Arquitectura PUF en FPGA para biometría cancelable

por

Lic. Luis Enrique Namigtle Jiménez

Tesis sometida como requerimiento parcial para obtener el grado de:

Maestría en Ciencias en la especialidad de Electrónica

En el:

Instituto Nacional de Astrofísica, Óptica y Electrónica

Marzo, 2023

Tonantzintla, Puebla

Asesor:

Dr. Juan Manuel Ramírez Cortés

Co-Asesor:

Dr. José de Jesús Rangel Magdaleno

Arquitectura PUF en FPGA para biometría cancelable

Tesis de Maestría

Por:

Luis Enrique Namigtle Jiménez

Asesor:
Dr. Juan Manuel Ramírez Cortés
Co-Asesor:
Dr. José de Jesús Rangel Magdaleno

Instituto Nacional de Astrofísica Óptica y Electrónica
Coordinación de Electrónica

Tonantzintla, Puebla. Marzo 2023

Agradecimientos

Al Instituto Nacional de Astrofísica Óptica y Electrónica Coordinación de Electrónica y al CONACYT por darme la oportunidad de realizar un posgrado y contribuir con investigaciones científicas para el desarrollo de la ciencia y la tecnología, fortaleciendo mis capacidades, habilidades y compromisos en el área de mi interés.

Al Dr. Juan Manuel Ramírez Cortes y al Dr. José de Jesús Rangel Magdaleno por permitirme formar parte de su grupo.

Al compañero Juan Carlos por resolver mis dudas, orientarme y hacer observaciones para realizar un buen trabajo de tesis.

Al compañero Sergio por la paciencia y tiempo en explicarme temas complicados.

A mi compañero Ciro, que me ayudo en los tiempos difíciles de trabajos y tareas.

A mi Madre Zoila por tenerme paciencia, estar conmigo en todo momento y ese amor leal, además de los grandes consejos que me comparte para ser una gran persona. A mi difunto Padre Artemio † que en paz descanse. Desafortunadamente, no pudo terminar conmigo esta gran aventura, pero sé que donde quiera que este, está muy orgulloso. A mis hermanos que con este confinamiento pude aprender más de ellos y describirlos con una sola palabra; Alfredo es una persona sabia, Jesús una persona audaz y Miriam una persona tenaz. A mi tío Federico por su apoyo incondicional. Sin demeritar a mis cuñadas Luz y Mariela, que también fueron un gran apoyo durante esta etapa. Y a mis sobrinos Matías, Melisa, Julieta y Lucia, que

sin duda, son seres de luz admirables por su gran inteligencia.

Dedicatoria

A la pandemia COVID, por enseñarme el valor de la VIDA.

A mi familia, por acompañarme y aconsejarme cuando más lo necesitaba.

A mi hermano Alfredo, que no deja de sorprenderme cada día y agradecido con él por todo los conocimientos que me comparte, desde lo académico hasta lo personal.

Y a todos aquellas personas que conocí durante esta gran etapa.

Resumen

En este trabajo se propone la implementación de las funciones físicas no clonables con enrutado manual y automático para determinar cuál es la apropiada para aplicaciones criptográficas.
Dentro de las funciones físicas no clonables orientadas en silicio, se pueden encontrar arquitecturas basadas en memorias, oscilaciones y retardos. Este trabajo se centra en implementar las arquitecturas basadas en retardos como son el Arbiter PUF, Arbiter PUF XOR, Arbiter PUF Feed Forward, Arbiter PUF Lightweight Secure, Arbiter PUF XOR Lightweight Secure y Arbiter PUF Lightweight Secure Feed Forward.
Estas arquitecturas fueron implementadas en dos tarjetas FPGA de la familia Spartan 6 (Amiba 2 [1] y Atlys), con la finalidad de observar el comportamiento ante ciertas pruebas de medición de calidad.
Estas claves aleatorias fueron aplicadas al rasgo biométrico ECG para hacer biometria cancelable. Los datos provenientes de la señal ECG ya fueron procesados anteriormente, es decir, tiene su previa etapa de filtración para hacer uso de ellas.
Posteriormente se propuso una nueva técnica de cancelación que mejora la seguridad de los individuos y por lo tanto evita que la plantilla original tenga correlación con las plantillas cancelables.
Se concluyó que la mejor arquitectura PUF para la generación de números estocásticos es el Arbiter PUF con diseño automático, logrado a partir del uso de las medidas de calidad PUF y las métricas NIST. Por otro lado, la técnica propuesta para la cancelación de datos nos muestra resultados mejores que la tesis referenciada en este trabajo empleando las medidas de evaluación para técnicas de protección de plantillas biométricas.

Summary

This paper proposes the implementation of physically unclonable functions with manual and automatic routing to determine which one is suitable for cryptographic applications. Among the physically unclonable functions oriented in silicon, architectures based on memories, oscillations, and delays can be found. This document focuses on implementing delay-based architectures such as Arbiter PUF, Arbiter PUF XOR, Arbiter PUF Feed Forward, Arbiter PUF Lightweight Secure, Arbiter PUF XOR Lightweight Secure, and Arbiter PUF Lightweight Secure Feed Forward. These architectures were implemented on two FPGA boards of the Spartan 6 family (Amiba 2 [1] and Atlys) to observe their behavior under certain quality measurement tests. These random keys were applied to the ECG biometric trait to make it cancellable. The data coming from the ECG signal has been previously processed, i.e., it has its previous filtering stage to make use of it. Subsequently, a new cancellation technique was proposed, which improves individuals security and, therefore, prevents the original template from being correlated with cancellable templates. It was concluded that the best PUF architecture for generating stochastic numbers is the Arbiter PUF with automatic design, achieved from the use of PUF quality measures and NIST metrics. On the other hand, the proposed data cancellation technique shows better results than the thesis referenced in this document using evaluation measures for biometric template protection techniques.

Lista de Siglas

AES	*Estándar de Cifrado Avanzado*
APUF	*Arbiter PUF*
APUF FF	*Arbiter PUF Feed Forward*
APUF LS	*Arbiter PUF Lightweight Secure*
APUF LS FF	*Arbiter PUF Lightweight Secure Feed Forward*
APUF XOR	*Arbiter PUF XOR*
APUF XOR LS	*Arbiter PUF XOR Lightweight Secure*
ARM	*Ataques a través de la Multiplicidad de Registros*
AROC	*Área bajo la curva de Característica operativas del receptor*
BPT	*Medidas de evaluación para técnicas de protección de plantillas biométricas*
CI	*Circuito Integrado*
CLB	*Configurable Logic Block*
CRP	*Challenge-Response Pair*
DCT	*Transformada Discreta Wavelete*
DFT	*Transformada Discreta de Fourier*
DWT	*Alineamiento temporal dinámico*
DWT	*transformada wavelet discreta*
ECG	*Electrocardiograma*
EER	*Tasa de Error Igual*
EFV	*Hashing del Vector de características Extendido*
FAR	*Tasa de Falsos Aceptados*
FMR	*Tasa de Falsas Coincidencias*
FrFT	*Transformada Fraccional de Fourier*
FRR	*Tasa de Falsos Rechazos*

FPGA	*Field Programmable Gate Array*
FSM	*Máquina de estados Finita*
GAR	*Tasa de Aceptación Genuina*
LFSR	*Registro de desplazamiento de retroalimentación lineal*
LUT	*Look Up Table*
MAC	*Código de Autenticación de Mensajes*
PUF	*Función Fisicamente Inclonable*
ROC	*Característica Operativa del Receptor*

Tabla de Contenido

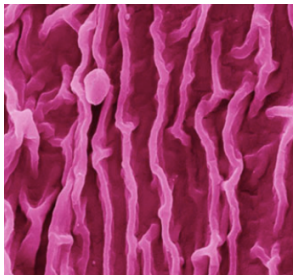
Lista de Figuras

Lista de Tablas

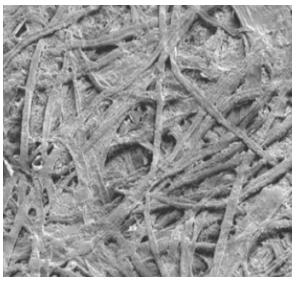
Capítulo 1

Introducción

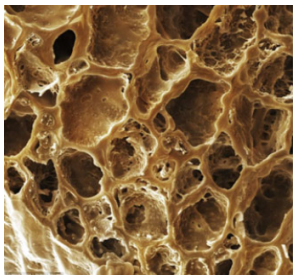
Las anomalías físicas hacen hincapié en las imperfecciones intrínsecas aleatorias de objetos físicos que se encuentran en nuestro entorno y que suelen visualizarse a un nivel microscópico. En la figura 1.1 se muestra ejemplos de la estructura de una rosa, una hoja de papel, una semilla de café y un diente. Se puede notar cómo las estructuras morfológicas tienen distintos patrones de diseño entre ellas. Si comparamos 100 rosas, a simple vista se puede decir que todas son iguales, pero visualizando a nivel microscópico estas tienen irregularidades únicas que las diferencian de las demás.



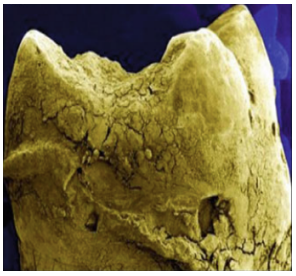
(a) Una rosa



(b) Una hoja de papel



(c) Una semilla de café



(d) Un diente

Figura 1.1: Ejemplo de irregularidades físicas a nivel microscópico [2].

Todas estas alteraciones son encontradas en el diseño de circuitos integrados (CI)

debido a la difícil fabricación de componentes de tamaño preciso. Supongamos un chip semiconductor con una tecnología de 90nm como en la figura 1.2 donde se observa irregularidades en los conductores metálicos [2].

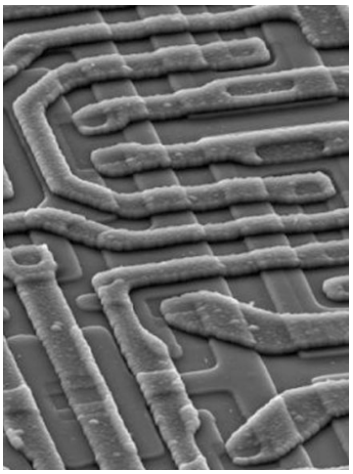


Figura 1.2: Irregularidades en un Circuito Integrado (CI) [2].

Estas imperfecciones internas son usadas para crear identidades únicas también conocidas como como funciones físicas no clonables (PUFs).

1.1. Justificación

Las claves de seguridad regularmente son almacenadas en discos o memorias, donde solo personal autorizado puede acceder a estos datos. Esto es un problema de vulnerabilidad debido a que fácilmente se pueden robar estos dispositivos y clonar los datos.

Una de las posibles alternativas para resolver este caso, es con la generación de números aleatorios a través de diversos programas, el problema es que los atacantes emplean ingeniería inversa para poder descifrar los datos, quedando vulnerable nuevamente la confiabilidad de los datos.

La solución para estos problemas es la generación de llaves a través de funciones físicas no clonables (PUFs). Una PUF se encarga de generar claves aleatorias no predecibles, utilizando las variaciones intrínsecas de fabricación de un chip para generar una identidad única, lo que hará que sea imposible de clonar. Se pueden considerar estas características intrínsecas como un carácter biométrico único de un

CI físico. Con estas claves generadas se podrá hacer biometría cancelable a cualquier carácter biométrico. En esta tesis se enfocara únicamente al carácter biométrico del electrocardiograma (ECG).

1.2. Objetivo general

Experimentar, analizar e implementar arquitecturas basadas en retardo con enrutado automático y manual; posteriormente concluir cual de ellas es la mejor para la generación de números aleatorios con ayuda de las métricas de evaluación PUF y las métricas *National Institute of Standars and Technology* (NIST); finalmente realizar la biométrica cancelable basada en señales ECG.

1.2.1. Objetivo específicos

- Implementar con enrutado manual y automático las siguientes arquitecturas de Arbiter PUF basadas en retardo:
  - Arbiter PUF.
  - Arbiter PUF XOR.
  - Arbiter PUF Feed Forward.
  - Arbiter PUF Lightweight Secure.
  - Arbiter PUF XOR Lightweight Secure.
  - Arbiter PUF Lightweight Secure Feed Forward.
- Implementar la métrica *Dynamic Time Warping* (DWT) y Distancia Euclidiana como método de clasificador.
- Realizar la plantilla cancelable con la extracción de características de la señal ECG y la base de datos obtenida de las respuestas PUF.
- Proponer una nueva técnica de cancelación que mejore las Medidas de evaluación para técnicas de protección de plantillas biométricas (BPT).
- Implementar el diseño en dos tipos de modelos de FPGA para observar y analizar las diferencias en los resultados

1.3. Estado del arte

La biometría es utilizada en muchos ámbitos de la vida cotidiana como son en las licencias, identificación oficial, pasaporte, acceso a áreas restringías etc. La biometría es un campo tecnológico en pleno desarrollo que utiliza rasgos biométricos para establecer la identidad de una persona [20]. Es importante mencionar que estos datos son fáciles de robar y por lo tanto, se puede hacer mal uso de esta información. En la literatura se encuentran técnicas de biometría cancelable donde el objetivo principal es proteger los datos de una persona a través de una serie de algoritmos que se crean para evitar el robo de identidad.

A continuación se presentan algunos trabajos basados en técnicas de biométrica cancelable.

Non-Invertible cancellable fingerprint template for fingerprint biometric [21]: Este artículo muestra la técnica llamada Delaunay Triangulation. A partir de esta técnica se extraen los puntos característicos de la huella dactilar modificada con una representación binaria de algunos puntos característicos aleatorios. Esto genera que la plantilla de puntos característicos sea modificada y por lo tanto sea no invertible.

Alignment-free cancelable fingerprint templates with dual protection [22]: Se realizó una plantilla de biometría cancelable con la combinación del modelo ventana-desplazamiento-XOR y la transformada wavelet discreta (DWT) parcial. La DWT parcial introduce la no invertibilidad y mejora la precisión del reconocimiento.

Hardware Implementation of Cancellable Biometric Systems [23]: En esta investigación se propuso una transformación de biometría cancelable con mapas caóticos 3D para una base de datos de rostros y huellas dactilares donde se logra una alta seguridad. También se implementó este algoritmo en una FPGA.

Bio-PUF-MAC authenticated encryption for iris biometrics [24]: Se presentó un método basado en funciones físicas no clonables (PUFs) y la transformada wavelet discreta (DWT) para una base de datos del iris. La ventaja de este trabajo es la autenticación PUF a nivel dispositivo lo que hace que no exista posibilidad de robo de identidad.

A novel cancellable Iris template generation based on salting approach [25]: Este artículo trabajó en el enfoque de aproximación salting que va a depender de la mezcla de código binario original del iris con un patrón sintético utilizando la operación XOR.

A PUF- and Biometric-Based Lightweight Hardware Solution to Increase Security

at Sensor Nodes [26]: Existen varios enfoques para usar una PUF, en este trabajo se presentó el desarrollo del diseño SRAM PUF para ocultar claves secretas de huellas dactilares.

Cancellable biometric system based on linear combination of trigonometric functions with special application to forensic dental biometrics [20]: Este trabajo generó plantillas biometricas dentales cancelables a través de una combinación lineal de funciones trigonométricas (suma de seno y coseno) y una clave secreta del usuario para proteger sus identidades. La función es fácil de evaluar pero difícil de invertir.

One-factor Cancellable Biometrics based on Indexing-First-Order Hashing for Fingerprint Authentication [27]: Se empleó la técnica de Hashing de primer orden de indexación, ocupando como base de datos huellas dactilares. La propuesta se evaluó a cuatro criterios de protección de plantillas, como son, la no invertibilidad, la renovabilidad, la no vinculabilidad y el rendimiento de la precisión.

One-factor Cancellable Scheme for Fingerprint Template Protection: Extended Feature Vector (EFV) Hashing [28]: Se propuso una plantilla cancelable de huella dactilar con el factor Hashingh de vector de características extendido (EFV), donde utiliza una clave permutada que se separa de los datos biométricos para ocuparlo como identificador. Considera dos entradas para el proceso de inscripción, un vector de características biométrica y un vector binario aleatorio.

Cancellable multimodal biometric user authentication system with fuzzy vault [29]: Se aplicó la transformación del algoritmo de distorsión en imágenes del rostro y huella dactilar para cambiar en ángulo de una posición a otra. Con el fin de que la imagen transformada sea ocupada para evaluación y la imagen original almacenada para otros fines.

Discrete Transforms and Matrix Rotation Based Cancelable Face and Fingerprint Recognition for Biometric Security Applications [30]: Se introdujo un método de transformadas discretas y rotación de matrices. La desventaja que tienen las transformadas es que son invertibles y por lo tanto no es funcional para la biometría cancelable, por lo que se propone utilizar múltiples rotaciones de matrices en un dominio de transformación para asegurar un proceso irreversible y así garantizar una mayor seguridad.

Securing Minutia Cylinder Codes for Fingerprints through Physically Unclonable Functions: An Exploratory Study [31]: Destacando el hardware propone la combinación de P-MCCS y PUF generado a partir de SRAM. La ventaja sobre otros autores es que no almacena ninguna clave secreta, ya que la única forma de acceder a estos

datos es cuando el dispositivo este encendido. El autor hace mención, en que tener un vector de características más largo serán más precisos y con la combinación PUF más seguros.

A Secure Multimodal Authentication System Based on Chaos Cryptography and Fuzzy Fusion of Iris and Face [32]: Se propone ocupar protección caótica basado en mapas logísticos de Henon y 2D. No hay plantillas de datos expuestas, ya que estas solo pueden ser reveladas durante los pasos internos del sistema.

Efficient hardware implementation of AES algorithm using bio metric key [33]: Se propone un sistema criptográfico tomando como referencia una base de datos del iris y aplicando el algoritmo AES para crear una plantilla cancelable.

Cancelable Face Using Chaos Permutation [34]: Se crea un nuevo esquema para la protección de plantillas con el Biohashing que emplea mapa de caos logístico para permutar los vectores de características.

En el diagrama de la figura 1.3 y en la tabla 1.1 se presenta un resumen de los trabajos relacionados con biometría cancelable mencionados anteriormente.

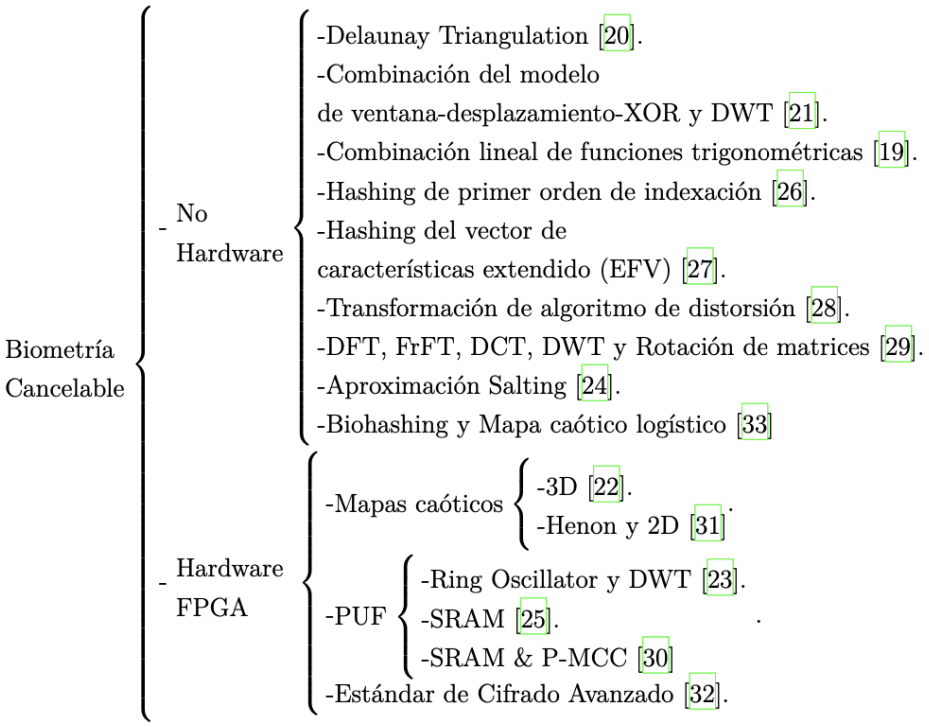


Figura 1.3: Recopilación de artículos asociados con biometría cancelable.

Autor	Año	Base de datos	Característica Biométrica	Estrategia	Resultados
Diana Torres	2020	BIDMC PPG and Respiration Dataset	ECG	PUF	EER=0.071 AUC= 98.32
Muhammad	2020	FCV2002 FCV2004	Huella Dactilar	Combinación del modelo de ventana -desplazamiento -XOR & la Transformada Wavelet discreta parcial	Protocolo 1v1 FVC2002 DB1-EER=0 DB2-EER=0 DB3-EER=1.63 FVC2004 DB1-EER=7.35 DB2-EER=4.69 Protocolo FVC FVC2002 DB1-EER=1.57 DB2-EER=1.50 DB3-EER=4.93 FVC2004 DB1-EER=10.49 DB2-EER=8.62
Lamiaa	2020	ORL	Rostro & Huella Dactilar	Mapas Caóticos 3D	EER=6.2460x10-13 AROC=0.99
Sivasankari	2020	CUHK: Multimedia Laboraty	Iris	Ring Oscillator & Transformada Wavelet Discreta (DWT)	
Abeer	2020	Imágenes Faciales: ORL, FRERET & LFW Huellas Dactilares FVC 2002	Rostro & Huella Dactilar	Transformada discreta de Fourier (DFT), Transformada Fraccional de Fourier (FrFt), Transformada discreta de Coseno (DCT), Transformada Wavelet Discreta (DWT) & Rotación de Matrices	AROC=0.998 EER=0.0023 FAR= 0.008 FRR=0.003
Ahmed	2020	CASIA V3	Iris	Aproximación salting	EER=0.43 GAR=0.9957
Mahroosh	2019	Mandíbula dental auto-consulta	Plantilla Dental	Combinación lineal de funciones trigonométricas	EER=1.86 GAR=98.62
Amit,Dalton & Shyamosree	2019	FCV2002	Huella Dactilar	Delaunay Triangulation	DB1-EER=1.2 DB2-EER=2.1
Rosario Arjona	2018	FVC2002 FVC2000	Huella Dactilar	SRAM PUF	FVC2000 DB2a-EER=0.04 FVC2002 DB1a-EER=0.22
Jihyeon	2018	FVC2002 FVC2004	Huella Dactilar	Hashing de primer orden de indexación	FVC2002 DB1-EER=0.22 DB2-EER=0.30 DB3-EER=1.63 FVC2004 DB1-EER=1.69 DB2-EER=4.08 DB3-EER=1.97
Ming Jie Lee	2018	FVC2002 FVC2004	Huella Dactilar	Hashing del vector de características extendido (EFV)	FVC2002 DB1-EER=0.30 DB2-EER=0.56 FVC2004 DB1-EER=2.42 DB2-EER=6.27
Rosario Arjona	2018	FVC2002	Huella Dactilar	SRAM PUF & P-MCC	FVC2002 DB2a-EER=1.23 FVC2002 DB3a-EER=4.97
Marwa	2017	CASIA Iris Faces94 Rostro	Iris Rostro	Mapas lógisticos Henon y 2D	FAR=0.0345 % FRR=0.001 %
Soruba	2016		Rostro & Huella Dactilar	Transformación de algoritmo de distorsión	FAR=2 % FRR=1.8 % GAR=98.1 % %
Sridevi	2015	CASIA	Iris	AES	
Sara Nazari	2014	ORL	Rostro	Biohashing Mapa caótico logístico	

Tabla 1.1: Tabla comparativa de trabajos relacionados con biometría cancelable.

Como se puede notar, estas investigaciones proponen diferentes técnicas y caracteres biométricos con la finalidad de lograr una tasa de reconocimiento alta y una tasa de error mínimo. Para esta tesis se consideró la última referencia del autor [35] donde se propone utilizar la misma base y técnica, pero utilizando el retardo como base de la arquitectura PUF.

1.4. Organización de la tesis

La tesis está organizada de la siguiente forma:

- El capítulo 1 contiene la introducción de está investigación donde se muestra la justificación, los objetivos específicos y generales en los que se basa este trabajo.
- El capítulo 2 presenta una recopilación de investigaciones relacionadas con la biometria cancelable, conceptos básicos que son utilizados a lo largo de la tesis, mostrando las ventajas y desventajas de las funciones físicas no clonables así como los diferentes diseños empleados.
- El capítulo 3 se enfoca en la metodología para la implementación de Arquitecturas PUFs con enturado manual y automático. Haciendo énfasis en la manipulación de las celdas básicas, también llamadas celdas primitivas, que utiliza el FPGA para realizar la implementación de los diseños digitales.
- El capítulo 4 expone los resultados de la biometría cancelable utilizando dos técnicas diferentes para este proceso. Además de tablas donde se visualizan las métricas BPT para determinar si cumplen con una buena seguridad.
- El capítulo 5 contiene las conclusiones generales de la tesis presentada con las mejoras y algunas recomendaciones para trabajos a futuros.

Capítulo 2
Marco Teórico

2.1. FPGA

Una *Field Programmable Gate Array* (FPGA) son dispositivos electrónicos programables para implementar circuitos digitales y realizar una gran variedad de tareas específicas. Pueden ser reprogramadas para realizar distintas aplicaciones e implementar soluciones en hardware y software.

Una FPGA esta compuesta por bloques E/S programables que están diseñadas para interconectar señales internas de la FPGA con señales externas. Internamente, el FPGA contiene bloques lógicos configurables (CLB) que consta de uno o más circuitos lógicos programables para la implementación de circuitos secuenciales y combinacionales [4] como se muestra en la figura 2.1.

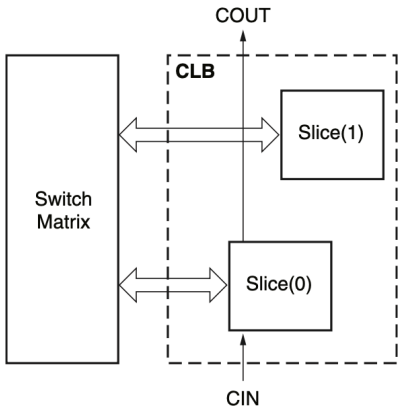


Figura 2.1: Bloques lógicos programables [3].

Las Look Up Table (LUT) mostrada en la figura 2.2 nos permiten implementar tablas de verdad de hasta 6 entradas creando así la posibilidad de implementar

cualquier compuerta lógica o circuito combinacional.

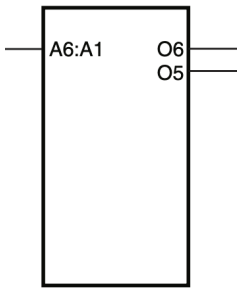


Figura 2.2: LUT especificada para Spartan 6 [4].

En la figura 2.3 se muestra la arquitectura general de una FGPA. Dentro de la FPGA se encuentran relojes del sistema para implementar diseños síncronos para la E/S y para el funcionamiento interno. El funcionamiento síncrono utiliza un flanco de reloj para registrar los resultados de la lógica ascendente y mantenerlos estables para su uso por la lógica descendente hasta el siguiente flanco de reloj. El uso de la operación sincrónica permite realizar gráficos de flujo en cadena que procesan múltiples muestras en paralelo. Las interfaces de comunicaciones digitales externas utilizan relojes de E/S para transferir datos hacia y desde la FPGA. También se pueden encontrar bloques de memoria RAM y ser amplias como sea necesario. Otros componentes importantes son los registros o Flip Flops que son utilizados para elementos de retardos de señales o para almacenamiento de información (registro).

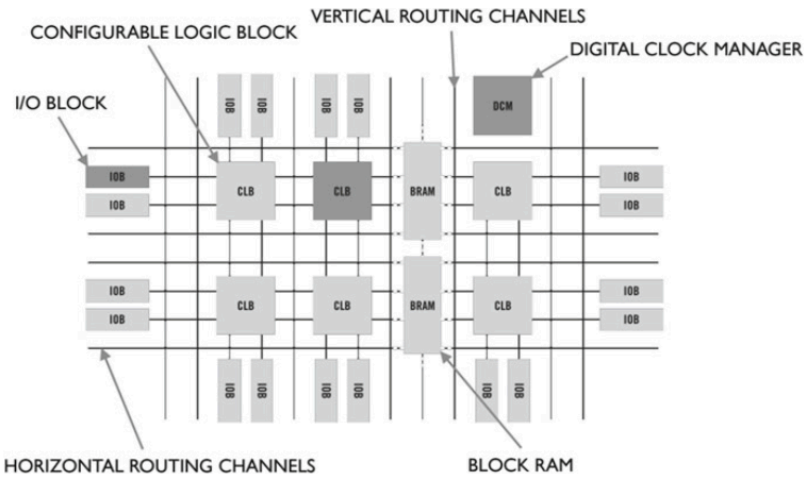


Figura 2.3: Arquitectura FPGA [4].

Cada CLB contiene un par de slices, donde estos no están relacionados directamente, son independientes como se muestra en la figura 2.4. Los Bloques lógicos configurables son variables de acuerdo a cada familia de chip de la FPGA.

La estructura de las tarjetas de Xilinx, etiquetan los slices con coordenadas haciendo uso de las letras **X** seguida de un número para hacer referencia a la columna. Las letras **Y** seguida de un número, hacen referencia a las filas de los slices. Con ello se tiene una organización perfecta y se puede hacer uso de cualquiera de ellas con solo etiquetar el slices que se requiere. En la figura 2.4 se observa como están posicionados los slice X y Y.

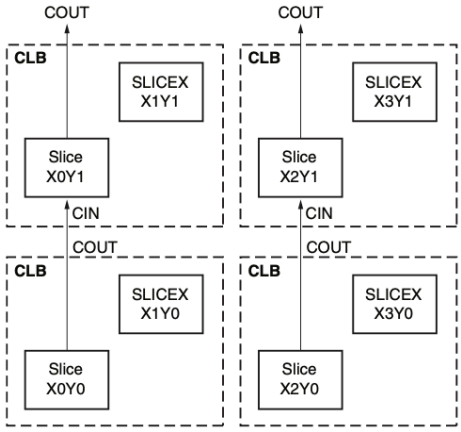


Figura 2.4: Coordenadas de CLB [3].

Al momento de hablar de slice, se encuentran 3 tipos que básicamente varían en la cantidad de recursos. SLICE**X**, tiene 4 LUT y 8 elementos de almacenamiento; SLICE**L** contiene una estructura de acarreo y multiplexores; SLICE**M** contienen estructura de acarreo, multiplexores y la posibilidad de usar LUT's como RAM distribuida. En la figura 2.5 se aprecia una tabla de las características de cada slice mencionado.

Feature	SLICEX	SLICEL	SLICEM
6-Input LUTs	√	√	√
8 Flip-flops	√	√	√
Wide Multiplexers		√	√
Carry Logic		√	√
Distributed RAM			√
Shift Registers			√

Figura 2.5: Características de SLICE X,L,M [3].

2.2. Floorplanning

Es una herramienta esencial ya que permite visualizar como están distribuidos los componentes de un diseño y verificar que estos, estén en la posición deseada cuando se hace un enrutado manual. En la figura 2.6 se muestra la estructura interna de una FPGA de la familia Spartan 6 con sus respectivos componentes y la forma en que estos están distribuidos. Es una herramienta fundamental para el desarrollo de esta tesis.

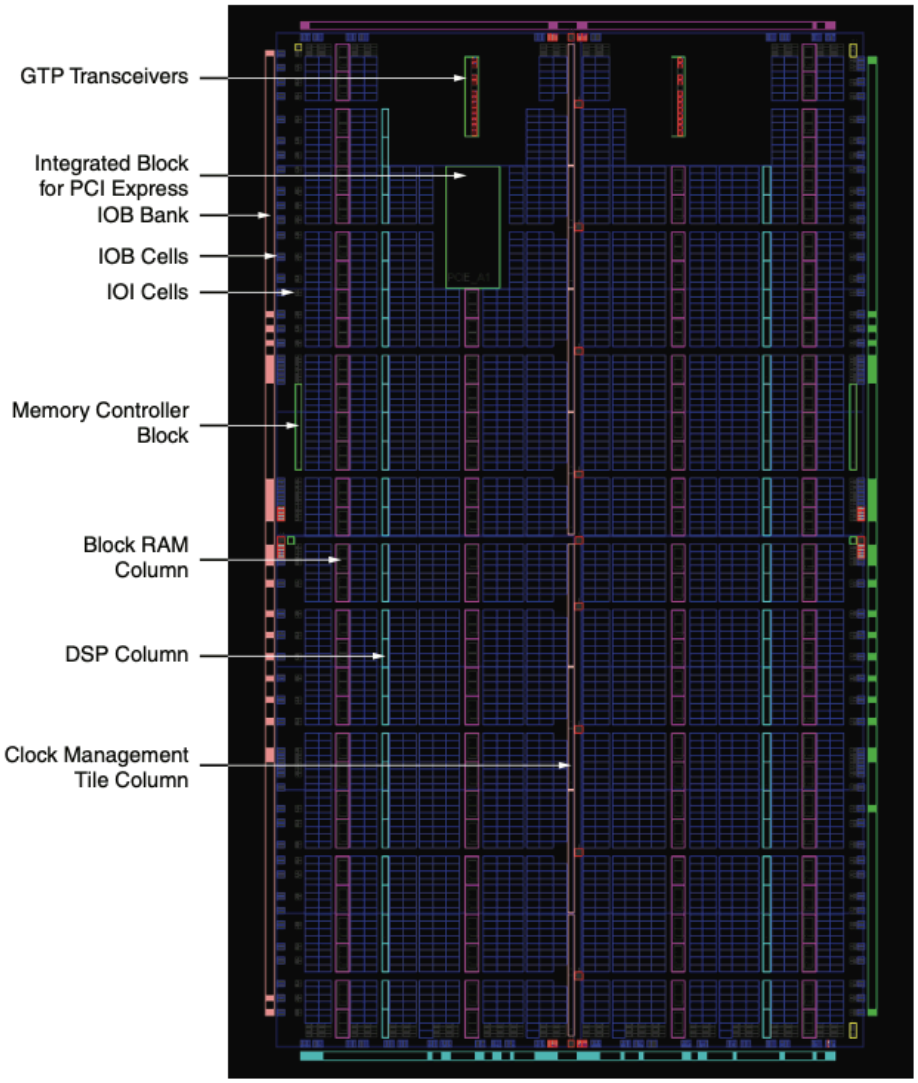


Figura 2.6: FloorPlan vista desde PlanAhead [3].

Haciendo un zoom más preciso, se pueden apreciar los Slices y los componentes

que contienen tal y como se muestra en la figura 2.7.

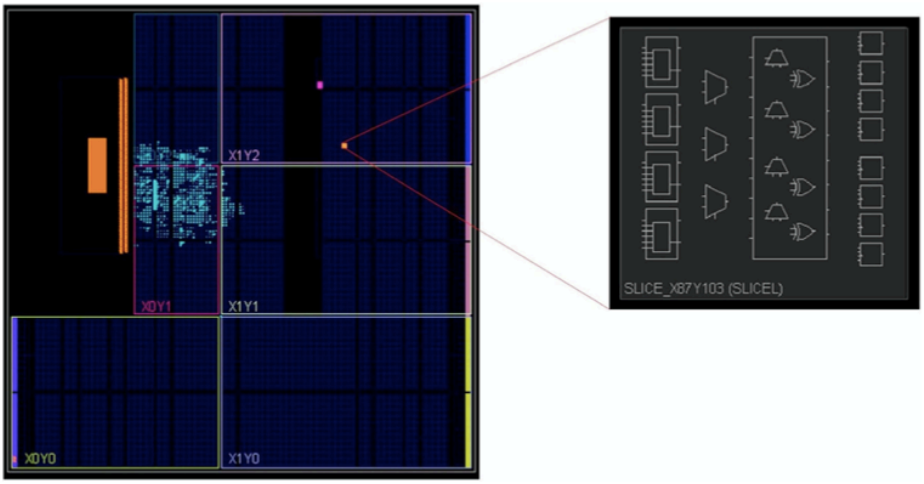


Figura 2.7: Slice tipo X [5].

2.3. Funciones Físicas no clonables (PUFs)

Las funciones físicas no clonable (PUFs) se pueden definir como una característica inherente e inclasificable de un objeto [11]. En otras palabras una PUF es una identificación única de un circuito que se obtiene a partir de las variaciones de los procesos de fabricación. Estas variaciones intrínsecas son por ejemplo oscilaciones en la longitud y anchura, el grosor del óxido y los niveles de dopaje en transistores lo que da como conclusión no poder crear dos dispositivos idénticos. Se puede tener 2 o más dispositivos PUF que tengan la misma estructura o configuración pero el comportamiento de las salidas sera diferente debido a estas variaciones que se presentan durante el proceso de fabricación. Una gran ventaja que se tiene con las PUFs es que éstas deben ser reconfigurables para poder generar diferentes salidas (conocidas como respuestas (R)) con diferentes entradas (conocidas como desafíos (C)) lo cual indica que las claves son generadas solo cuando se necesitan y no necesariamente son almacenadas en un fuente de memoria.

Dentro de las ventajas que se tiene con las PUFs es que aprovecha las variaciones aleatorias de fabricación, los efectos ambientales, el envejecimiento y el ruido térmico. Estos factores permiten que una PUF resista la falsificación y la duplicación y al mismo tiempo crea claves únicas para el dispositivo que no son visibles cuando éste está apagado. También se evita el uso de la memoria no volátil para almacenar

llaves y por lo tanto evitar el robo de claves criptográficas. El par desafío-respuesta también llamado CRP (Challenge-Response Pair) es posible después de la fabricación del dispositivo.

2.3.1. Evaluaciones internas

Evaluar una PUF internamente, nos proporciona dos ventajas importantes mencionadas por [11]:

1.- Tener un equipo de medición integrado en cada diseño PUF, significa que cada diseño PUF puede evaluarse a sí misma sin ninguna restricción externa. Estas evaluaciones son más precisas ya que no hay interferencias externas que provoquen errores de medición.

2.- Tener evaluaciones internas en una PUF provoca que las respuestas se originen dentro del chip incrustado, es decir, mientras una instancia no revele una respuesta al mundo exterior, se considera una llave secreta.

Una posible desventaja de la evaluación interna es que hay que confiar en el equipo de medición incorporado, ya que es imposible verificar externamente si la medición tiene lugar como se espera, a diferencia de la evaluación externa, en la que a menudo es posible observar la medición en curso

2.3.2. Fuertes y Débiles

Dentro de los diseños PUFs, se pueden clasificar como fuertes o débiles. El primero se basa en proporcionar a un impostor acceso a una instancia PUF durante un determinado tiempo donde el adversario no logra conocer la respuesta. Esto nos lleva a considerar 2 puntos fundamentales

1.- El PUF debe tener una gran cantidad de retos muy grandes para que el impostor no tenga el tiempo suficiente para poder consultar todos los retos.

2.- Es imposible reconstruir un modelo preciso del PUF basado en los pares reto-respuesta observados, o en otras palabras el PUF es impredecible.

Aquellos que no satisfacen estos dos puntos cruciales son denominados como PUFs débiles.

2.3.3. Atributos

De acuerdo a [11] se enumeran una serie de atributos para la evaluación de las PUFs.

- Constructibilidad: Todos los diseños PUFs considerados son construibles ya que para todas ellas existen implementaciones conocidas, sin embargo algunas serán más difíciles que otras.
- Evaluabilidad: Todos los diseños PUFs, al ser implementados en un chip se pueden evaluar a través de una serie de experimentos para obtener resultados.
- Reproducibilidad: Es una propiedad de los PUFs, en donde al evaluar un diseño PUF **n** veces, se espera tener las mismas respuesta ante los mismos retos de la PUF.
- Unicidad: La unicidad se encarga de diferenciar las respuestas de una PUF respecto a otra PUF con la misma familia del chip.
- Identificabilidad: Es una combinación de la reproducibilidad y la unicidad. Como se ha mencionado anteriormente, a pesar de ser la misma PUF con la misma familia del chip, las respuestas serán totalmente diferentes.
- Físico no clonable: Debido a los procesos físicos y aleatorios de fabricación, hacen que cada chip tenga una característica única y diferenciable entre un conjunto de chips de la misma familia. Es imposible poder clonar un chip ya que las variaciones son a nivel microscópicos e incluso submicroscópico.
- Impredecibilidad: Se espera que en una instancia PUF las respuestas sean altamente aleatorias e imprevisibles.
- Matemática no clonable: Las respuestas PUF son impredecibles y por lo tanto no pueden ser modelados matemáticamente. En caso de que las respuestas se ven comprometidas, el impostor debe memorizar una cantidad de CRP para poder realizar un algoritmo de predicción.
- Verdadero no clonable: Es una combinación entre lo físico y matemático no clonable, donde se concluye que es difícil clonar una instancia PUF observándola desde dos perspectivas diferentes.

- Unidireccionalidad: Para una instancia PUF es fácil crear un conjunto de CRP, pero hacerlo de manera inversa, donde a partir de una respuesta encontrar su desafío correspondiente es difícil.
- Evidencia de manipulación: Cuando una instancia PUF se ve involucrada a una transformación física, se debe actuar de manera adecuada borrando toda la información confidencial para evitar mal uso de esos datos.

En la literatura se puede encontrar diferentes configuraciones de funciones físicas no clonables, incluso se pueden combinar dos configuraciones diferentes para obtener un único diseño, tomando el nombre de funciones físicas no clonables híbridas. Dentro de las funciones físicas no clonables, podemos clasificarlas como PUFs basadas en silicio (electrónicos) y no silicio (no electrónicos) donde la identidad de cada dispositivo se obtiene analizando explícitamente cada configuración. En el esquema de la figura 2.8 se observa más a detalle esta clasificación.

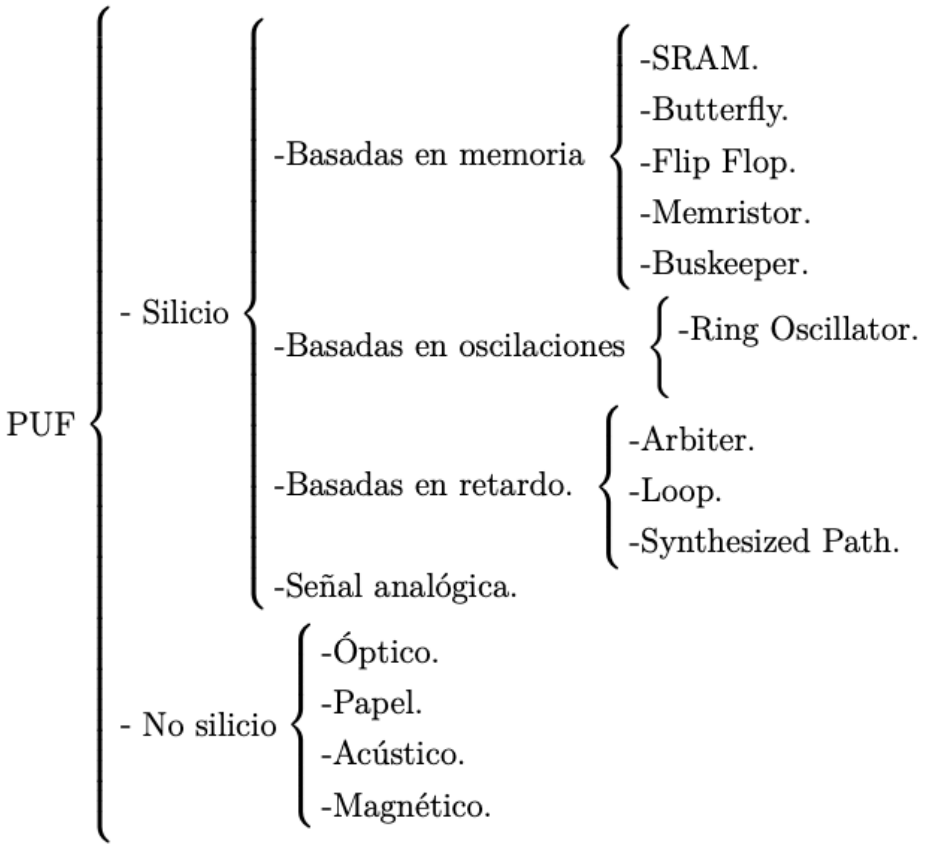


Figura 2.8: Clasificación PUF

2.3.4. Silicio: Basadas en memoria

2.3.4.1. SRAM

Es una memoria estática con la capacidad de almacenar un valor binario en cada una de sus celdas cada vez que se enciende la SRAM. Cada una de estas celdas tiene su propio estado prioritario al ser encendido [36]. En la figura 2.9 se aprecia el diseño de esta memoria, que se construye por dos inversores acoplados en cruz donde explotan la imposibilidad de saber el valor inicial de las celdas de memoria ya que son causadas por las variaciones de enrutamiento interno de la celda. Por lo tanto la respuesta de una SRAM produce un valor aleatorio, que se puede usar como una identidad del circuito. Las celdas que toman valores entre 1 o 0, son celdas estables, mientras que las que no están definidas, son consideradas inestables [37].

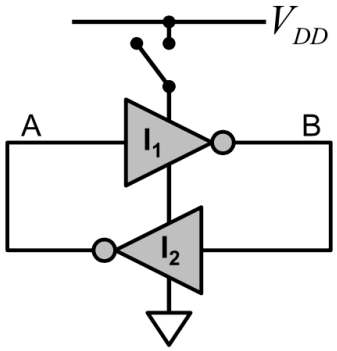


Figura 2.9: Celda SRAM: circuito lógico [6].

2.3.4.2. Butterfly

Este diseño surge a partir de que en las tarjetas FPGA las SRAM tienen un valor inicial predefinido por lo tanto no tiene aleatoriedad. Este circuito imita el comportamiento de la SRAM conectando dos latches en forma de cruz [7] tal y como se muestra en la figura 2.10. Los Latches tienen la señal de preset y clear donde están conectados mediante una señal de excitación. La salida de un latch sirve como entradas para el otro latch haciendo un circuito de lazo cerrado [38]. Su funcionamiento es que la señal de excitación se mantenga en un estado alto durante ciertos ciclos de reloj, para que el circuito empiece a oscilar, una vez que la señal de excitación se cambia de estado bajo, la salida del circuito nos devolverá un 1 o 0.

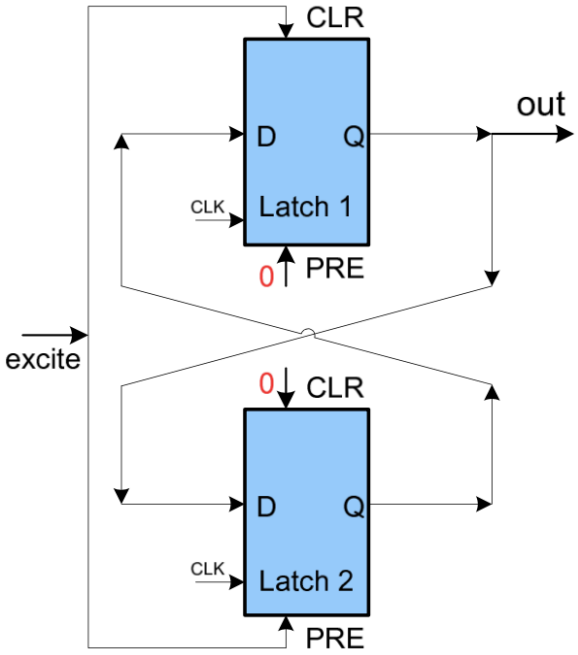


Figura 2.10: Butterfly: acoplamiento en cruz [7].

2.3.4.3. Flip Flop

El PUF Flip Flop propuesto por [6] toma valores al momento de encendido de los Flip Flops, como ocurre en el SRAM. Una de las grandes ventajas que tiene este diseño es que pueden duplicarse por todo un circuito integrado, lo que hace que sea difícil al momento que un intruso quiera manipular este diseño. Con ayuda de esta idea, es posible crear diseños para poder leer y mantener los valores de encendido de los Flip Flops en una FPGA [36, 6]. En la figura 2.11 se aprecia una celda de Flip Flop D.

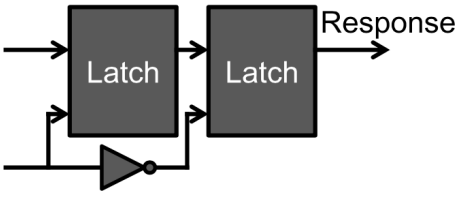


Figura 2.11: Celda Flip Flop D [6]

.

2.3.4.4. Memristor

El comportamiento del memristor es como una resistencia con carga controlada con memoria. En [8] se utiliza la tecnología de memristores para la creación de funciones físicas no clonables a través de la variación aleatoria del proceso. También se estudia el estado lógico impredecible de las celdas del memristor dentro de una región indefinida como se muestra en la figura 2.12. Las operaciones de memoria de los memristores van a depender de la duración de tiempo de acceso y del valor de tensión de la alimentación [36].

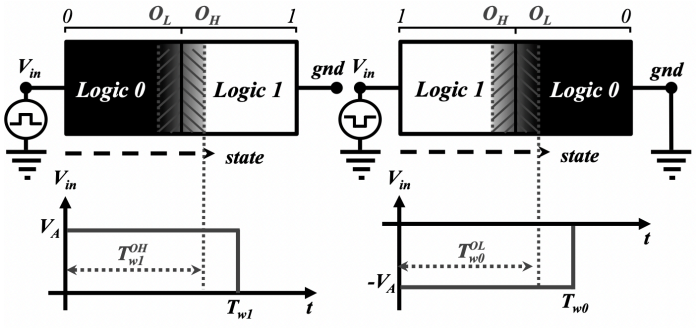


Figura 2.12: Escritura de 1's y 0's [8]

.

2.3.4.5. Buskeeper

Es una latch sin señales de control que se muestra en la figura 2.13, con la idea de ser utilizados posteriormente con buses en chips que tienen múltiples controladores como se muestra en la figura 2.14. Este PUF fue propuesto por [9], donde se comparan los resultados con un diseño PUF Flip Flop D, obteniendo resultados más eficientes en la cantidad de recursos de hardware. La gran ventaja de este PUF es que es muy pequeño y tiene menor complejidad que los Flip Flops D. La idea principal de este PUF esta basado en guardar valores iniciales al momento de inicializar la memoria [36].

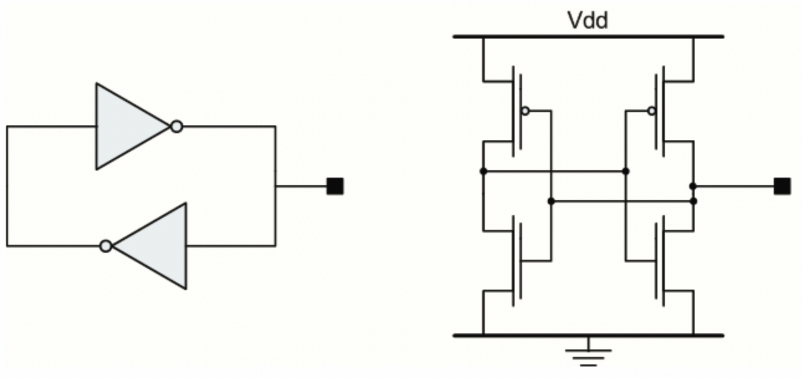


Figura 2.13: Celda Buskeeper de alto nivel y a nivel transistor [9].

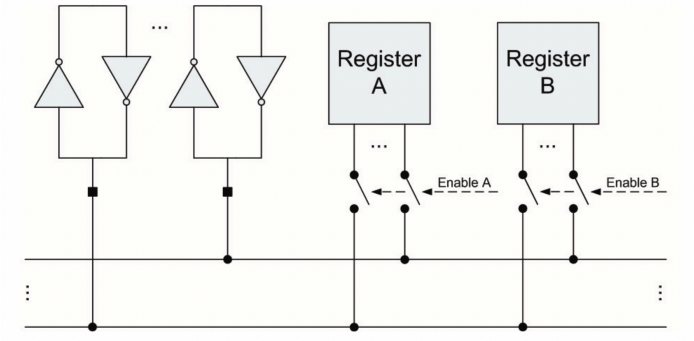


Figura 2.14: Ejemplo de implementación Buskeepers [9].

2.3.5. Silicio: Basadas en oscilaciones

2.3.5.1. Ring Oscillator

El oscilador de anillo se crea por ciclos de retardos con etapas impares de inversores y un contador de frecuencia. El oscilador esta formado por una compuerta NAND e inversores conectados en serie como se observa en la figura 2.15. Al final del último inversor, se retroalimenta a la entrada para que oscile el circuito. Mediante un contador de frecuencia se detectan los flancos ascendentes durante las oscilaciones y almacena el total de flancos durante un periodo de tiempo.

Lo primero que se realiza es comparar dos osciladores con las mismas características y simetría, posteriormente les asigna una señal de entrada durante un tiempo determinado para que empiecen a oscilar, sin embargo, no oscilarán a la misma frecuencia, debido a las variaciones del procesos de fabricación. Al final se revisan ambos

contadores y se selecciona el mayor, lo que dará como resultado una probabilidad de obtener 1 o 0 como respuesta. Su implementación ha sido la más popular [10].

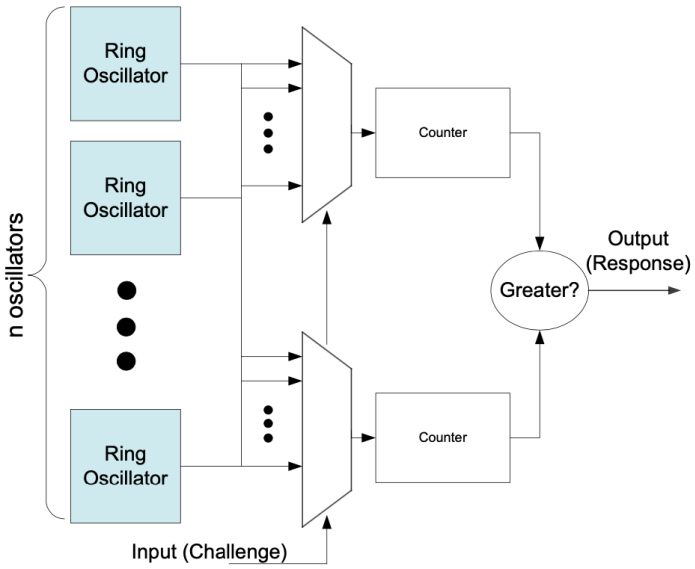


Figura 2.15: Ring Oscillator PUF [10].

2.3.6. Silicio: Basadas en retardos

2.3.6.1. Arbiter

Se puede apreciar como una carrera que se basa en dos caminos idénticos y al final, con ayuda de un árbitro se determina cuál camino fue más rápido como se muestra en la figura 2.16 y 2.17. Se compone de switches de conmutación y un arbitro que es un Flip Flop o latch. El bloque de conmutación tiene dos multiplexores controlados por un solo selector. Cada switch de conmutación debe estar conectado en serie con el siguiente switch de conmutación y así hasta **n-bloques**, dependiendo del diseño para construir dos caminos simétricos, y al final se coloca el árbitro. Con el selector que controla ambos multiplexores se puede decidir si el camino es de forma recta ($S = 0$) o en forma cruzada ($S = 1$). Debido a la latencia aleatoria de cada bloque de interruptores que existe por las variaciones de fabricación, hay una ligera preferencia entre un camino del otro [39].

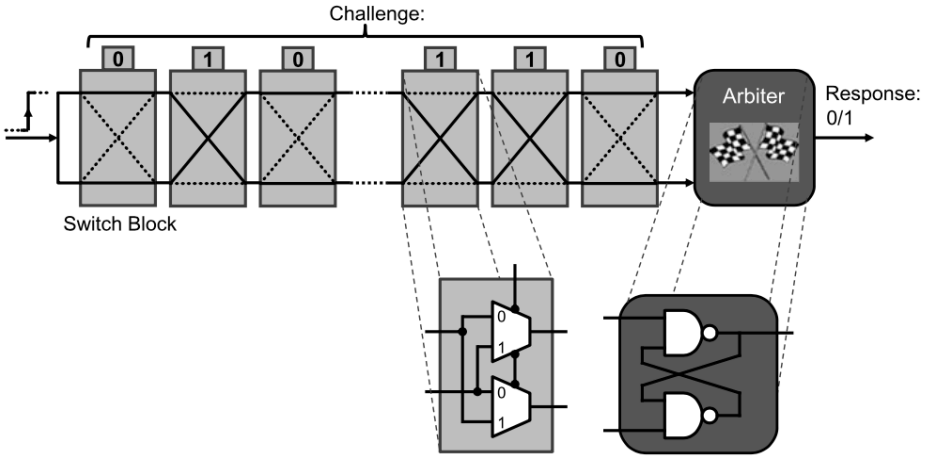


Figura 2.16: Arbiter PUF con un solo desafío [11].

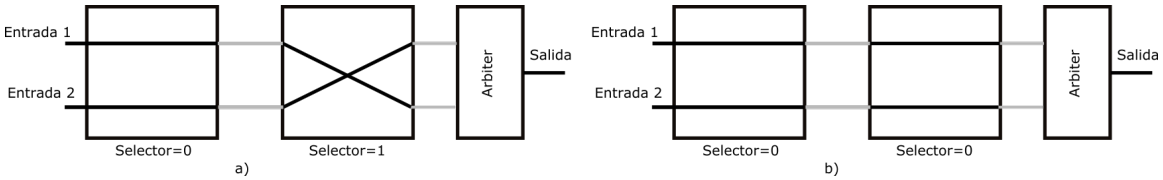


Figura 2.17: Arbiter PUF con bloques de conmutación.

La salida de cada circuito es un único bit, si se requieren más bits de salida, entonces se pueden realizar más circuitos en paralelo con la misma entrada y generar una cadena de bits. Este diseño se puede realizar de diferentes formas para tener una mejor arquitectura PUF y por lo tanto, mejorar la generación de llaves aleatorias.

La primera arquitectura a mencionar es el **Arbiter PUF XOR**. Este diseño consiste en unir varias filas de los Arbiter PUFs simples a un XORing para obtener solo un bit de salida como se aprecia en la figura 2.18. La ventaja de esta arquitectura es que el circuito sea más resistente a los ataques de modelados. Las características principales de este diseño es la longitud del bit de desafío y el tamaño de la entrada del XOR, que indica el número de filas que serán utilizadas [39, 40].

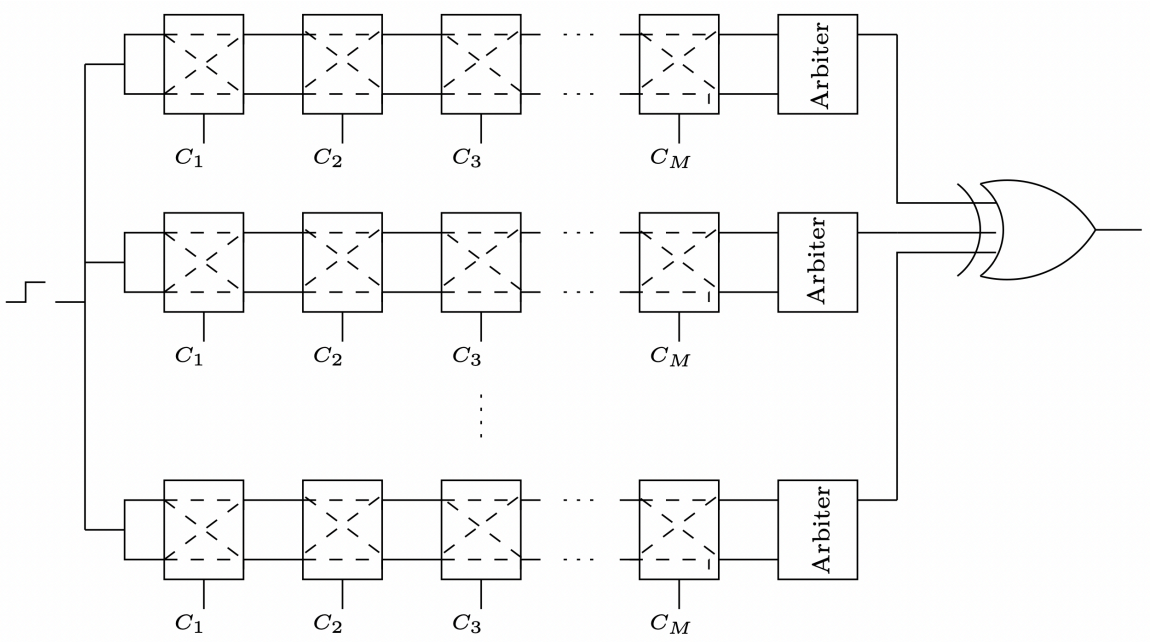


Figura 2.18: Arbiter PUF XOR con 3 Arbiter simples conectados a un XOR para obtener solo un bit [12].

Otra alternativa de Arbiter PUF es **Arbiter PUF Feed Forward**, que surge a partir de las inquietudes sobre la clonación de PUF, por ejemplo, que un adversario pueda clonar todas las combinaciones de desafíos-respuestas; puedan intentar fabricar una PUF para obtener los mismos desafíos-respuestas, aunque este intento es muy difícil de realizar debido a las variaciones de fabricación del chip; o incluso modelar el comportamiento del PUF y obtener los mismos desafíos-respuestas [13].

Este último intento sobre el modelado del comportamiento del PUF ya es un problema resuelto, debido a que al diseño convencional de Arbiter PUF se le agrega una etapa de no linealidad con una retroalimentación hacia adelante, obteniendo que un bit de desafío o varios bits sean determinados por los retrasos de una etapa intermedia en lugar de ser asignados por un usuario. La figura 2.19 muestra el diseño de Arbiter PUF Feed-Forward.

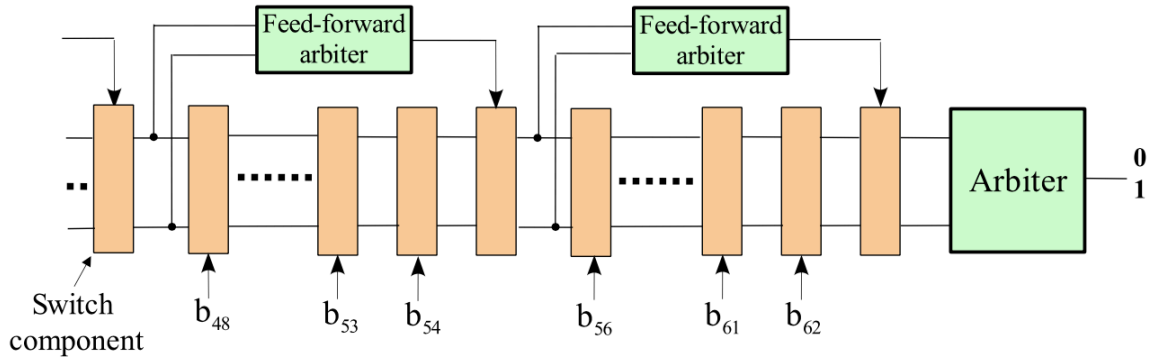


Figura 2.19: Arbiter PUF Feed-Forward con desafíos de bits determinados por retardos intermedios de las etapas [13].

En la figura 2.20 se puede observar el **Arbiter PUF Lightweight Secure** propuesto por [14], donde describe una nueva metodología para el diseño de Arbiter PUF. En ella se añaden múltiples lineas de retardo para la generación de cada bit de respuesta, la combinación de los bits de desafío y la combinación de bits de respuesta obteniendo una mejor seguridad contra la ingeniera inversa o fallos del circuito. Para la combinación de bits de desafío como los de respuesta, se emplea un desplazamiento circular.

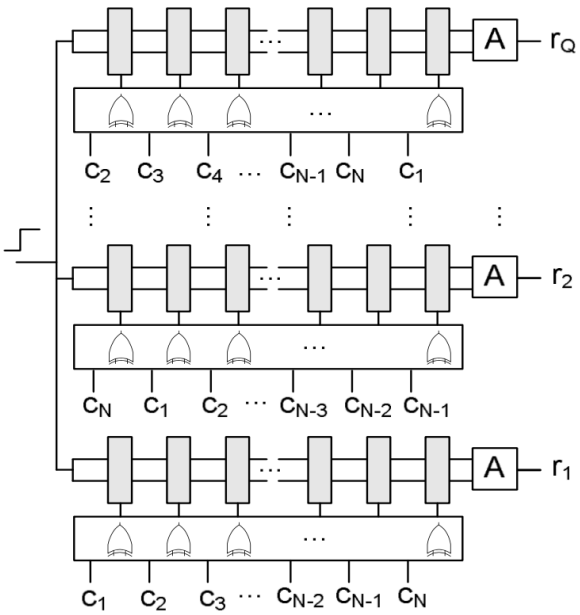


Figura 2.20: Arbiter PUF Lightweight Secure con una conexión de desplazamiento circular en los desafíos y respuestas [14].

A partir de la combinación de las 3 técnicas mencionadas anteriormente, surgen

dos PUF híbridas: **APUF XOR Feed Forward** y **APUF LS Feed Forward**. La primera básicamente es tener dos arquitecturas Feed Forward para generar un solo bit con la salida XOR tal y como se muestra en la figura 2.21.

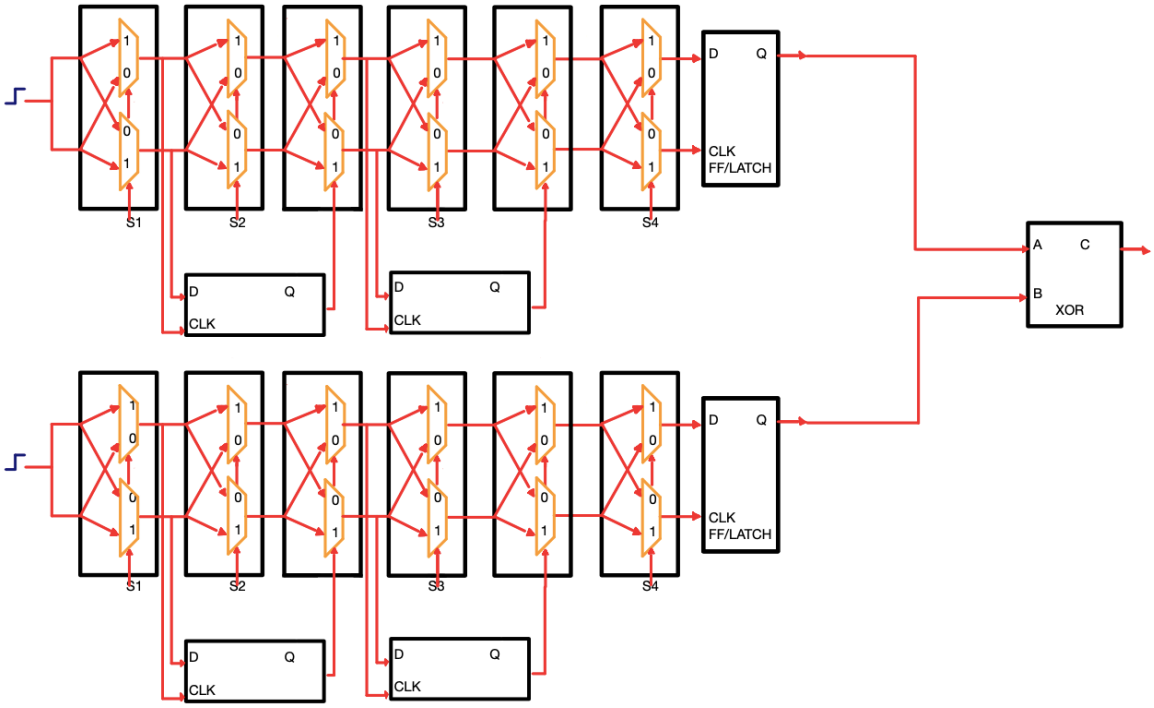


Figura 2.21: Arbiter PUF XOR Feed Forward.

Y la otra es tener dos arquitecturas LS Feed Forward para generar un solo bit con la salida XOR como se ilustra en la figura 2.22.

Estos diseños presentados van de lo más sencillo a lo más complejo, tomando en cuenta que al necesitar mayor seguridad se va a requerir mayor recurso al momento de implementarlo en una tarjeta embebida y por lo tanto se requerirá más tiempo de procesamiento.

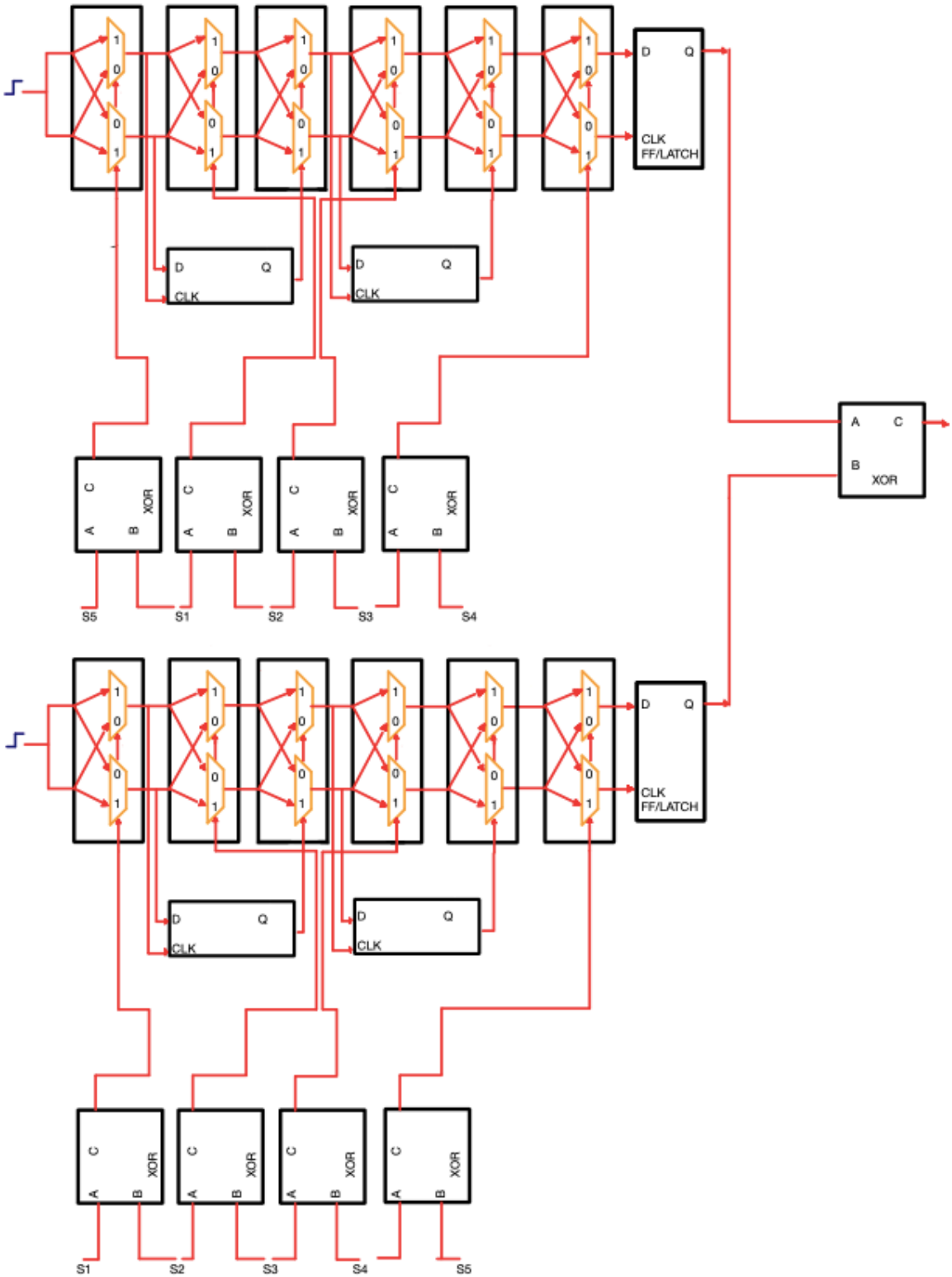


Figura 2.22: Arbiter PUF LS Feed Forward.

2.3.7. No Silicio

2.3.7.1. Óptico

Esta basado en patrones aleatorios de reflexión óptica, con el objetivo de identificar armas estratégicas. Su funcionamiento es hacer que el haz de un rayo láser sea dirigido a un material para que este disperse la luz, generando un fenómeno de esparcimiento tal y como se muestra en la figura 2.23. Este esparcimiento generara un patrón aleatorio y único el cual sera capturado por una cámara para ser procesado digitalmente y generar una identidad única. Para generar distintos tipos de identidades, se utilizan diferentes direcciones del láser [36, 12].

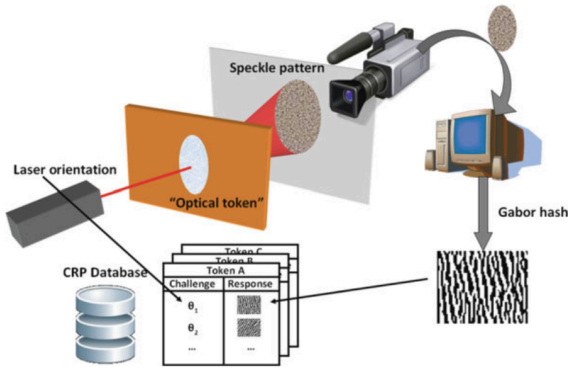


Figura 2.23: Operación del PUF óptico [15].

2.3.7.2. Papel

Este PUF propuesto por [41] consiste en realizar un escaneo a la fibra del papel normal o modificado. El reflejo generado del rayo láser en la estructura irregular de la fibras de la hoja se utiliza como una huella dactilar [15, 36].

2.3.7.3. Acústico

La PUF acústica se construye observando el espectro de frecuencia característico de una línea de retardo acústica. Se extrae una cadena de bits realizando un análisis de componentes. Para obtener un cantidad suficiente de par desafío-respuesta cada unidad se sondea con diferentes frecuencias [36, 15].

2.3.7.4. Magnético

El PUF magnético está basado en la determinación del ruido remanente en un medio magnético a través de la saturación de corriente continua. Este ruido se puede digitalizar y guardar en el mismo medio magnético para obtener una huella digital [36]. Son utilizados para evitar el fraude de tarjetas de crédito [15].

2.3.8. Métricas de evaluación

Para utilizar un diseño PUF orientado a una aplicación de seguridad, es necesario realizarle un examen de métricas de evaluación para determinar si el PUF es capaz de satisfacer las necesidades de la aplicación. Antes de comenzar con las métrica de evaluación, existen dos clases de variaciones. Variación **inter-chip** y la variación **intra-chip**. La primera se debe a variaciones entre chips iguales teniendo en cuenta un promedio de 50 % y la segunda es a variaciones aleatorias ambientales, donde se espera que estas variaciones sean cero lo cual indicaría que no se ve afectada por estos valores externos.

Para evaluar la calidad de un diseño PUF se introducen las siguientes métricas que se son obtenidas mediante la distancia Hamming (HD) y el peso Hamming (HW).

Unicidad: Este parámetro indica la capacidad de distinguir un conjunto de respuestas de una PUF de un chip entre un grupo de chips o sistemas embebidos del mismo tipo. Si se tiene dos chips R_i y R_j, con **n**-bits para un una misma entrada, la distancia Hamming inter-chip media para **k** dispositivos se describe en la eq. 2.3.1 [42, 43].

$$Unicidad = \frac{2}{k(k-1)} \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} \frac{HD(R_i(n), R_{ij}(n))}{n} X100\,\% \tag{2.3.1}$$

- HD: Distancia Hamming.
- R_i: Dispositivo 1.
- R_j: Dispositivo 2.
- k: Número de dispositivos.
- n: Número total de bits.

Una unicidad ideal es del 50 %.

Fiabilidad: Es la eficiencia de la PUF para poder producir la misma respuesta con un determinado desafío en condiciones totalmente diferentes como es la temperatura ambiente o variaciones de voltaje. Este parámetro se calcula con la distancia Hamming intra-chip. Representemos un chip como R_i en condiciones normales y a R'_i en condiciones diferentes, ambas con el mismo desafío, la distancia Hamming intra-chip media para **k** muestras se escribe como se muestra en la eq. 2.3.2 [42, 43].

$$HD_{intra} = \frac{1}{n}\sum_{l=1}^{n}\frac{HD(R_i(n), R'_i(n))}{n} X 100\,\% \tag{2.3.2}$$

$$Fiabilidad = 100\,\% - HD_{intra}$$

- HD: Distancia Hamming.
- R_i: Dispositivo 1 en condiciones normales.
- R'_i: Dispositivo 1 en condiciones diferentes al R_i.
- n: Número total de bits

Una fiabilidad ideal es del 100 %.

Uniformidad: Es la proporción de 0's y 1's en los bits de respuesta de una arquitectura PUF. Una respuesta altamente aleatoria debe contener una uniformidad del 50 %. Eso se puede calcular con ayuda del peso Hamming promedio de las respuesta. Donde $r_{i,l}$ es el **l**-ésimo bit de una respuesta de **n** bits de un chip **i** (eq. 2.3.4) [42, 43].

$$Uniformidad = \frac{1}{n}\sum_{l=1}^{n} r_{i,l} X 100\,\% \tag{2.3.3}$$

- n: Número total de bits

2.4. Biometría basada en Electrocardiograma

La biometría es considerada una ciencia que estudia distancias y geometría de algunas partes del cuerpo para poder identificar a una persona. Estos rasgos son ad-

quiridos mediante sensores para extraer características distintivas. Los rasgos biométricos mas utilizados son el iris, huellas dactilares y el rostro. Esos rasgos biométricos son utilizados para mantener un control de acceso a ciertas áreas permitidas, revisar la autenticación de una persona con sus pasaportes o licencias e incluso poder identificar criminales. Existen otros rasgos biométricos como son las señales electrocardiogramas (ECG) que realizan un registro de la diferencia de potencial de voltaje que es generado por el corazón durante los fenómenos de despolarización y repolarización [44]. Esta señal pasa por un proceso de filtración y amplificación para crear un complejo representativo para cada derivación a partir de una morfología dominante, de modo que se puedan realizar mediciones de las ondas y complejos que forman parte del electrocardiograma.
Una señal ECG tiene la siguiente forma singular, la cual en altas frecuencias se puede apreciar el complejo QRS, mientras que en frecuencias bajas las ondas P, T y U como se observa en la figura 2.24.

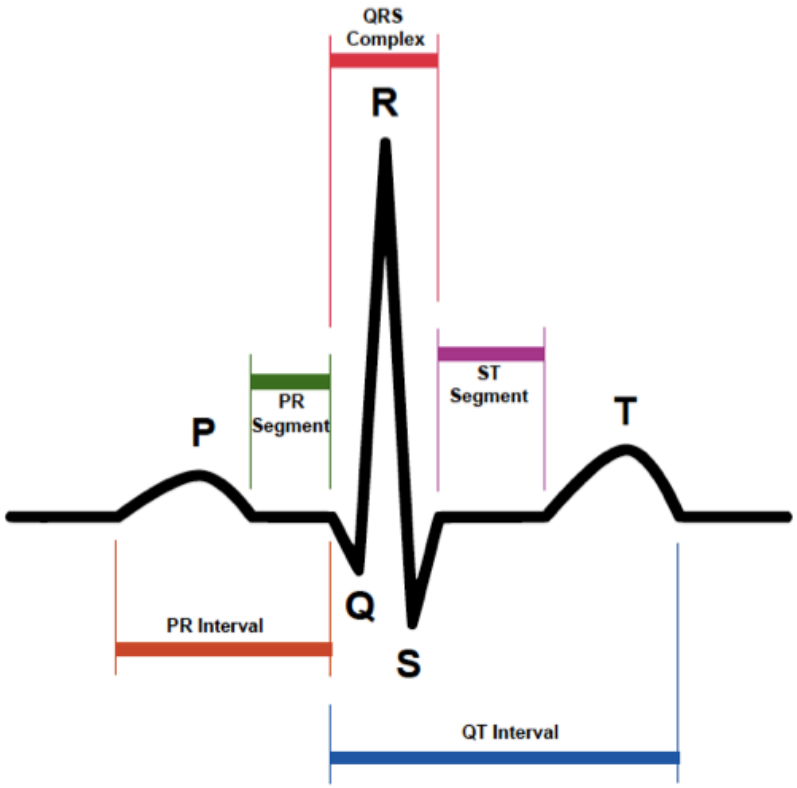


Figura 2.24: Ciclo cardíaco de la señal ECG [16].

Estos rasgos biométricos presentados anteriormente sirven para el reconocimiento

de una persona. En la literatura encontramos dos tipo de sistemas para el reconocimiento:

Verificación: También conocido como autenticación, la persona se identifica mediante alguna clave secreta, posteriormente se determina a través de una salida binaria si el individuo a verificar con la base de datos almacenadas coinciden, entonces se dice que la persona es quien dice ser, de lo contrario se considera una persona impostora. La comparación de la característica biométrica que la persona ingrese debe ser la misma con la que se registró en la base de datos.

Identificación: Este sistema es más complejo, ya que la persona no proporciona alguna identificación previa y por lo tanto hace comparaciones una contra todas de la base de datos. Esto arrojará como resultado a la persona que tenga una coincidencia similar con la entrada de la persona a identificarse. Cuando la persona a identificar no se asemeja a alguna característica de la base de datos, entonces se determina que la persona a identificar no está registrada en la base de datos. Otro caso que suele suceder es que al identificar una persona, como resultado nos arroje una lista de personas con las características biométricas similares.

Estos sistemas mencionados pasan por un proceso previo llamado **Sistema de matriculación** [17]. En este paso se realiza el proceso de inscripción del rasgo biométrico a emplear junto con la identificación de la persona. Se tiene que tener en consideración que la base de datos a generar tenga la mayor calidad del rasgo biométrico para poder hacer una identificación o autenticación correctamente. A continuación se muestra en la figura 2.25 los sistemas de reconocimientos antes mencionados.

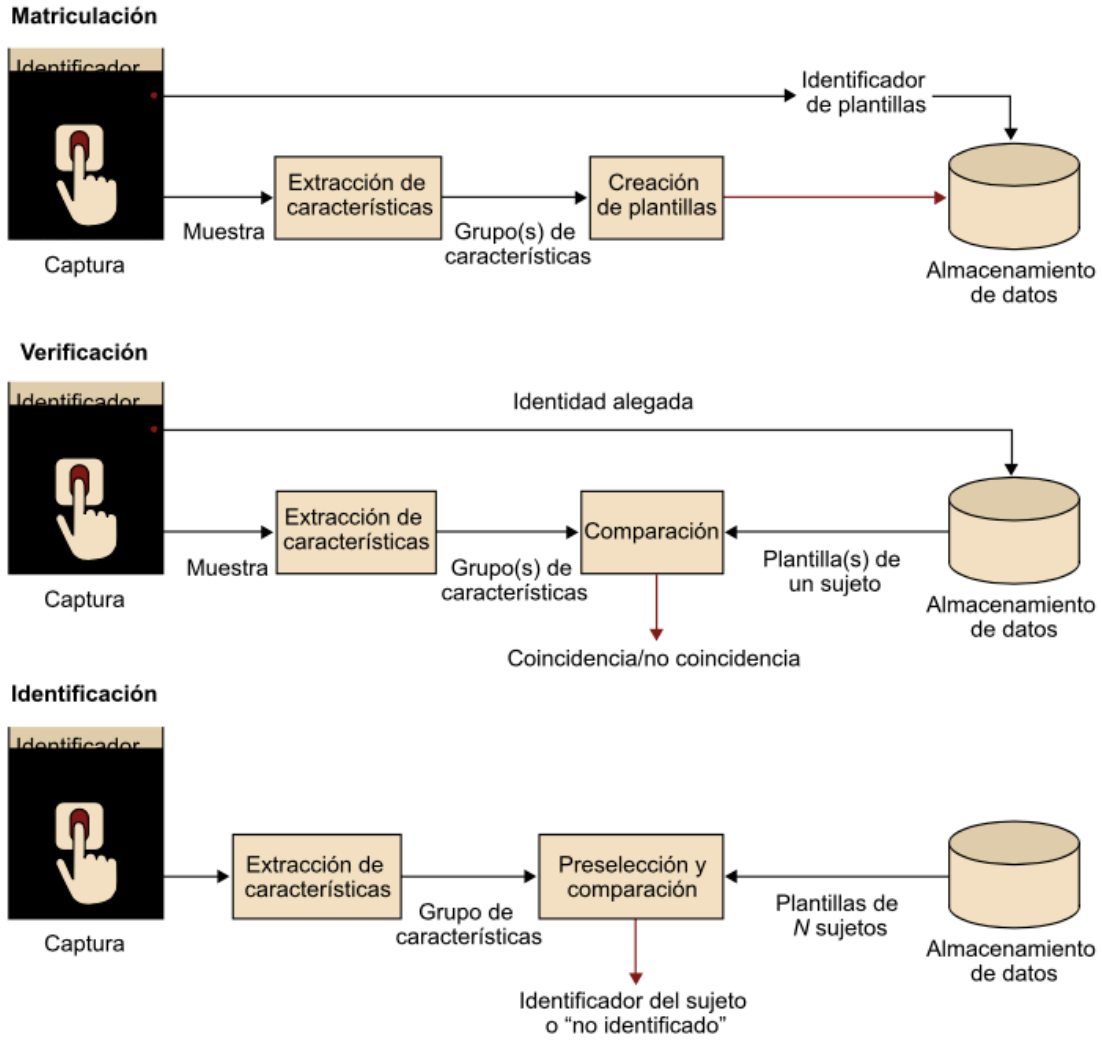


Figura 2.25: Sistemas de reconocimiento [17].

Al definir algún característica biométrica para la verificación o identificación se deben cumplir las siguientes atributos [17].

- Universalidad: Cada persona debe tener el rasgo biométrico a matricular.
- Particularidad: Es contraste cada individuo.
- Permanencia: Debe ser inalterable en el tiempo .
- Medible: Tiene que tener la capacidad de ser medido cuantitativamente.

- Rendimiento: Debe asegurar una buena precisión y robustez en diferentes escenarios ambientales.
- Aceptabilidad: El individuo debe comprometerse a usar ese rasgo biométrico para su identificación.
- No falsificable: Se debe avalar que su corrompimiento sea difícil de acertar.

2.4.1. Métricas de distancia

Su función es comparar dos vectores o listas de números que nos cuantifique el grado de similitud entre ellos; si esta relación es semejante entonces se puede tomar una decisión de clasificación para englobar vectores con características similares. Las distancias más utilizadas son la euclidiana y Dynamic Time Warping (DWT).

Distancia euclidiana: La distancia euclidiana se considera una de las mas importantes ya que es utilizada para estimar distancias en espacios físicos (bidimensionales o tridimensionales). Es una formula sencilla donde la distancia euclideana es la longitud de la hipotenusa de un triángulo, que se describe matemáticamente en la eq. 2.4.1.

$$d_{eucl}(A, B) = \sqrt{(X_s - X_t)(X_s - X_t)^T} \tag{2.4.1}$$

Dynamic Time Warping (DTW): Al igual que la distancia euclidiana es una de las más representativas, se encarga de medir la distancia de dos objetos numéricos deformados y variantes en el tiempo. Minimiza la distorsión en el tiempo y el desplazamiento para la detección de formas similares con diferentes fases [45]. Es utilizada en el reconocimiento de voz.

La figura 2.26 obtenida de [46] muestra un ejemplo básico para encontrar la similitud entres dos vectores similares pero variantes en el eje temporal.

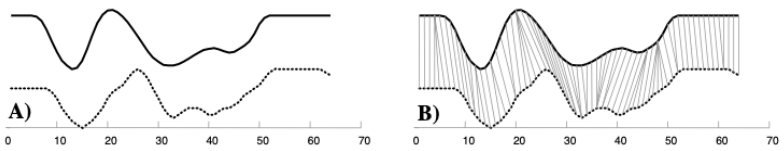


Figura 2.26: A) Se observa que las dos señales tiene la misma forma pero con diferentes desplazamiento. B) Se encuentra una alineación adecuada para la cuantificación distancias.

2.4.2. Matriz de confusión

En el área del aprendizaje automático se analizan problemas de clasificación donde se utiliza la matriz de confusión. Esta matriz es un diseño de tabla que nos indica el índice de desempeño analizando las clasificaciones correctas de las incorrectas respecto al número total de muestras. De esta forma se pueden comprobar las predicciones del modelo entrenado con los valores reales. Cada fila de la matriz representa las instancias en una clase real, mientras que cada columna representa las instancias en una clase predicha, o viceversa. En la tabla 2.1 se muestra los 4 posibles resultados de una matriz de confusión [47].

Valor Actual			
		Positivos	Negativos
Predicción de	Positivos	Verdadero positivo	Falso positivo
valores	Negativos	Falso negativo	Verdadero negativo

Tabla 2.1: Matriz de confusión

- Positivo (P): Casos positivos en la predicción.
- Negativo (N): Casos negativos en la predicción.
- Verdadero Positivo (TP): El resultado donde el clasificador predice correctamente la clase positiva.
- Verdadero Negativo (TN): El resultado donde el clasificador predice correctamente la clase negativa.
- Falso Positivo (FP): El resultado donde en que el clasificador predice incorrectamente la clase positiva cuando en realidad es negativa.
- Falso Negativo (FN): El resultado donde el clasificador predice incorrectamente la clase negativa cuando en realidad es positiva.

A partir de la matriz de confusión se puede determinar el rendimiento en el modo de autenticación e identificación.

Tasa de falsos positivos (FAR): Representa la probabilidad de que un sistema biométrico reconozca de manera incorrecta un vector de entrada con otro vector no coincidente de un conjunto de datos. En otras palabras, ¿Cuál es el porcentaje de clases positivas reales agrupadas de manera negativa (eq. 2.4.2)? [48].

$$FAR = \frac{FP}{FP + TN} \tag{2.4.2}$$

Tasa de falsos negativos (FRR): Representa la probabilidad de que un sistema biométrico no tenga la capacidad para hacer coincidir el vector de entrada con el vector coincidente de un conjunto de datos. En otras palabras, ¿Cuál es el porcentaje de clases negativas reales que fueron agrupadas de manera positiva (eq. 2.4.3)?[48].

$$FAR = \frac{FN}{TP + FN} \tag{2.4.3}$$

Tasa de error igual (EER): Evalúa la precisión del sistema biométrico para poder rechazar a un impostor. Este valor se puede encontrar a través del punto de intersección entre la tasa de falsos positivos y la tasa de falsos negativos a un concreto umbral de decisión, conocido también como umbral óptimo.

Precisión: Representa la probabilidad de la predicción del valor positivo de un sistema biométrico que fueron correctas. En otras palabras, ¿Cuál es el porcentaje de clases positivas fueron correctas (eq. 2.4.4)?

$$Precisión = \frac{TP}{TP + FP} \tag{2.4.4}$$

Exactitud: Representa la probabilidad de predicciones correctas (verdadero positivo y verdadero negativo) entre la suma de todos los casos de un sistema biométrico (eq. 2.4.5).

$$Exactitud = \frac{TP + TN}{TP + FP + FN + TN} \tag{2.4.5}$$

Sensibilidad: También conocida como tasa de verdaderos positivos (TPR), representa la probabilidad de que clases positivos de un sistema biométrico sean identificadas correctamente. En otras palabras, ¿Cuál es el porcentaje de clases positivas identificadas correctamente (eq. 2.4.6)?

$$Sensitividad = \frac{TP}{TP + FN} \tag{2.4.6}$$

Especificidad: También conocida como tasa de verdadero negativos (TNR), representa la probabilidad de que clases negativas de que un sistema biométrico sean identificadas correctamente (eq. 2.4.7).

$$Especificidad = \frac{TN}{FP + TN} \tag{2.4.7}$$

2.4.3. Curva ROC

La gráfica de curva característica de operación recibida (ROC) representa la tasa de verdaderos positivos (TPR) en el eje de las ordenadas y la tasa de falsos negativos (FPR o FAR) en el eje de las abscisas tal y como se muestra en la figura 2.27. Otra manera de representar la misma curva ROC es colocar la sensibilidad en el eje de las ordenadas y 1 − especificidad en el eje de las abscisas. De forma matemática se describe de la siguiente manera: $y = f(x)$, donde de acuerdo a la eq. 2.4.8 se obtiene el ROC [47].

$$ROC(c) = \begin{cases} Y = S(c) \\ X = 1 - E(c) \end{cases} \tag{2.4.8}$$

El área bajo una curva ROC es un método eficiente para discriminar el rendimiento de las pruebas de diagnósticos [49]. Se tienen en consideración 3 tipos de curvas habituales: curva perfecta, curva mala y curva aceptable.

Una prueba perfecta tiene un área bajo la curva ROC de 1. La sensibilidad y la especificidad coinciden con un valor de 1.

Si la prueba fuera mala, se traza una diagonal que va del punto (0,0 a 1,1), obteniendo un área bajo la curva ROC de 0.5. La sensibilidad es igual a la taza de falsos positivos y por lo tanto es una prueba sin capacidad para clasificar correctamente.

Las pruebas que se encuentren entre estas dos opciones se consideran como una área bajo la curva ROC aceptable. Entre mejor se acerque la curva a la esquina superior izquierda, se tendrá un mejor rendimiento de diagnóstico y por lo tanto una clasificación más perfecta.

En la literatura existe una gran variabilidad para clasificar la precisión de un sistema, en este caso se toma como referencia los estudios realizados por Centro Médico de la Universidad de Nebraska [50], donde se proponen rangos específicos.

- 1-0.90 = excelente (E).

- 0.90-0.80 = bueno (B).
- 0.80-0.70 = justo (C).
- 0.70-0.60 = pobre (D).
- 0.60-.50 = deficiente (F).

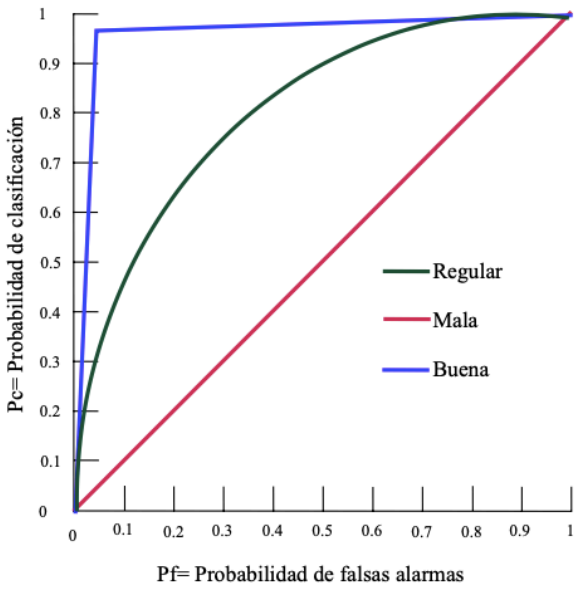


Figura 2.27: Curva ROC [18].

2.5. Biometría cancelable

La biometría cancelable consiste en una transformación propósito a un rasgo biométrico con la finalidad de poder salvaguardar la información propia de un individuo y que ésta no sea factible para los impostores. Se busca que estas distorsiones proporcionen una buen irreversibilidad de las plantillas biométricas [51].

La protección de plantillas biométricas se suelen clasificar como criptosistemas biométricos y deben satisfacer dos puntos importantes para la protección de la información biométrica de acuerdo a la norma (ISO/IEC FCD 24745) [52] que son: la irreversibilidad y desvinculación.

2.5.1. Medidas de evaluación para técnicas de protección de plantillas biométricas(BPT)

Se consideran 5 métricas de acuerdo al estándar ISO/IEC 24745:2022 para la evaluación de la biométrica cancelable y poder concluir que tan eficiente y segura puede ser.

1.- **Eficiencia:** Realiza una evaluación de la degradación de la técnica de protección. Esta evaluación esta basada en el número de revocaciones existentes como se muestra en la eq. 2.5.1.

$$ef = \frac{\frac{1}{\beta}\sum_{i=1}^{\beta} RR_{P,i}}{RR_o} \tag{2.5.1}$$

- RR: Tasa de reconocimiento.
- β : Número de revocaciones.
- $RR_{P,i}$: Tasa de reconocimiento protegida para la revocación i.
- RR_O: Tasa de reconocimiento del sistema original.

Si la eficiencia es igual a 1, entonces la técnica no degrada la tasa de reconocimiento. Si la eficiencia es menor a 1, indica que está degradando la tasa de reconocimiento y si la técnica es mayor 1 indica que aumenta la capacidad para distinguir y mejorar la tasa de reconocimiento.

2.- **Costo de almacenamiento:** Se basa en los recursos mínimos necesarios para almacenar la información protegida en bytes (eq. 2.5.2).

$$SC = \theta(SC_{PI}) \tag{2.5.2}$$

- θ: Número total de usuarios.
- SC_{PI}: Estimación de bits en punto fijo.

3.- **Capacidad de revocabilidad y renovabilidad:** Es la cantidad de plantillas protegidas generadas a partir de la técnica de protección.

- Limitada: La capacidad depende la técnica de protección.

- No limitada: La capacidad depende de la generación de números aleatorios.

4.- **No vinculabilidad:** Mide la relación lineal y no lineal entre las diferentes revocaciones (renovabilidad) de información biométrica generada por una técnica de protección, como se muestra en la eq. 2.5.3. Un valor en cero ($UNI = 0$) indica buena diversidad por lo tanto no hay vinculabilidad entre las versiones de plantillas protegidas. No deben tener correlación entre las diferentes revocaciones existentes.

$$UNI = \phi \sum_{i=1}^{\theta} \sum_{j=1}^{\beta-1} \sum_{k=j+1}^{\beta} \sum_{z=1}^{\omega} I(T_{i,j,z}, T_{i,k,z}) \tag{2.5.3}$$

- ω: Número de observaciones por cada usuario.
- $T_{i,j,z}$: Plantilla para el sujeto fijo i, revocación j y observación z.
- $T_{i,k,z}$: Plantilla para el sujeto fijo i, revocación k y observación z.

5.- **Irreversibilidad:** Muestra que tanta probabilidad existe para poder estimar la plantilla original (eq. 2.5.4). Tiene que ser difícil reconstruir la plantilla original a partir de los datos de referencia. Una irreversibilidad ideal es igual a 1 ($URI = 1$) lo que indica buena seguridad y privacidad de la información biométrica.

$$IRI = \frac{1}{\theta\beta\omega} \sum_{i=1}^{\theta} \sum_{j=1}^{\beta} \sum_{z=1}^{\omega} \left(\frac{h(T_{i,z,O}|T_{i,j,z})}{h(T_{i,z,O})} \right) \tag{2.5.4}$$

- $T_{i,z,O}$: Plantilla biométrica original para el sujeto i, revocación z y observación O.
- $T_{i,j,z}$: Plantilla biométrica para el sujeto i, revocación j y observación z.

2.6. Métricas NIST

Las métricas NIST nos permite examinar y comprobar la generación de números aleatorios que deben ser impredecibles en ausencia del conocimiento de las entradas [53]. Estos números son de gran ayuda para aplicaciones criptográficas. Las métricas NIST consta de 15 pruebas para analizar la aleatoriedad de secuencias binarias

(ciertos parámetros indican el mínimo requerido de bits) producidas por generadores de números aleatorios basados en hardware o software [53]. A continuación se describen las 15 pruebas:

1.- **Frecuency (Monobit):** Esta prueba nos indica si en una secuencia, la cantidad de 1's y 0's es el mismo, evaluando la proximidad de la fracción de unos a $\frac{1}{2}$. Un dato importante es que las pruebas siguientes depende del progreso de esta prueba.

2.- **Frequency Test within a Block:** Esta prueba determina la frecuencia de unos dentro de bloques de **M** bits. Se quiere comprobar que la frecuencia de unos en un bloque de **M** bits sea $\frac{M}{2}$.

3.- **Runs Test:** Esta prueba determina si la oscilación de ceros y unos es muy rápida o muy lenta de una determinada longitud **k** de bits.

4.- **Test for the Longest Run of Ones in a Block:** Esta prueba determina si una longitud de la serie de unos es consistente con la longitud de la serie mas larga de unos que se esperaría en una secuencia aleatoria. Las series están dentro de bloques de **M** bits.

5.- **Binary Matrix Rank Test:** Esta prueba determina la dependencia lineal entre subcadenas de una cierta longitud fija de bits.

6.- **Discrete Fourier Transform (Spectral) Test:** Esta prueba detecta patrones periódicos en una secuencia de bits que indican la desviación de la suposición de aleatoriedad.

7.- **Non-overlapping Template Matching Test:** Esta prueba se encarga de encontrar ocurrencias de un patrón no periódico en una ventana de **m** bits. Si no es localizado el patrón, la ventana se desliza una posición de bit, de lo contrario la ventana se restablece al bit posterior al patrón hallado y se reinicia la búsqueda.

8.- **Template Matching Test:** Esta prueba utiliza ventanas de **m** para hallar algún patrón especifico de **m** bits, al igual que la prueba anterior, si no es hallado el patrón, la ventana se desliza una posición de bit.

9.- **Maurer's "Universal Statistical" Test:** Esta prueba identifica si una secuencia de bits puede ser comprimida sin perder información. Si se puede comprimir entonces se concluye que la longitud de bits no es aleatoria.

10.- **Linear Complexity Test:** Esta prueba utiliza la longitud de un registro de desplazamiento de retroalimentación lineal (LFSR) para determinar si una secuencia de bits es bastante compleja para ser clasificada como aleatoria. Un LFSR demasiado largo implica que la secuencia de bits es altamente aleatoria.

11.- **Serial Test:** Esta prueba determina si el número de repeticiones de los 2^m patrones de **m** bits superpuesto es aproximadamente el mismo que se esperaría de una secuencia aleatoria. En otras palabras, busca los posibles patrones superpuestos de **m** bits en toda la secuencia.

12.- **Approximate Entropy Test:** Esta prueba compara la frecuencia de bloques superpuestos de dos longitudes consecutivas.

13.- **Cumulative Sums (Cusum) Test:** Esta prueba determina si la suma de una serie de bits parcial es muy grande o muy pequeña en comparación con el comportamiento esperado de esa suma acumulativa para las secuencias aleatorias. Si hay ceros en nuestra serie de bits, se cambia por -1.

14.- **Random Excursions Test:** Esta prueba se encarga de hallar si el número de vistas a una serie (longitud tomada al azar que empiezan y terminan en el mismo punto de origen) se desvía para poder clasificarla como una secuencia aleatoria.

15.- **Random Excursions Variant Test:** Esta prueba determina el número total de veces que se repite una secuencia de bits en concreto, se espera detectar desviaciones del número esperado de repeticiones a varias secuencias de bits.

Capítulo 3

Implementación de Arquitecturas PUFs

La implementación de las arquitecturas a diseñar son los Arbiter PUFs con enrutado manual y con enrutado automático. El diseño del Arbiter PUF va a determinar una respuesta por efecto aleatorio de las variaciones intrínsecas de fabricación. Para que la respuesta sea justa se deben cumplir dos condiciones:

1.- Los caminos de retardos entre cada switch de conmutación deben ser simétricos.

2.- El Arbiter debe ser justo, no favorece alguna entrada [54].

Por cada etapa (bloque de conmutación) del Arbiter PUF ocupa dos multiplexores y como árbitro (también llamado juez) un Flip Flop tipo D. Se analizaron también otros tipos de árbitros como son SR NAND, Latch D, Latch SR, Flip Flop Preset-Reset, sin embargo, el que mejor sincronización y estabilidad tuvo fue el Flip Flop D, ocupando la entrada D como el camino superior y la entrada CLK como el camino inferior. En la figura 3.1 se muestra el diseño y en la tabla 3.1 se observa su comportamiento.

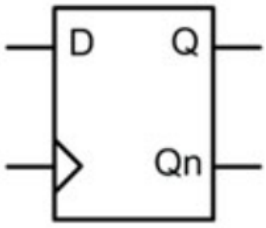


Figura 3.1: Esquemático Flip-Flop D [19].

CLK	D	Q	Qn
0	X	Last Q	Last Qn
1	X	Last Q	Last Qn
↑	0	0	1
↑	1	1	0

Tabla 3.1: Tabla de verdad del Flip-Flop D

La manera en que se hicieron las pruebas para el funcionamiento del Flip Flop fueron los siguientes casos:

- Que el camino superior(D) llegue primero que el inferior (CLK).
- Que el camino inferior(CLK) llegue primero que el camino superior(D).

Con los dos casos anteriores se asegura que el Flip Flop D mantenga ese mismo valor hasta que las entradas sean nuevamente cambiadas por un flanco de reloj, a diferencia de los otros modelos donde los valores de la salida cambian sin tomar en cuenta el flanco de reloj, por lo tanto no aseguran tener el valor correcto ya que la transición varia por las entradas y no por el reloj.

Una vez que se definió el tipo de arbitro a emplear, se implementaron los multiplexores de manera primitiva. Se utilizo la tarjeta de la familia Spartan 6 para el diseño de las arquitecturas. El manual [55] (recordar que por cada familia de Spartan son características únicas) es sumamente indispensable para la implementación de componentes primitivos, en este caso multiplexores y flip flops. La ventaja que se tiene al manipular los componentes primitivos es que se pueden colocar de manera precisa al lugar que se requiere, creando de esta manera diseños idénticos con los mismos componentes (es el caso que nos corresponde) y el comprimir un circuito en un lugar especifico de la tarjeta FPGA para reducir los tiempos de propagación. La primera arquitectura a diseñar fue el Arbiter PUF de enrutado manual. En este caso se diseñó con 3, 4, 5 y 6 etapas respectivamente para poder tener una visión y un rango más amplio al momento de evaluarlas y observar el comportamiento al aumentar o disminuir las etapas.

3.1. Funciones primitivas

Las funciones primitivas, que son propias de cada familia de FPGA son bloques que ya están definidos, como por ejemplo los Multiplexores, Flip-Flop, LUTs, Buffer, ROM, RAM etc y son utilizados por el compilador para crear cualquier diseño digital. La ventaja de estas funciones primitivas es que solo se agregan los componentes declarados, por ejemplo, suele suceder que cuando describimos un Multiplexor en VHDL o Verilog, el sintetizador agrega bloques extras como buffers y esto es lo que se quiere evitar.

En este trabajo se hizo uso de la manipulación de las funciones primitivas del multiplexor de 2 entradas y un Flip Flop D.

3.1.1. Colocación y enrutamiento de particiones.

En este apartado, lo que se realizó fue colocar el diseño en una región o SLICE (L, M o X) específico de la FPGA.

Se utilizaron atributos que son modificados desde el compilador (ISE Xilinx) y que permiten extraer información adicional sobre un objeto. Los atributos que se ocuparon son los siguientes:

- Attribute LOC: Permite colocar el diseño en un SLICE utilizando coordenadas. Ejemplo:
  attribute LOC of Componente_instanciado: label is SLICE_X1Y2
- Attribute BEL: Si existen varios componentes dentro de un SLICE, con este atributo se puede identificar a cada uno con una etiqueta. Ejemplo:
  attribute BEL of Componente_instanciado: label is D6LUT.
- Attribute DONT_TOUCH: Se asegura que el sintetizador ISE no mueva el componente o evita que se elimine cuando el sintetizador optimiza los diseños. Ejemplo:
  attribute DONT_TOUCH of Componente_instanciado: label is TRUE

Una vez explicado las funciones primitivas y los atributos, se describe a continuación el Multiplexor y Flip Flop D que son la base para el diseño de la arquitectura Arbiter PUF.

3.2. Descripción de Multiplexor primitivo

Un multiplexor de **n** entradas, es un circuito que permite pasar solo una de sus múltiples entradas a una única salida en función de una entrada del selector. El multiplexor tiene **n** lineas de selección, 2^n entradas y una salida.

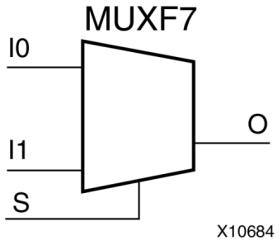


Figura 3.2: Multiplexor Primitivo [3].

La señal S es el selector, cuando tiene un valor bajo seleccionará I0 de lo contrario seleccionara I1 como se observa en la tabla 3.2.

Tabla de verdad			
Entradas			Salidas
S	**I0**	**I1**	**O**
0	I0	X	I0
1	X	I1	I1
X	0	0	0
X	1	1	1

Tabla 3.2: Tabla lógica del Multiplexor primitivo.

Además se puede encontrar una descripción breve de las entradas y salidas que tiene este componente primitivo en la tabla 3.3.

Puertos	**Dirección**	**Tamaño**	**Función**
O	Output	1	Salida del MUX
I0	Input	1	Entrada
I1	Input	1	Entrada
S	Input	1	Selector de entrada a MUX

Tabla 3.3: Descripción de puertos.

A continuación se presenta el código VHDL y la forma en que se instancia la celda del Mutiplexor

```
-- MUXF7: CLB MUX to tie two LUT6's together with general output
--        Spartan-6
-- Xilinx HDL Libraries Guide, version 14.7
MUXF7_inst : MUXF7
port map (
   O => O,     -- Output of MUX to general routing
   I0 => I0,   -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
   I1 => I1,   -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
   S => S      -- Input select to MUX
);
-- End of MUXF7_inst instantiation
```

Listing 3.1: Código VHDL para llamar al Multiplexor primitivo

Una vez teniendo el código para la función primitiva, se empieza declarando la librería **UNISIM** para hacer uso de todos los recursos primitivos que nos ofrece Xilinx. Posteriormente se declaran las entradas/salidas a utilizar y finalmente se instancia.

```
library UNISIM;
use UNISIM.VComponents.all;

entity Multiplexor_primitivo is
port(
C:  out std_logic;
A:  in std_logic;
B:  in std_logic;
Sel: in std_logic
);
end Multiplexor_primitivo;

architecture Behavioral of Multiplexor_primitivo is
attribute LOC : string;
attribute LOC of MUXF7_inst: label is "SLICE_X12Y3";

begin
MUXF7_inst : MUXF7
port map (
O => C, -- Output of MUX to general routing
I0 => A, -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
I1 => B, -- Input (tie to MUXF6 LO out or LUT6 O6 pin)
S => Sel -- Input select to MUX
);
end Behavioral;
```

Listing 3.2: Código VHDL instanciado con el Multiplexor primitivo

3.3. Descripción de Flip Flop D

Un Flip Flop D como el que se muestra en la figura 3.3 es un dispositivo utilizado como almacenamiento en la lógica secuencial. El almacenamiento de información se realiza a través de una transición de una señal de reloj de manera ascendente o descendente.

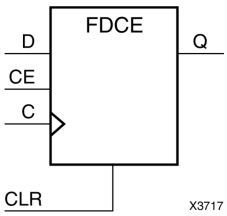


Figura 3.3: Flip Flop D Primitivo [3].

Cuando el reloj (CE) esta en alto entonces el dato de la entrada (D) de este diseño se transfiere a la salida (Q) durante la transición baja a alta del reloj (C). Cuando clear (CLR) esta en alto entonces las entradas son borradas y la salida (Q) se mantiene en bajo. Se puede ver en la tabla 3.4 su respectivo funcionamiento.

Tabla de verdad				
Entradas				**Salidas**
CLR	**CE**	**D**	**C**	**Q**
1	X	X	X	0
0	0	X	X	No cambio
0	1	D	↑	D

Tabla 3.4: Tabla lógica del Flip Flop D primitivo

Al igual que el multiplexor, también se encuentra su descripción VHDL de esta función primitiva y la forma para poder instanciarlo.

```
-- FDCE: Single Data Rate D Flip-Flop with Asynchronous Clear and
--       Clock Enable (posedge clk).
--       Spartan-6
-- Xilinx HDL Libraries Guide, version 14.7
FDCE_inst : FDCE
generic map (
   INIT => '0') -- Initial value of register ('0' or '1')
port map (
Q => Q,
C => C,
CE => CE,
CLR => CLR, -- Asynchronous clear input D=>D -- Data input
-- Data output
-- Clock input
-- Clock enable input
);
-- End of FDCE_inst instantiation
```

Listing 3.3: Código VHDL para llamar al Flip Flop D primitivo

Se siguen los mismos pasos mencionados anteriormente, utilizando siempre la librería **UNISIM**.

```
library ieee;
use ieee.std_logic_1164.all;
library UNISIM;
use UNISIM.VComponents.all;
entity flip_flop_d_general is
  port(
  RST : in std_logic;
  CLK : in std_logic;
  D : in std_logic;
  Q : out std_logic
  );
  end entity;
architecture ff of flip_flop_d_general is
attribute LOC : string;
attribute BEL : string; ----Es una Identificación
attribute LOC of FDCE_inst: label is "SLICE_X23Y17";
attribute BEL of FDCE_inst: label is "FFA";
begin
FDCE_inst : FDCE
generic map (
INIT => '0') -- Initial value of register (0 or 1)
port map (
Q => Q, -- Data output
C => CLK, -- Clock input
CE => '1', -- Clock enable input
CLR => RST, -- Asynchronous clear input
D => D -- Data input
);
end architecture ;
```

Listing 3.4: Código VHDL instanciado con el Flip Flop D primitivo

Explicado los dos componentes básicos e importantes para las arquitecturas se procede a diseñar. Los diseños fueron colocados en la parte inferior derecha usando la posición $X1Y0$ y $X1Y1$ donde arrojaron buenos resultados de fiabilidad, a comparación de los otras posiciones donde la fiabilidad era bastante baja, es decir, que los bits no eran estáticos, sino que tenían variaciones al reproducirlas varias veces, lo que hace que no sean tan confiables esas respuestas de la PUF.

Se duplicaron 8 veces las arquitecturas en paralelo tal y como se observa en la figura 3.4, con la finalidad de usar el protocolo Rs-232 para hacer la comunicación de la tarjeta FPGA a Matlab.

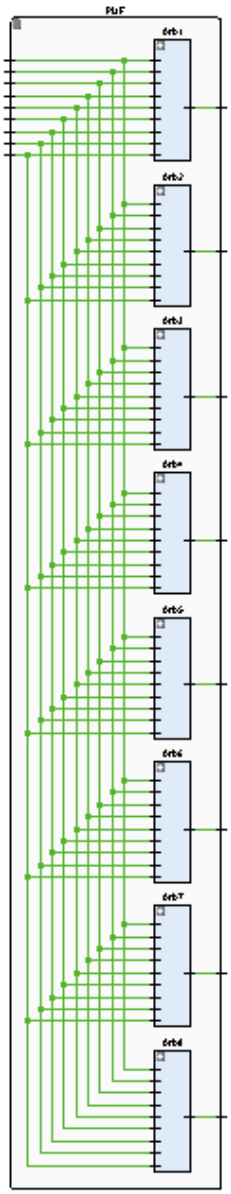


Figura 3.4: RTL de las 8 Arquitecturas Arbiter PUF.

Se implementó una máquina de estados (FSM) compuesta por RS-232, un Flip Flop de entrada y salida y la arquitectura Arbiter PUF. En la figura 3.5 se muestra los bloques antes mencionados.

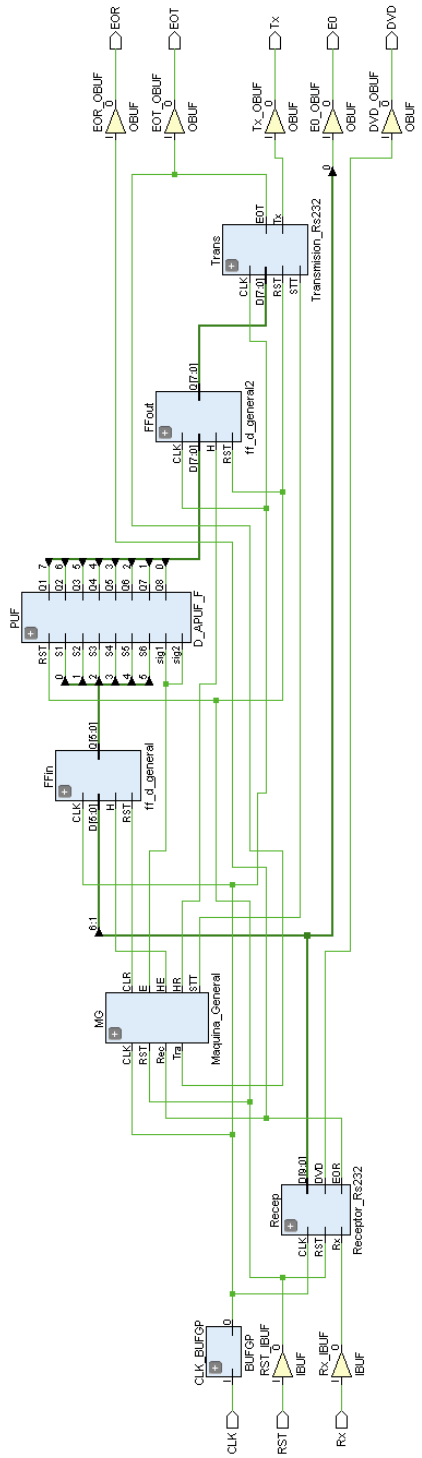


Figura 3.5: RTL del circuito final.

A continuación se presenta el funcionamiento de cada bloque de la máquina de

estados:

1.- Receptor: Se encarga de recibir la señal desde Matlab.

2.- Flip Flop Receptor: Guardara el dato de la señal del receptor para posteriormente mandarla al Arbiter PUF.

3.- APUF: Esperara que se active el Enable para poder hacer uso de las salidas del Flip Flip Receptor y tener una respuesta.

4.- Flip Flop Transmisor: Almacena la salida del Arbiter PUF y la manda al bloque del transmisor.

5.- Transmisor: Espera la señal para ser activado y devolver la salida Arbiter PUF al computador en Matlab.

En las figuras 3.6, 3.7, 3.8 y 3.9 se presentan las interconexiónes de los multiplexores del Arbiter PUF de 6, 5, 4 y 3 etapas.

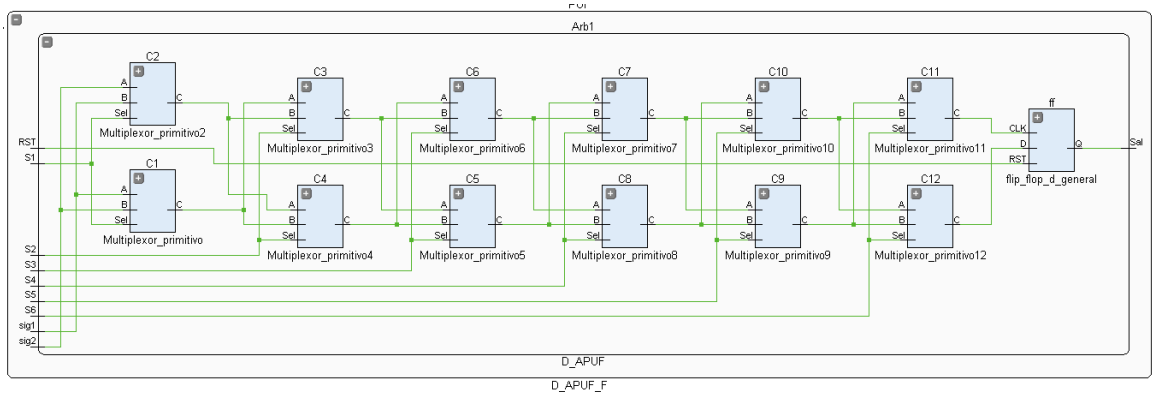


Figura 3.6: RTL Arbiter PUF de 6 etapas.

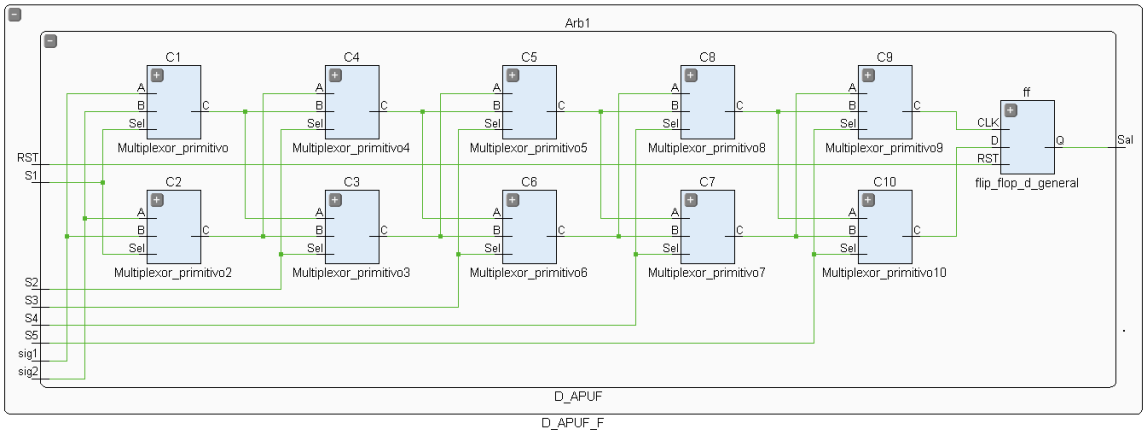


Figura 3.7: RTL Arbiter PUF de 5 etapas.

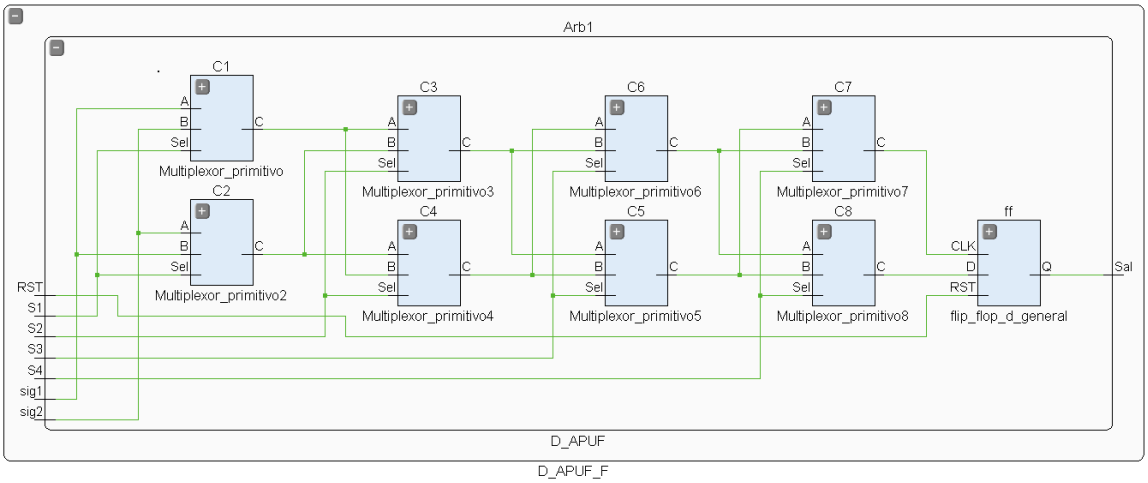


Figura 3.8: RTL Arbiter PUF de 4 etapas.

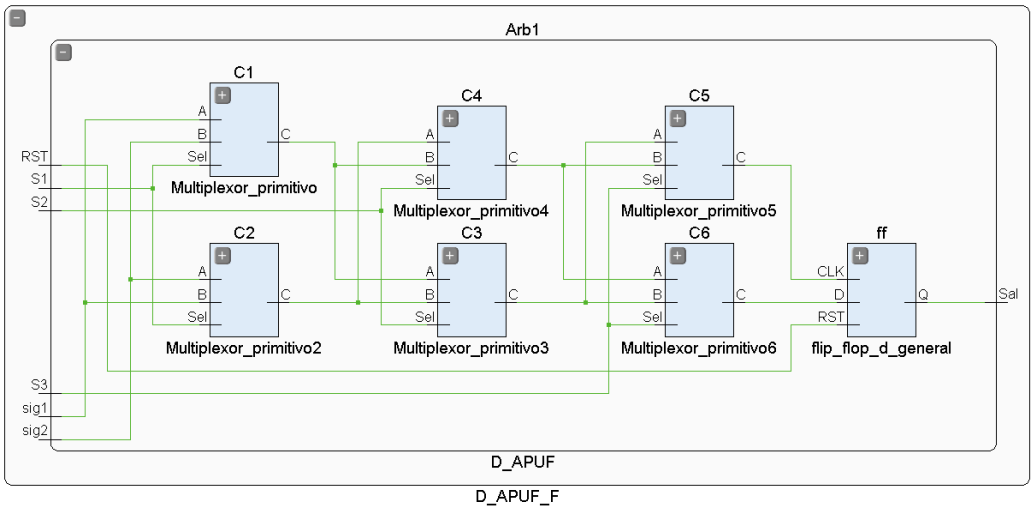


Figura 3.9: RTL Arbiter PUF de 3 etapas.

El RTL de cada Arbiter PUF tiene diferentes cantidades de etapa de retardo, con la finalidad de poder observar cuál de ellas tiene un mejor rendimiento con ayuda de las métricas de calidad PUF. También se puede observar la distribución de estos componentes primitivos en la tarjeta FPGA Spartan 6 en la figura 3.10.

Figura 3.10: Ubicación de los componentes primitivos en la FPGA.

En la figura 3.11 los multiplexores en la parte de abajo se consideran como el camino inferior y los multiplexores en la parte de arriba como el camino superior. Y en la parte final se encuentra el arbitro.

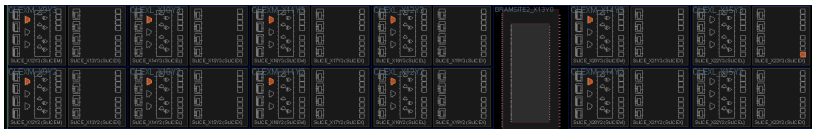


Figura 3.11: Colocación de Multiplexores y Flip Flop para Arbiter PUF de 5 etapas con enrutado manual.

Por otra parte también se diseñaron las arquitecturas PUF con enrutado automático, es decir, que el propio sintetizador del software ISE Xilinx colocara el diseño donde exista una mejor optimización, esto se ilustra en la figura 3.12. Una característica esencial que diferencia el enrutado manual y automático, es que en el manual los componentes son distinguidos con un color anaranjado mientras que el automático

son de color azul.

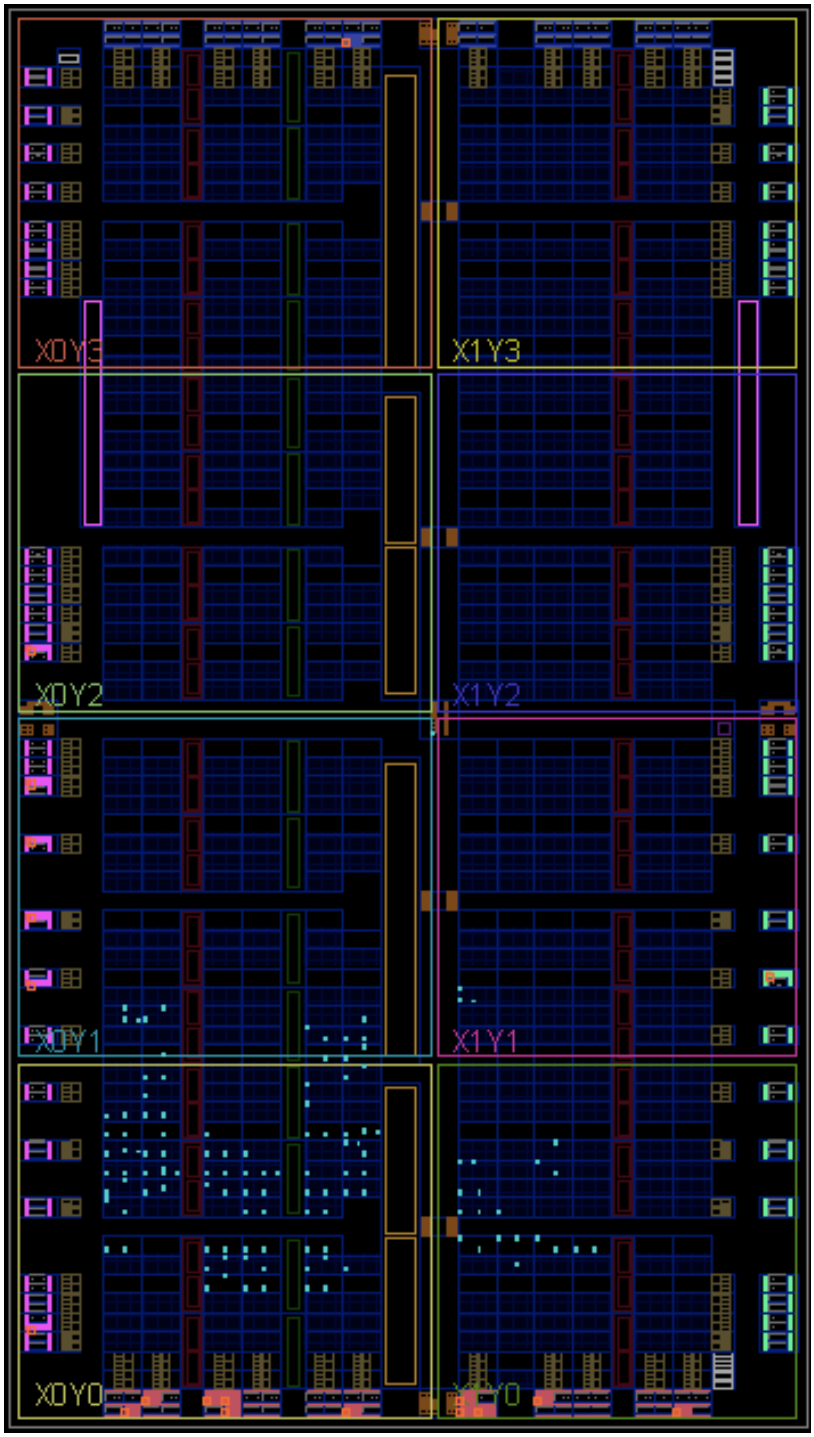


Figura 3.12: Colocación de Multiplexores y Flip Flop para Arbiter PUF de 5 etapas con enrutado automático.

En este enrutado todos los componentes están distribuido automáticamente, por lo cual no hay necesidad de mostrar las arquitecturas con sus respectivas etapas. El RTL sigue siendo el mismo que el del enrutado manual.

Hasta este punto se ha presentado el diseño de la arquitectura Arbiter PUF con enrutado automático y enrutado manual. Se omitirán los RTL de los demás diseños ya que el proceso es el mismo.

Capítulo 4
Resultados

4.1. Resultados de los PUFs

Antes de entrar a detalles con los resultados, hay un punto importante a mencionar acerca de la información suplementaria común o específica, ya que los resultados van a depender de estas dos opciones.

- **Común:** La información común se basa en crear una sola respuesta PUF para todos los usuarios. La ventaja es que el costo computacional es estable, es decir que el diseño del circuito no cambiará así aumente o disminuya una base de datos. Esta tesis se centró en la información común.
- **Específica:** Crea una respuesta PUF para cada uno de los usuarios. La ventaja es que elevará la tasa de reconocimiento pero el costo de almacenamiento y recursos será alto, ya que el costo computacional estará en función del número de usuarios. Cada vez que se agregue o quite un usuario se tendrá que modificar el diseño del circuito.

En este apartado se presentarán los resultados de las 6 arquitecturas PUFs. Cada diseño fue testeado 10, 50 y 100 veces con la finalidad de observar variabilidad de bits donde de acuerdo a la métrica de fiabilidad, la respuesta debe ser la misma ante **n** consultas. En las tablas 4.1 y 4.2 se puede observar cómo las métricas de calidad PUF son mejores en el enrutado automático, que son valores cercanos a los ideales. Tomando una recopilación de los datos de las tablas se puede concluir y observar que la arquitectura Arbiter PUF realizada con enrutado automático es la mejor para la generación de bits aleatorios, por lo cuál se tomará como referencia esta

arquitectura PUF para realizar la cancelación de la biometría ECG, además de que el costo computacional es menor que en las demás arquitecturas.

Arquitecturas PUF				
Arquitecturas	Fiabilidad	Uniformidad	Unicidad	Etapas
APUF	99.6438	51.4023	49.1513	6
	97.0625	53.4922	44.5565	5
	100	51.5625	50.4167	4
	100	43.75	16.0714	3
APUF XOR	98.9375	50.9297	48.3197	6
	99.375	46.8906	44.9597	5
	98.625	63.4531	43.2292	4
	97.25	48.7813	46.875	3
APUF FF	100	64.0625	41.25	6
	100	54.6875	39.5833	5
	99.75	62.4688	44.6429	4
	97	53.5	56.25	3
APUF XOR FF	97	38.6875	40.625	6
	97.375	41.7344	49.8958	5
	99.5	45.375	37.9464	4
	88.5	57.9375	39.5833	3
APUF LS	99.6094	52.7109	46.2721	6
	99.2813	53.0313	38.8889	5
	100	50	42.1371	4
	99.375	43.5938	46.6667	3
APUF LS FF	99.875	41.4375	46.875	6
	99.5625	45.3281	50.252	5
	94.75	42.2188	45.4167	4
	100	50	36.6071	3

Tabla 4.1: Evaluación de las Arquitecturas PUFs con enrutado automático.

Las arquitectas presentadas en la tabla 4.2 con enrutado manual son bastante deficientes en uniformidad y unicidad.

Arquitecturas PUF				
Arquitecturas	Fiabilidad	Uniformidad	Unicidad	Etapas
APUF	98.3438	79.7461	23.8219	6
	100	25.7813	27.3185	5
	100	16.4063	25.7292	4
	100	87.5	21.4286	3
APUF XOR	99.125	81.875	31.5625	6
	99.375	81.7134	20.7419	5
	100	70.8519	32.255	4
	99.25	80.6291	30.6863	3
APUF FF	100	15.1643	19.5784	6
	99.75	22.6518	28.8531	5
	98.375	31.5	28.6574	4
	98.625	13.5841	20.9642	3
APUF XOR FF	98.375	75.4318	21.9784	6
	99.6438	23.5647	25.6859	5
	99.75	18.0421	23.6547	4
	100	81.2353	17.1568	3
APUF LS	87.5675	13.2381	25.5876	6
	89.75	17.0975	19.6654	5
	87.0625	14.1768	25.6785	4
	84.5	21.9951	20.4563	3
APUF LS FF	100	78.9792	18.3197	6
	100	10.9378	18.3197	5
	88.5	83.5	21.25	4
	97	17.7813	26.875	3

Tabla 4.2: Evaluación de las Arquitecturas PUFs con enrutado manual.

4.1.1. Resultados de la biometría cancelable

Para realizar la biometría cancelable ya se cuenta con una base de datos procesadas tomados de [35]. Con la finalidad de poder comparar los resultados de ese trabajo con el presente. Se tiene un total de 50 sujetos, donde cada uno de ellos tiene alrededor de 684 señales ECG, a los cuales se les aplicó la extracción de características temporales y estadísticas que se en listan a continuación:

- Amplitud y diferencia de instantes entre ondas R.

- Actividad (Parámetro Hjorth).
- Movilidad (Parámetro Hjorth).
- Complejidad (Parámetro Hjorth).
- Curtosis.

Estas características se concatenan para formar un vector de longitud 7, por lo tanto para cada sujeto se tiene una matriz de tamaño 684x7 vectores de extracción de características.

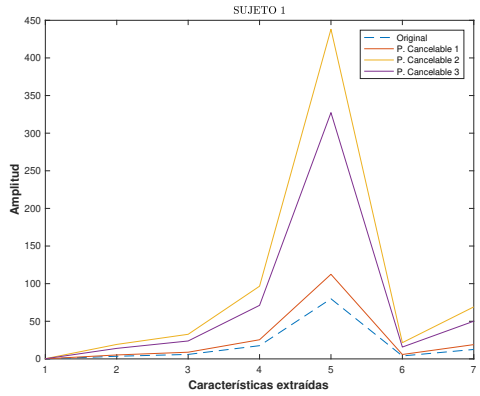
Tomando nuevamente de referencia [35] se tomó solo 6 vectores de extracción de características para realizar la cancelación de plantilla y tener una matriz de 6 por 7, dando un total de 42 datos. Posteriormente se necesita una base de datos de 42 bits de la PUF para forma la plantilla cancelable.

Cada arquitectura Arbiter PUF nos proporciona 1 bit de respuesta. Al contar con 4 etapas se tiene 16 posibles respuestas, entonces se propone colocar 8 Arbiter PUF en paralelo para tener un total de 128 bits. Como solo se necesitan 42 bits para la cancelación de plantilla, podemos tener hasta 3 plantillas disponibles, es decir hasta 3 revocaciones posibles.

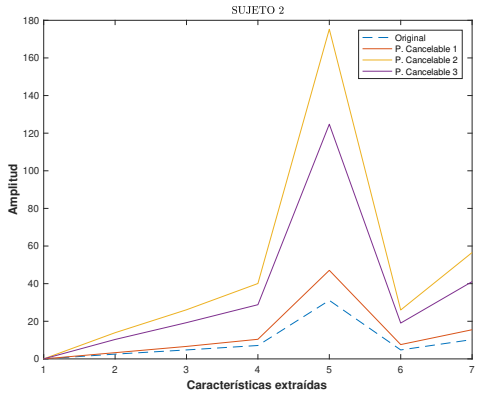
Para el método de cancelación se siguieron los siguientes pasos.

- Tomar la matriz de 6 por 7 de la biometría.
- Tomar una matriz de 6 por 7 de la PUF.
- Se multiplican celda por celda ambas matrices y se suman todos los datos de las 7 columnas.
- Se genera un vector de tamaño 1 por 7.
- Para generar una base de datos, se tomaron en cuenta 10 señales de la señal ECG para cada sujeto. Para obtener una matriz total de 500 por 7.

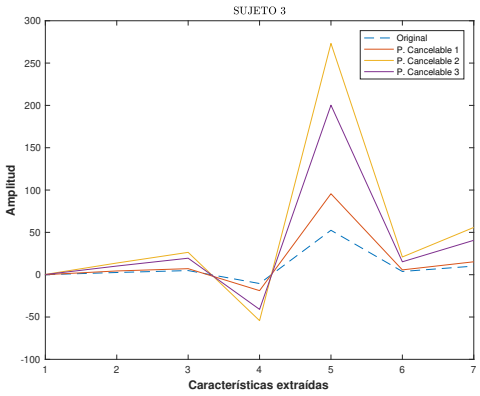
En las figuras 4.1 se puede visualizar una semejanza entre la plantilla original y las plantillas cancelables. Esta técnica fue evaluada con las métricas de seguridad BPT mencionadas anteriormente, donde se observó que la técnica anterior no es factible ya que los parámetros de no vinculabilidad y de irreversibilidad arroja valores alejados a los esperados como se muestra en la tabla 4.3.



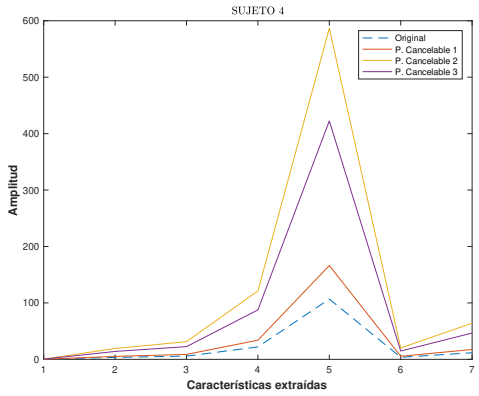
(a) Sujeto 1 observación 5.



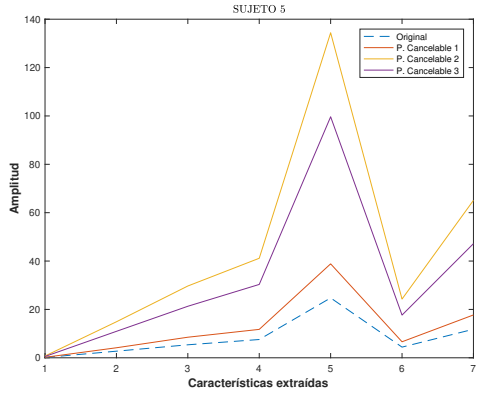
(b) Sujeto 2 observación 5.



(c) Sujeto 3 observación 5.



(d) Sujeto 4 observación 5.



(e) Sujeto 5 observación 5.

Figura 4.1: Utilizando la Técnica antes mencionada para 5 sujetos (elegidos aleatoriamente) se observa una similitud a la plantilla original.

BPT						
EF	**SC**	**RRC**	**UNI**	**IRI**	**ERR**	**AUC**
0.9330	1400	3	6.8188	-0.1917	16.64	91.46

Tabla 4.3: Evaluación de las métricas BPT.

Debido a lo obtenido, se planteó cambiar de técnica para la cancelación de plantilla basada en los siguientes pasos:

1.- Tomar la matriz de 6 por 7 del vector característica de la biometría y sumar las columnas para obtener un vector de 1 por 7.

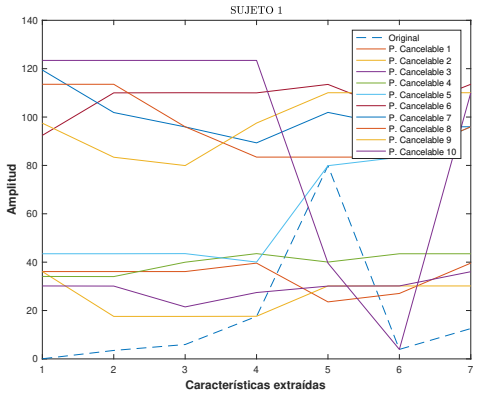
2.- Tomar una matriz de tamaño 7 por 7 de la PUF.

3.- Aplicar la siguiente operación $(P * B^T)^T$ para generar un vector de tamaño 1 por 7.

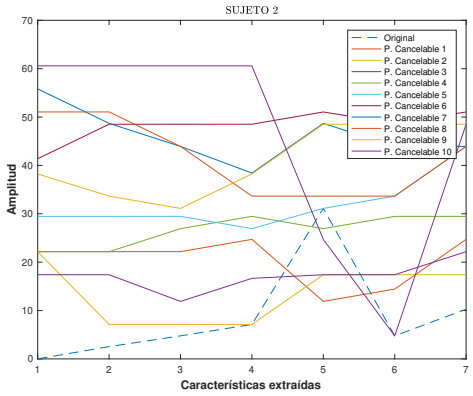
4.- Se toman 10 señales de la señal ECG para generar la base de datos de 500 por 7.

Para esta técnica se propone una arquitectura Arbiter PUF de 6 etapas para generar un total 64 bits. Duplicando esta Arbiter PUF 8 veces en paralelo, se obtiene un total de 512 bits por lo tanto se pueden crear 10 matrices posibles, es decir, tener 10 revocaciones disponibles. Para fines prácticos, también se realizaron éstas 10 revocaciones con la arquitectura Arbiter PUF manual.

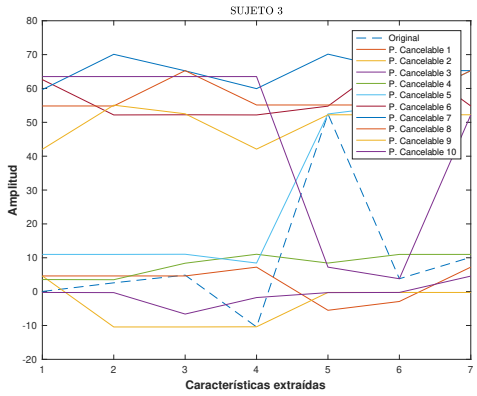
En las figuras 4.2 y 4.3 se puede observar como hay un gran diferencia con la primera técnica, en la cual no hay similitud entre la plantilla original y las plantillas cancelables que es el objetivo de esta tesis, salvaguardar la información de usuarios.



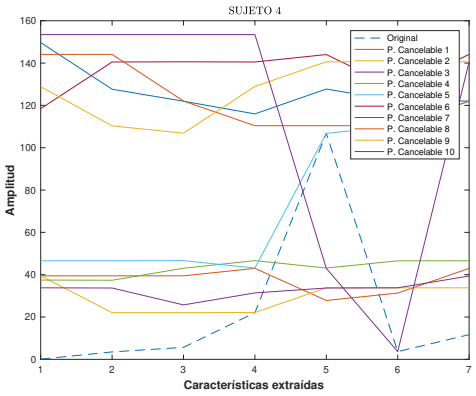
(a) Sujeto 1 observación 5.



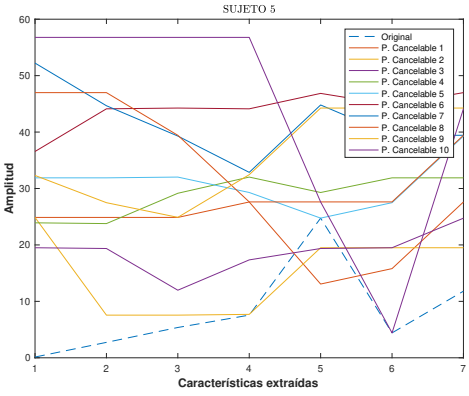
(b) Sujeto 2 observación 5.



(c) Sujeto 3 observación 5.

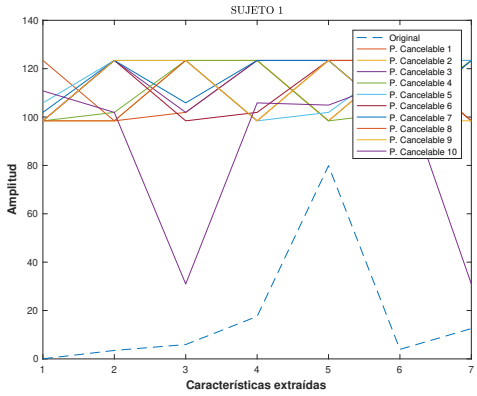


(d) Sujeto 4 observación 5.

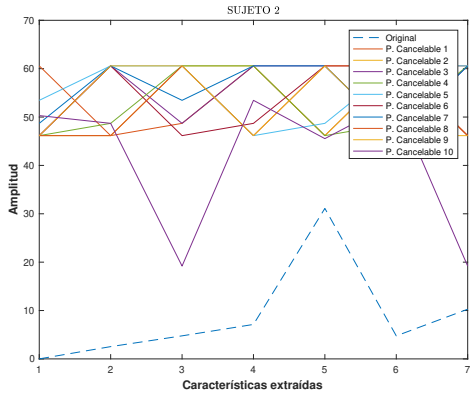


(e) Sujeto 5 observación 5.

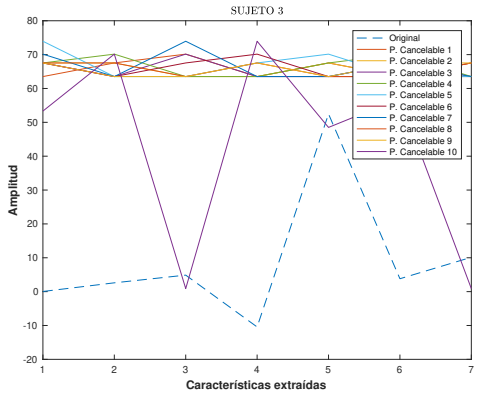
Figura 4.2: Utilizando la nueva técnica para los mismo 5 sujetos se observa una disimilitud a la plantilla original. También con la nueva arquitectura se obtienen 10 plantillas cancelables.



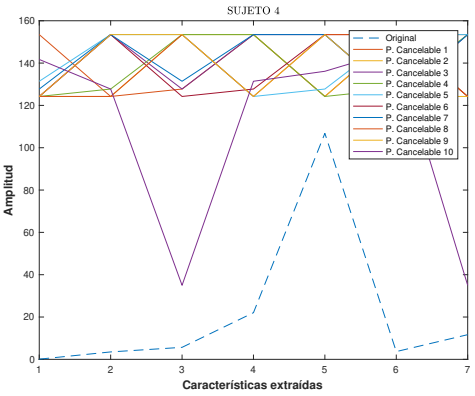
(a) Sujeto 1 observación 5.



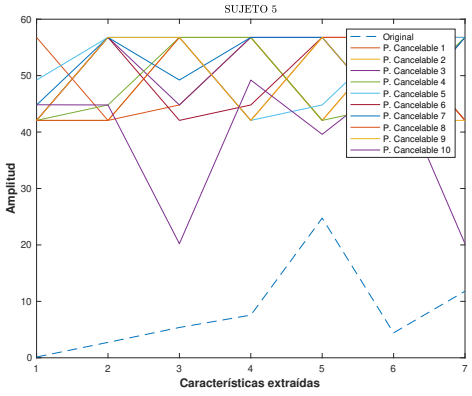
(b) Sujeto 2 observación 5.



(c) Sujeto 3 observación 5.



(d) Sujeto 4 observación 5.



(e) Sujeto 5 observación 5.

Figura 4.3: En este caso se empleó el Arbiter PUF manual para los 5 sujetos. Se observa una disimilitud a la plantilla original.

4.1.2. Clasificador

El método de clasificación que se utilizó fue en modo verificación, la cual se realizó con las distancias euclidiana y alineamiento temporal dinámico.

En las tablas 4.4, 4.5, 4.6 y 4.7 se presenta el error y el área bajo la curva del enrutado manual y automático. Como se mencionó anteriormente, el alineamiento temporal dinámico es para señales variantes en el tiempo, lo cual no es muy recomendable para este caso, ya que se están analizando señales estáticas. La diferencia del error y el área bajo la curva es mínima entre ambas distancias.

APUF Manual												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.1587	0.1716	0.1698	0.1716	0.1724	0.1707	0.1729	0.1707	0.1728	0.1728	0.1662	0.17115
AUC	0.9258	0.9118	0.9109	0.9118	0.9111	0.9121	0.9110	0.9121	0.9109	0.9109	0.9194	0.9122

Tabla 4.4: Error y Área bajo la curva de las 10 revocaciones con distancia Euclidiana.

APUF Manual												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.1613	0.1742	0.1680	0.1742	0.1764	0.1742	0.1676	0.1742	0.1680	0.1680	0.1662	0.1711
AUC	0.9258	0.9117	0.9107	0.9117	0.9111	0.9121	0.9109	0.9121	0.9107	0.9107	0.9194	0.9112

Tabla 4.5: Error y Área bajo la curva de las 10 revocaciones con distancia DWT.

Se observa en las tablas 4.5 y 4.6 que el error en el enrutado automático es menor que el manual, aunque aún así son valores aceptables en ambos casos.

APUF Automático												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.16	0.1453	0.1378	0.1364	0.1391	0.1573	0.1707	0.1729	0.1729	0.1724	0.1702	0.1575
AUC	0.9258	0.9403	0.9415	0.9461	0.9396	0.9247	0.9138	0.9060	0.9105	0.9136	0.9144	0.9258

Tabla 4.6: Error y Área bajo la curva de las 10 revocaciones con distancia Euclidiana.

APUF Automático												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.1613	0.1404	0.1360	0.1373	0.1458	0.1578	0.1733	0.1756	0.1738	0.1747	0.1667	0.15814
AUC	0.9258	0.9401	0.9415	0.9459	0.9395	0.9247	0.9137	0.9060	0.9104	0.9135	0.9143	0.92496

Tabla 4.7: Error y Área bajo la curva de las 10 revocaciones con distancia DWT.

Finalmente se evaluaron las 5 métricas BPT para el enrutado manual y automático como se aprecia en las tablas 4.8 y 4.9. Se estimó que el costo de almacenamiento para un formato de punto fijo de 32 bits es de 1400 bytes. Se tomó como referencia 32 bits, ya que es un rango adecuado para tener la señal lo más precisa posible. Lo anterior se obtiene de hacer la operación siguiente: SC= 50 Usuarios∗(7 Extracción de características ∗32 bits)= 11200 bits= 1400 bytes. Esta estimación se realizó para observar los recursos necesarios si se requiere implementar en un futuro.

BPT				
EF	**SC**	**RRC**	**UNI**	**IRI**
1.0022	1400	10	0.2378	0.9779

Tabla 4.8: Evaluación de las métricas BPT con arquitectura APUF automático.

BPT				
EF	**SC**	**RRC**	**UNI**	**IRI**
0.9852	1400	10	0.2080	0.9807

Tabla 4.9: Evaluación de las métricas BPT con arquitectura APUF manual.

Con los datos recabados en las tablas 4.1 y 4.2, se puede decir hasta el momento que la arquitectura manual ofrece una mejor no vinculabilidad e irreversabilidad a pesar de que en las gráficas se observó que no hay diferencias entre la plantilla original y las cancelables. Se recurre entonces ocupar las métricas NIST sobre los bits generados por las arquitecturas PUF, que indicarán si un conjunto de datos son altamente aleatorios o no. Solo se emplearon las 3 primeras métricas por limitantes de tiempo, obteniendo los siguientes resultados para la arquitectura Arbiter PUF automático y manual: si el valor P calculado es $< 0{,}01$ se concluye que la secuencia no es aleatoria de lo contrario se dice que la secuencia es aleatoria. En la tabla 4.10 se muestran los P-Value de los 3 test.

NIST			
	Test 1	**Test 2**	**Test 3**
Manual	$3{,}7692e^{-41}$	$3{,}1044e^{-35}$	$0{,}26312^{-20}$
Automático	0.288844	0.374515	0.169180

Tabla 4.10: P-Value de las métricas NIST.

Se aprecia que la mejor opción de generación de bits aleatorios es con las arquitecturas de enrutado automático.

Capítulo 5
Conclusiones

Las funciones físicas no clonables en FPGA son bastante eficientes para experimentar diferentes diseños de PUFs en las que se pueden aprovechar las diferencias de retardo de propagación generadas por variaciones intrínsecas de fabricación. Este nuevo uso ha permitido que las PUFs sean utilizadas en muchos procesos criptográficos.

Con los experimentos realizados del diseño de arquitectura Arbiter PUF se concluye que el enrutado automático es mucho mejor que el enrutado manual, esto se observó al implementar las métricas de calidad de la PUF y las 3 primeras métricas NIST. Esto es una gran ventaja porque el enrutado de manera manual toma más tiempo, ya que se tiene que crear un archivo extensión VHD por cada componente a utilizar debido a que cada uno de estos tiene una coordenada especifica, por lo tanto consume bastante tiempo ir revisando cada uno de esos pasos, a diferencia del enrutado automático que el propio software ISE Xilinx coloca cada componente de forma óptima al momento de sintetizar el código, ahorrando bastante tiempo.
Con la nueva técnica se mejora la eficiencia de las plantillas generando que la plantilla original no tenga alguna correlación con las plantillas cancelables, de tal modo que al comprometer una platilla cancelable, el intruso no sea capaz, a partir de esa plantilla, descifrar la plantilla original.

Las primeras conclusiones que se obtienen comparado con el trabajo [35] es que del 100 % de recursos (1280 componentes) que utilizó, en este trabajo de tesis solo se empleó el 8.125 % (96 componentes) para generar las plantillas cancelables; se generaron 2 plantillas extra; se tiene una mayor seguridad e irrevocabilidad, es decir que si la plantilla cancelable se ve involucrada de robo, ésta no es fácil de decodificar para llegar a la original; finalmente se tiene una mayor eficiencia y por lo tanto no degrada la tasa de reconocimiento.

El trabajo [35] obtiene una buena tasa de reconocimiento en el clasificador debido a que está utilizando dos fuentes de discriminación: la parte biométrica y un generador único para cada usuario. Debido a lo anterior, se reconoce más rápido a un usuario, ambas aportan información para reconocer a una persona pero aumenta el costo computacional.

Se ocuparon dos tarjetas de la familia Spartan 6 Amiba y Atlyx con la finalidad de observar que ambas conservaran la misma calidad de las PUF y que tuvieran respuestas diferentes, con lo que se afirmaría que tienen una identidad única.

Como trabajo futuro se espera continuar con las PUF con enrutado automático, ya que los resultados presentes en este trabajo arrojaron un mejor desempeño, es una gran ventaja ya que el enrutado manual es más laborioso en el sentido de tener cuidado con las coordenadas. Las arquitecturas PUF presentadas son basadas en retardo, trabajos anteriores han realizado arquitecturas basadas en oscilaciones por lo que queda implementar arquitecturas basadas en memoria. Se puede cambiar también el rasgo biométrico y esperar resultados de medidas BPT similares o mejores a los presentes en esta trabajo.

Apéndice

Anexos

A.1. Pruebas con la tarjeta FPGA Atlys

A continuación se anexan los resultados de una tarjeta FPGA Atyls de la familia Spartan 6 en las tablas A.1 y A.2. Se analizó este caso para poder observar que al menos las respuestas fueran diferentes entre las tarjetas y que además las métricas de calidad PUF estuvieron en el mismo rango de evaluación.

Arquitecturas PUF				
Arquitecturas	Fiabilidad	Uniformidad	Unicidad	Etapas
APUF	98.5313	50.5430	49.1310	6
	97.7500	53.8750	46.3911	5
	100	49.8438	50.9375	4
	100	59.3750	47.3214	3
APUF XOR	97.5938	46.5586	47.2284	6
	97.4375	39.4766	45.0353	5
	96.1250	48.1563	50.4167	4
	100	40.6250	40.1786	3
APUF FF	99.9688	56.2461	46.2302	6
	97.9375	45.5703	50.2016	5
	99.8750	54.7031	43.7500	4
	99.7500	56.1875	25	3
APUF XOR FF	99.8750	46.1094	45.1389	6
	99.5000	47.7188	42.4395	5
	99.3750	34.4219	38.3333	4
	87.7500	48.4375	34.6450	3
APUF LS	99.4688	52.2656	45.3373	6
	97.3125	54.3750	44.1532	5
	98.6250	48.4375	43.5417	4
	97.7500	46.8750	49.1071	3
APUF LS FF	98.3750	40.7188	46.9742	6
	98.1875	47.4844	44.2540	5
	99.8750	48.4063	43.7500	4
	99.7500	34.5000	33.9286	3

Tabla A.1: Evaluación de las Arquitecturas PUFs Automático.

Arquitecturas PUF				
Arquitecturas	Fiabilidad	Uniformidad	Unicidad	Etapas
APUF	99.0313	88.5430	17.9005	6
	100	26.9531	29.2087	5
	97.6250	25.2969	28.1250	4
	97.2500	84	20.5357	3
APUF XOR	92.500	78.6929	32.2550	6
	94.3750	17.0975	21.2500	5
	100	81.3195	19.3185	4
	91.6438	22.7134	17.8500	3
APUF FF	100	20.6815	33.9272	6
	100	25.8519	20.8567	5
	99.1250	23.1378	15.5876	4
	100	27.1378	20.7931	3
APUF XOR FF	81.5675	76.6518	20.6340	6
	78.1250	81.0421	13.9135	5
	84.500	83.500	18.3197	4
	89.7500	79.9951	16.8750	3
APUF LS	79.500	31.5	21.2500	6
	100	13.5841	21.8219	5
	99.250	10.9378	22.5684	4
	99.3750	15.7813	20.8750	3
APUF LS FF	92.500	50.9792	30.9784	6
	99.7500	70.8519	30.8212	5
	98.3750	79.0421	29.6518	4
	99.1250	67.1378	31.2556	3

Tabla A.2: Evaluación de las Arquitecturas PUFs Manual.

También se analizó el comportamiento del clasificador, obteniendo lo siguientes resultados de las tablas A.3 y A.4.

APUF Manual												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.1560	0.1510	0.1524	0.1613	0.1720	0.1509	0.1690	0.1777	0.1569	0.1667	0.1559	0.16138
AUC	0.9268	0.9220	0.9234	0.9190	0.9111	0.9225	0.9110	0.9221	0.9289	0.9121	0.9264	0.91985

Tabla A.3: Error y Área bajo la curva de las 10 revocaciones con distancia Euclidiana.

APUF Manual												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.1519	0.1677	0.1593	0.1623	0.1754	0.1755	0.1661	0.1542	0.1663	0.1621	0.1582	0.16471
AUC	0.9171	0.9267	0.9187	0.9279	0.9011	0.9021	0.9239	0.9151	0.9247	0.9267	0.9164	0.91833

Tabla A.4: Error y Área bajo la curva de las 10 revocaciones con distancia DWT.

APUF Automático												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.1511	0.1596	0.1381	0.1530	0.1377	0.1561	0.1651	0.1748	0.1769	0.1711	0.1628	0.15952
AUC	0.9241	0.9163	0.9459	0.9219	0.9464	0.9217	0.9189	0.9119	0.9129	0.9105	0.9191	0.92255

Tabla A.5: Error y Área bajo la curva de las 10 revocaciones con distancia Euclidiana.

APUF Automático												
	O	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**R9**	**R10**	**P**
EER	0.1509	0.1595	0.1366	0.1581	0.1372	0.1571	0.1649	0.1750	0.1759	0.1707	0.1622	0.15972
AUC	0.9255	0.9192	0.9476	0.9267	0.9441	0.9257	0.9177	0.9116	0.9120	0.9101	0.9185	0.92332

Tabla A.6: Error y Área bajo la curva de las 10 revocaciones con distancia DWT.

Bibliografía

[1] (2022) Tarjeta de desarrollo amiba2. [Online]. Available: https://intesc.mx/productos/tarjeta-de-desarrollo-amiba2/

[2] B. Halak, *Physically unclonable functions*. Springer, 2018.

[3] I. Xilinx, "Spartan-6 fpga configurable logic block," *Retrieved March*, vol. 20, p. 2019, 2010.

[4] S. Churiwala and I. Hyderabad, "Designing with xilinx® fpgas," in *Circuits &Systems*. Springer, 2017.

[5] P. Babu and E. Parthasarathy, "Reconfigurable fpga architectures: A survey and applications," *Journal of The Institution of Engineers (India): Series B*, vol. 102, pp. 143–156, 2021.

[6] R. Maes, P. Tuyls, and I. Verbauwhede, "Intrinsic pufs from flip-flops on reconfigurable devices," in *3rd Benelux workshop on information and system security (WISSec 2008)*, vol. 17, 2008, p. 2008.

[7] S. S. Kumar, J. Guajardo, R. Maes, G.-J. Schrijen, and P. Tuyls, "The butterfly puf protecting ip on every fpga," in *2008 IEEE International Workshop on Hardware-Oriented Security and Trust*. IEEE, 2008, pp. 67–70.

[8] P. Koeberl, U. Kocabas, and A.-R. Sadeghi, "Memristor pufs: a new generation of memory-based physically unclonable functions," in *2013 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2013, pp. 428–431.

[9] P. Simons, E. van der Sluis, and V. van der Leest, "Buskeeper pufs, a promising alternative to d flip-flop pufs," in *2012 IEEE International Symposium on Hardware-Oriented Security and Trust*. IEEE, 2012, pp. 7–12.

[10] R. Silwal, "Asynchronous physical unclonable function using fpga-based self-timed ring oscillator," Ph.D. dissertation, University of Toledo, 2013.

[11] R. Maes, "Physically unclonable functions: Properties," in *Physically Unclonable Functions*. Springer, 2013, pp. 49–80.

[12] Z. Cherif, "Modelling and characterization of physically unclonable functions," Ph.D. dissertation, Université Jean Monnet-Saint-Etienne, 2014.

[13] J. W. Lee, D. Lim, B. Gassend, G. E. Suh, M. Van Dijk, and S. Devadas, "A technique to build a secret key in integrated circuits for identification and authentication applications," in *2004 Symposium on VLSI Circuits. Digest of Technical Papers (IEEE Cat. No. 04CH37525)*. IEEE, 2004, pp. 176–179.

[14] M. Majzoobi, F. Koushanfar, and M. Potkonjak, "Lightweight secure pufs," in *2008 IEEE/ACM International Conference on Computer-Aided Design*. IEEE, 2008, pp. 670–673.

[15] R. Maes and I. Verbauwhede, "Physically unclonable functions: A study on the state of the art and future research directions," in *Towards hardware-intrinsic security*. Springer, 2010, pp. 3–37.

[16] C. E. Rose-Gómez and M. T. Serna-Encinas, "Procesamiento del electrocardiograma para la detección de cardiopatías," *Researchgate. Net, no. May*, pp. 3–6, 2015.

[17] F. Serratosa, "La biometría para la identificación de las personas," *Universitat Oberta de Catalunya*, pp. 8–20, 2008.

[18] A. Jiménez, "tesis de maestría," 2005.

[19] J. L. Brock, *Introduction to Logic Circuits & Logic Design with Verilog*. Spinger, 2019.

[20] M. Banday and A. H. Mir, "Cancellable biometric system based on linear combination of trigonometric functions with special application to forensic dental biometrics," *International Journal of Biometrics*, vol. 11, no. 4, pp. 342–371, 2019.

[21] A. K. Trivedi, D. M. Thounaojam, and S. Pal, "Non-invertible cancellable fingerprint template for fingerprint biometric," *Computers & Security*, vol. 90, p. 101690, 2020.

[22] M. Shahzad, S. Wang, G. Deng, and W. Yang, "Alignment-free cancelable fingerprint templates with dual protection," *Pattern Recognition*, vol. 111, p. 107735, 2021.

[23] S. Ibrahim, M. G. Egila, H. Shawkey, M. K. Elsaid, W. El-Shafai, F. E. Abd El-Samie *et al.*, "Hardware implementation of cancellable biometric systems," in *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2020, pp. 1145–1152.

[24] S. Narasimhan and M. Arunachalam, "Bio-puf-mac authenticated encryption for iris biometrics," *Computational Intelligence*, vol. 36, no. 3, pp. 1221–1241, 2020.

[25] A. A. Asaker, Z. F. Elsharkawy, S. Nassar, N. Ayad, O. Zahran, and F. E. Abd El-Samie, "A novel cancellable iris template generation based on salting approach," *Multimedia Tools and Applications*, vol. 80, no. 3, pp. 3703–3727, 2021.

[26] R. Arjona, M. Á. Prada-Delgado, J. Arcenegui, and I. Baturone, "A puf-and biometric-based lightweight hardware solution to increase security at sensor nodes," *Sensors*, vol. 18, no. 8, p. 2429, 2018.

[27] J. Kim and A. B. J. Teoh, "One-factor cancellable biometrics based on indexing-first-order hashing for fingerprint authentication," in *2018 24th International Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 3108–3113.

[28] M. J. Lee, Z. Jin, and A. B. J. Teoh, "One-factor cancellable scheme for fingerprint template protection: Extended feature vector (efv) hashing," in *2018 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2018, pp. 1–7.

[29] S. S. Sree and N. Radha, "Cancellable multimodal biometric user authentication system with fuzzy vault," in *2016 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE, 2016, pp. 1–6.

[30] A. D. Algarni, G. El Banby, S. Ismail, W. El-Shafai, F. E. A. El-Samie, and N. F Soliman, "Discrete transforms and matrix rotation based cancelable face

and fingerprint recognition for biometric security applications," *Entropy*, vol. 22, no. 12, p. 1361, 2020.

[31] R. Arjona, M. A. Prada-Delgado, I. Baturone, and A. Ross, "Securing minutia cylinder codes for fingerprints through physically unclonable functions: An exploratory study," in *2018 International Conference on Biometrics (ICB)*. IEEE, 2018, pp. 54–60.

[32] M. M. Eid and M. A. Mohamed, "A secure multimodal authentication system based on chaos cryptography and fuzzy fusion of iris and face," in *2017 Intl Conf on Advanced Control Circuits Systems (ACCS) Systems & 2017 Intl Conf on New Paradigms in Electronics & Information Technology (PEIT)*. IEEE, 2017, pp. 163–171.

[33] S. S. S. Priya, P. Karthigaikumar, N. S. Mangai, and R. Sandhya, "Efficient hardware implementation of aes algorithm using bio metric key," *International Journal of Information and Communication Technology*, vol. 7, no. 4-5, pp. 437–454, 2015.

[34] S. Nazari, M.-S. Moin, and H. R. Kanan, "Cancelable face using chaos permutation," in *7'th International Symposium on Telecommunications (IST'2014)*. IEEE, 2014, pp. 925–928.

[35] D. K. Torres, "Biometría cancelable basada en funciones físicas inclonables en fpga para señales ecg," 2020.

[36] W. Adi and A. Mars, "Physical and mechatronic security, technologies and future trends for vehicular environment," *arXiv preprint arXiv:1805.07570*, 2018.

[37] S. P. Balasubramanian, "An improved public unclonable function design for xilinx fpgas for hardware security applications," 2018.

[38] A. Mills, S. Vyas, M. Patterson, C. Sabotta, P. Jones, and J. Zambreno, "Design and evaluation of a delay-based fpga physically unclonable function," in *2012 IEEE 30th International Conference on Computer Design (ICCD)*. IEEE, 2012, pp. 143–146.

[39] T. Idriss, H. Idriss, and M. Bayoumi, "A puf-based paradigm for iot security," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE, 2016, pp. 700–705.

[40] K. T. Mursi and Y. Zhuang, "Experimental examination of component-differentially-challenged xor puf circuits," in *Journal of Physics: Conference Series*, vol. 1729, no. 1. IOP Publishing, 2021, p. 012006.

[41] P. Bulens, F.-X. Standaert, and J.-J. Quisquater, "How to strongly link data and its medium: the paper case," *IET Information Security*, vol. 4, no. 3, pp. 125–136, 2010.

[42] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded systems design with FPGAs*. Springer, 2013, pp. 245–267.

[43] H. Basel, "Physically unclonable functions—from basic design principles to advanced hardware security applications. doi: 10.1007."

[44] G. P. Garza, "El electrocardiograma y su tecnología," *Revista de Divulgación Médico Científica AVANCES*, vol. 8, no. 24, pp. 27–31, 2011.

[45] P. Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, no. 1-23, p. 40, 2008.

[46] E. J. Keogh and M. J. Pazzani, "Derivative dynamic time warping," in *Proceedings of the 2001 SIAM international conference on data mining*. SIAM, 2001, pp. 1–11.

[47] A. R. d. Valle Benavides, "Curvas roc (receiver-operating-characteristic) y sus aplicaciones," 2017.

[48] J. C. B. Romero, "Sistema biométrico basado en ecg e implementación en un sistema embebido con vhdl," 2020.

[49] S. H. Park, J. M. Goo, and C.-H. Jo, "Receiver operating characteristic (roc) curve: practical review for radiologists," *Korean journal of radiology*, vol. 5, no. 1, pp. 11–18, 2004.

[50] M. Thomas G. Tape, "Interpreting diagnostic tests."

[51] A. Cavoukian and A. Stoianov, "Encyclopedia of biometrics," in *Biometric Encryption*. Springer, 2009.

[52] C. Rathgeb and A. Uhl, "A survey on biometric cryptosystems and cancelable biometrics," *EURASIP journal on information security*, vol. 2011, no. 1, pp. 1–25, 2011.

[53] L. E. Bassham III, A. L. Rukhin, J. Soto, J. R. Nechvatal, M. E. Smid, E. B. Barker, S. D. Leigh, M. Levenson, M. Vangel, D. L. Banks *et al.*, *Sp 800-22 rev. 1a. a statistical test suite for random and pseudorandom number generators for cryptographic applications*. National Institute of Standards & Technology, 2010.

[54] L. Lin, D. Holcomb, D. K. Krishnappa, P. Shabadi, and W. Burleson, "Low-power sub-threshold design of secure physical unclonable functions," in *Proceedings of the 16th ACM/IEEE international symposium on Low power electronics and design*, 2010, pp. 43–48.

[55] X. Spartan, "Libraries guide for hdl designs (2009)," 6.