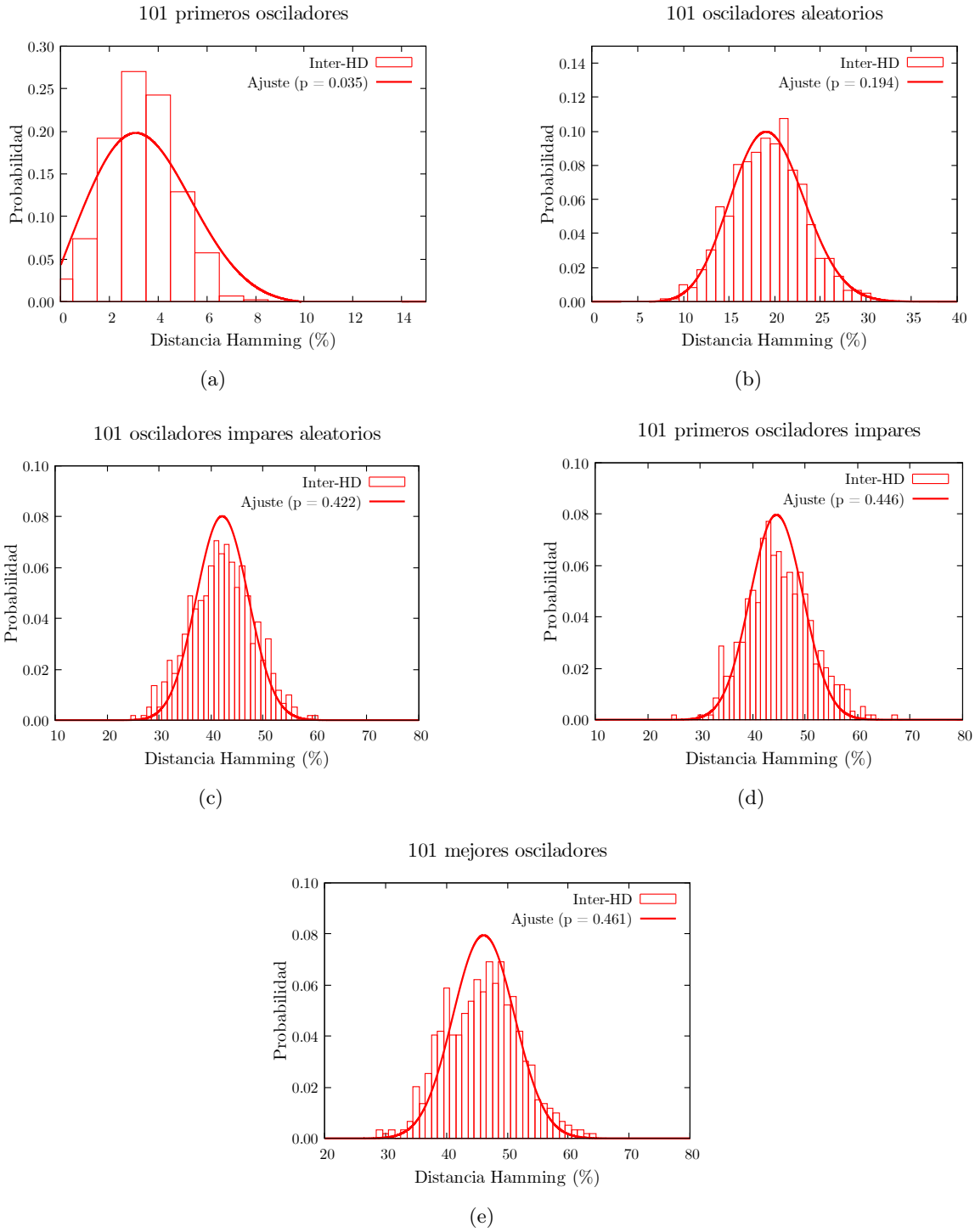



**Figura A.1:** Ajuste de las distribuciones experimentales de la Inter-*HD* a una binomial. a) *101 primeros.* b) *101 aleatorios.* c) *101 aleatorios impares.* d) *101 primeros impares.* e) *101 mejores.*

## B. Curvas *FAR* y *FRR* para cada una de las cinco estrategias

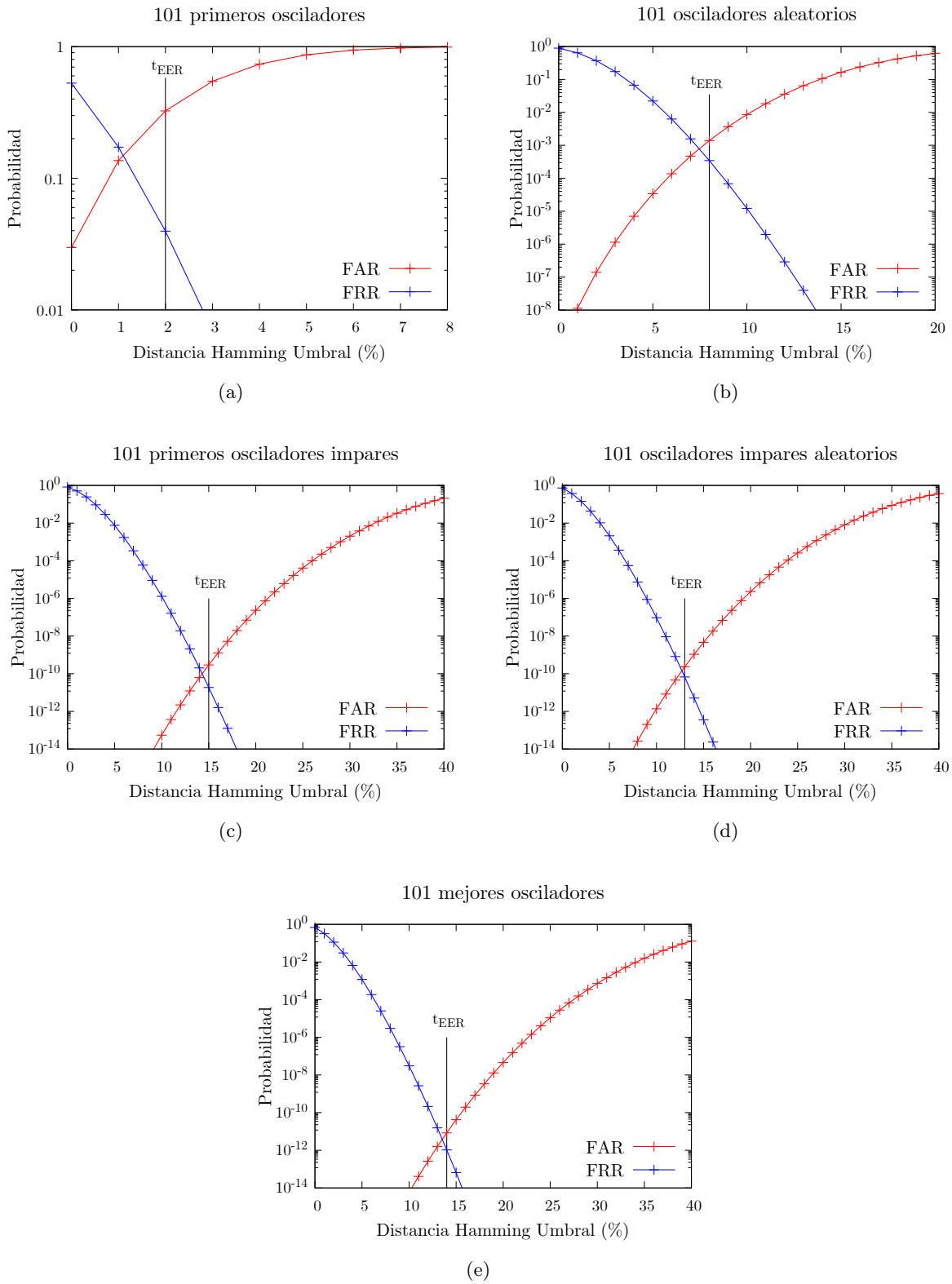



**Figura B.1:** Curvas $FAR$ y $FRR$ en función del umbral elegido para cada una de las estrategias sometidas a estudio. a) *101 primeros.* b) *101 aleatorios.* c) *101 impares aleatorios.* d) *101 primeros impares.* e) *101 mejores.*

## C. Efecto de los cambios de temperatura y del voltaje de alimentación en la frecuencia de los osciladores

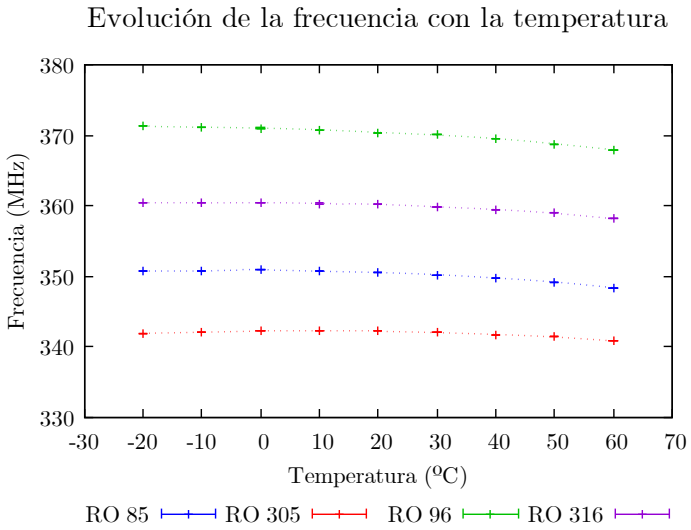



**Figura C.1:** Evolución de la frecuencia de cuatro osciladores con la temperatura de la cámara.

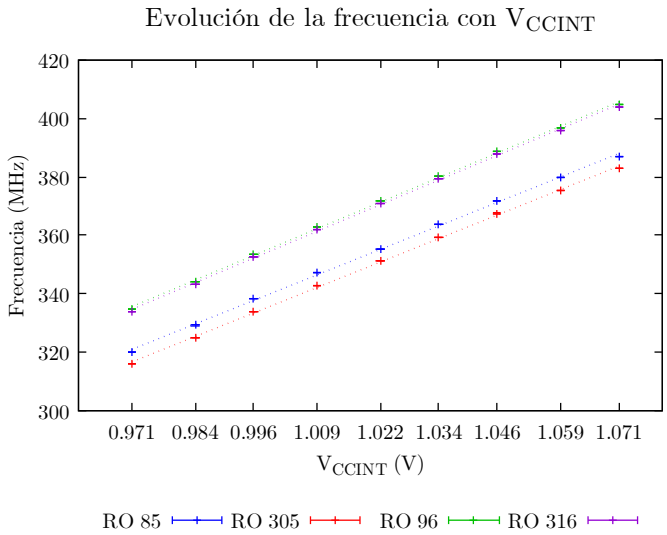



**Figura C.2:** Evolución de la frecuencia de cuatro osciladores con $V_{\mathrm{CCINT}}$.

## D. Evolución de la Intra-*HD* promedio con la temperatura

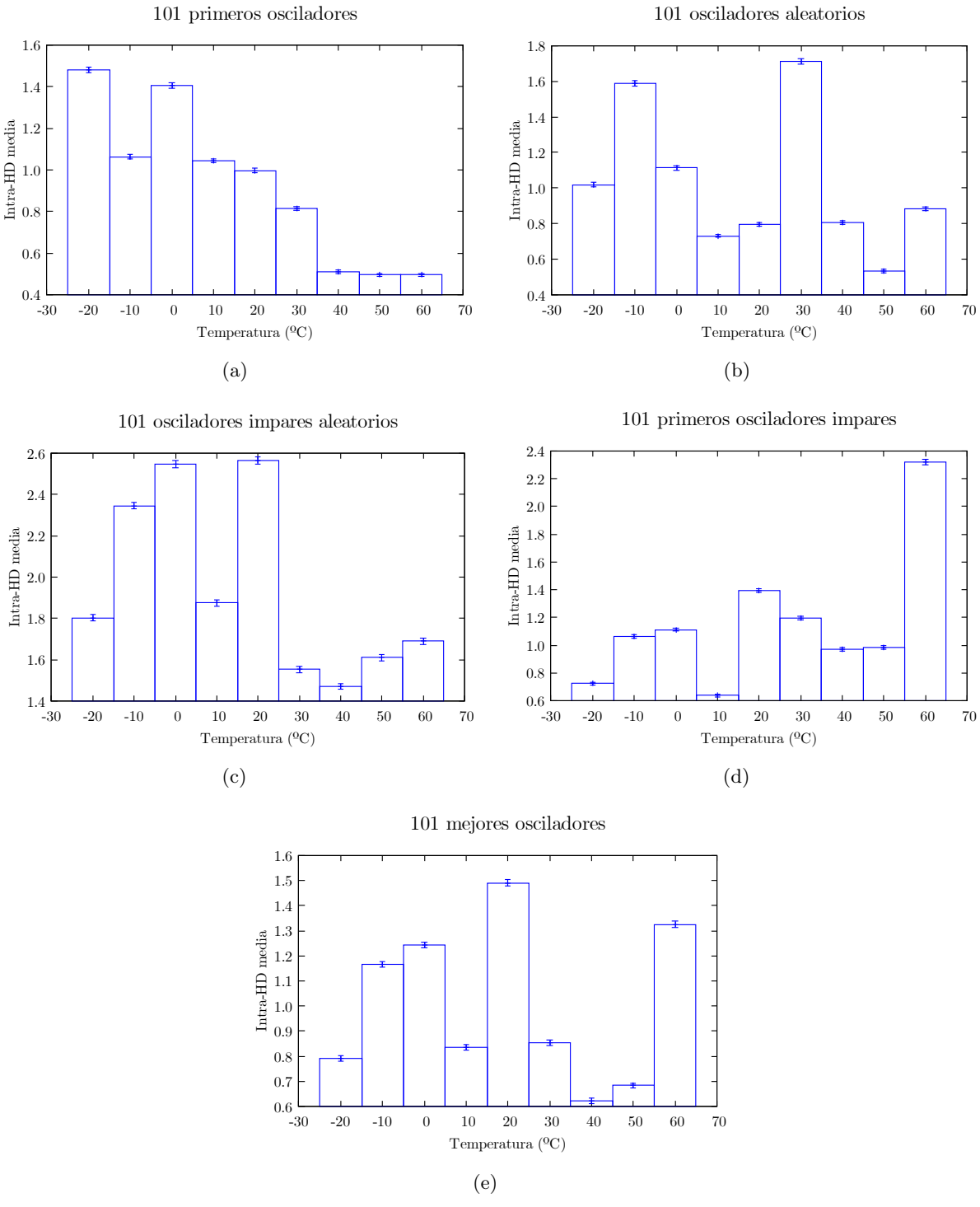



**Figura D.1:** Evolución de la Intra-*HD* promedio con la temperatura de la cámara térmica. a) *101 primeros*. b) *101 aleatorios*. c) *101 aleatorios impares*. d) *101 primeros impares*. e) *101 mejores*.

## E. Evolución de la Intra-*HD* promedio con el voltaje de alimentación

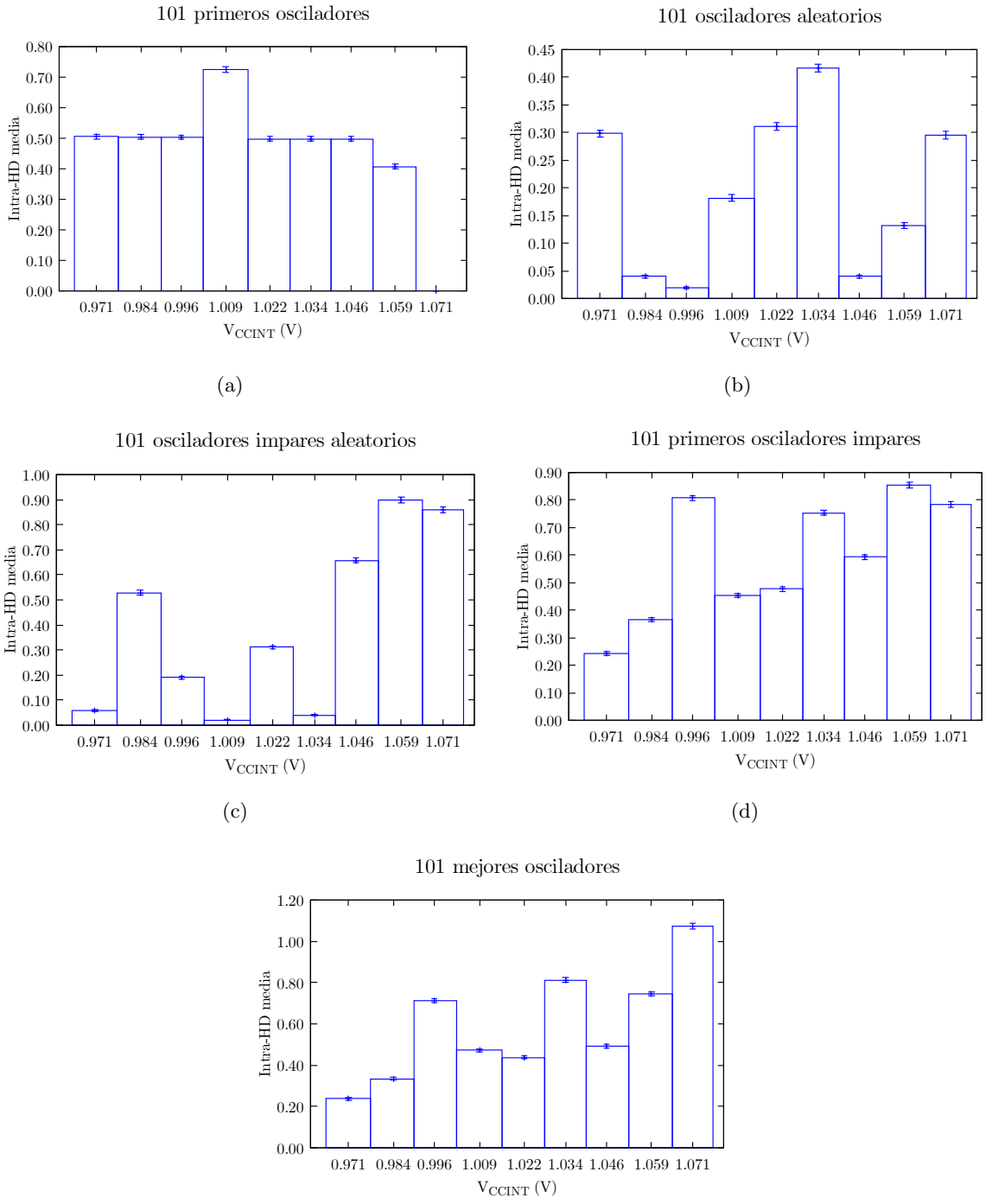



**Figura E.1:** Evolución de la Intra-*HD* promedio con el voltaje de alimentación. a) *101 primeros.* b) *101 aleatorios.* c) *101 aleatorios impares.* d) *101 primeros impares.* e) *101 mejores.*