UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES INSTITUTO DE POSTGRADO

TITULO DEL TRABAJO DE TITULACIÓN

EVALUACIÓN EXPERIMENTAL DE ALGORITMO DE ENCRIPTACIÓN AES EN HARDWARE FPGA PARA LA OPTIMIZACIÓN DE LA SEGURIDAD EN COMUNICACIONES AERONÁUTICAS DESARROLLADAS MEDIANTE EL USO DE SISTEMAS EMBEBIDOS EN AERONAVES DE ALA ROTATORIA

AUTOR

Andrade Reyes, Marcos Aurelio

TRABAJO DE TITULACIÓN

Previo a la obtención del grado académico en

MAGISTER EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR

Sánchez Aquino, José Miguel

Santa Elena, Ecuador

Año 2024

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES INSTITUTO DE POSTGRADO

TRIBUNAL DE SUSTENTACIÓN

Firmado electrónicamente por:
ALICIA GERMANIA
ANDRADE VERA

Ing. ALICIA ANDRADE VERA, Mgtr.
COORDINADORA DEL PROGRAMA

Firmado electrónicamente por:
JOSE MIGUEL SANCHEZ
AQUINO

Ing. JOSÉ SÁNCHEZ AQUINO, Mgtr.
TUTOR

Firmado electrónicamente por:
LUIS ENRIQUE
CHUQUIMARCA JIMENEZ

Ing. LUIS CHUQUIMARCA JIMÉNEZ, Mgtr.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
SAMUEL BALDOMERO
BUSTOS GAIBOR

Ing. SAMUEL BUSTOS GAIBOR, Mgtr.
DOCENTE ESPECIALISTA

Firmado electrónicamente por:
MARIA MARGARITA
RIVERA GONZALEZ

Abg. María Rivera, MSc.
SECRETARIO GENERAL UPSE

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES INSTITUTO DE POSTGRADO

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por Marcos Aurelio Andrade Reyes, como requerimiento para la obtención del título de Magister en Electrónica y Automatización.

TUTOR

Firmado electrónicamente por:
JOSE MIGUEL SANCHEZ AQUINO

José Sánchez Aquino

Santa Elena, 10 de junio de 2024

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES INSTITUTO DE POSTGRADO

DECLARACIÓN DE RESPONSABILIDAD

Yo, **MARCOS AURELIO ANDRADE REYES**

DECLARO QUE:

El trabajo de Titulación; **Evaluación experimental de algoritmo de encriptación AES en hardware FPGA para la optimización de la seguridad en comunicaciones aeronáuticas desarrolladas mediante el uso de sistemas embebidos en aeronaves de ala rotatoria**, previo a la obtención del título en Magister en Electrónica y Automatización, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 10 de junio de 2024

EL AUTOR

Firmado electrónicamente por:
MARCOS AURELIO
ANDRADE REYES

Ing. Marcos Andrade Reyes

UNIVERSIDAD ESTATAL PENÍNSULA DE SANTA ELENA

FACULTAD DE CIENCIAS DE LA INGENIERÍA INSTITUTO DE POSTGRADO

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado **Evaluación experimental de algoritmo de encriptación AES en hardware FPGA para la optimización de la seguridad en comunicaciones aeronáuticas desarrolladas mediante el uso de sistemas embebidos en aeronaves de ala rotatoria**, presentado por el estudiante, **Ing. Marcos Aurelio Andrade Reyes** fue enviado al Sistema Antiplagio COMPILATIO MAGISTER, presentando un porcentaje de similitud correspondiente al 5%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.

CERTIFICADO DE ANÁLISIS
magister

Andrade Marcos EyA

5%
Textos sospechosos

5% Similitudes
0% similitudes entre comillas
< 1% entre las fuentes mencionadas
< 1% Idiomas no reconocidos

Nombre del documento: Andrade Marcos EyA.docx
ID del documento: 6af06b5ce9d0d9199853c8c01617adb0a8c43fa5
Tamaño del documento original: 5,28 MB

Depositante: JOSE MIGUEL SANCHEZ AQUINO
Fecha de depósito: 26/6/2024
Tipo de carga: interface
fecha de fin de análisis: 26/6/2024

Número de palabras: 14.912
Número de caracteres: 99.808

TUTOR

Firmado electrónicamente por:
JOSE MIGUEL SANCHEZ AQUINO

José Sánchez Aquino

UNIVERSIDAD ESTATAL PENÍNSULA

DE SANTA ELENA

FACULTAD DE SISTEMAS Y TELECOMUNICACIONES INSTITUTO DE POSTGRADO

AUTORIZACIÓN

Yo, **MARCOS AURELIO ANDRADE REYES**

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales de artículo profesional de alto nivel con fines de difusión pública, además apruebo la reproducción de este artículo académico dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, 10 de junio de 2024

EL AUTOR

Firmado electrónicamente por:
MARCOS AURELIO
ANDRADE REYES

Ing. Marcos Andrade Reyes

AGRADECIMIENTO

Al Divino Creador por su infinita bondad, a las instituciones que hicieron de este viaje una constante aventura; rica en aprendizaje, en ellas a la Dirección de Innovación y Desarrollo de la Armada del Ecuador por brindarme el soporte para cumplir esta meta y a la UPSE por abrir sus puertas y trazar el camino, infinitas gracias.

Marcos Andrade Reyes

DEDICATORIA

A mi familia, quienes con su paciencia, respaldo y comprensión permitieron que pueda llegar a la meta, a mis Erickitas les dedico este esfuerzo que también es suyo.

Marcos Andrade Reyes

ÍNDICE GENERAL

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

RESUMEN

La evaluación de un algoritmo de encriptación AES mediante VHDL propone un ejercicio de investigación tendiente a validar código que pueda evidenciar el rendimiento en el uso de recursos en diversos circuitos integrados FPGA de las familias Cyclone IV, Cyclone V y Artix7. En ello supone el aprendizaje de la norma FIPS197, pero desde el punto de vista de la codificación en VHDL, este trabajo aborda un análisis interesante respecto del uso de hardware FPGA como barrera primaria para la protección de datos digitales en sistemas embebidos. El análisis de los reportes de síntesis e implementación de las herramientas de diseño digital Vivado y Quartus II hacen de este trabajo de investigación un referente en la validación de tecnologías FPGA. Los resultados obtenidos nos muestran una perspectiva interesante respecto de las razones para la adopción de un circuito integrado específico para la implementación soluciones electrónicas.

Palabras claves: AES128, VHDL, FPGA

ABSTRACT

The evaluation of an AES encryption algorithm using VHDL proposes a research exercise aimed at validating code that can demonstrate the performance in the use of resources in various FPGA integrated circuits of the Cyclone IV, Cyclone V and Artix7 families. This involves learning the FIPS197 standard, but from the point of view of VHDL coding, this work addresses an interesting analysis regarding the use of FPGA hardware as a primary barrier for the protection of digital data in embedded systems. The analysis of the synthesis and implementation reports of the digital design tools Vivado and Quartus II make this research work a reference in the validation of FPGA technologies. The results obtained show us an interesting perspective regarding the reasons for the adoption of a specific integrated circuit for the implementation of electronic solutions.

Keywords: AES128, VHDL, FPGA

INTRODUCCIÓN

El algoritmo de encriptación AES está amparado en el estándar federal para el procesamiento de información FIPS197, el cual describe el procedimiento para la encriptación de datos sensibles de forma simétrica; es decir, requiere de una clave que debe conocer el remitente y el destinatario, por tanto, la cadena de custodia para ésta debe ser establecida bajo niveles de confianza. En ese contexto, el desarrollo de algoritmos de encriptación a nivel mundial tiene a las universidades y las Fuerzas Armadas inmersas en la investigación y desarrollo de tecnologías de seguridad; sin embargo, por las características de la naturaleza de los datos que se intentan proteger, la divulgación de información inherente a estos avances es extremadamente limitada.

Con este estudio se pretende territorializar el modelamiento de algoritmos de encriptación AES sobre hardware FPGA como aporte a la seguridad en sistemas de transmisión de datos mediante sistemas embebidos, en este ámbito se conoce que universidades de prestigio en todo el mundo han trabajado algoritmos de encriptación para hardware FPGA centrándose en el mejoramiento de la seguridad, eficiencia y resistencia a ataques. En algunas regiones existen colaboraciones público-privadas entre instituciones gubernamentales, instituciones de educación superior y empresa privada para desarrollar encriptaciones robustas dados los escenarios de inteligencia financiera, militar, etc.

El eje central del desarrollo de esta investigación de carácter experimental está encaminado a evaluar un algoritmo de encriptación AES sobre diferentes circuitos integrados FPGA para evaluar su rendimiento. La evaluación exhaustiva considerará aspectos de seguridad, eficiencia de recursos y rendimiento en diferentes entornos FPGA, con el objetivo de

proporcionar una comparativa detallada y fundamentada entre las plataformas de síntesis más utilizadas en la industria.

A nivel del continente americano tanto en el norte como en el cono sur se han desarrollado proyectos de investigación sobre seguridad y criptografía, aunque la evidencia de tales hallazgos es también por su naturaleza poco publicado.

En Ecuador estos desarrollos son aún incipientes debido a la reducida sinergia entre la academia, las instituciones públicas y la empresa privada. Esto ha conllevado a que los niveles de seguridad de la información digital se vean comprometidos y con ello se generen escenarios de fraude financiero, tributario o de seguridad física según el contexto y escenario de operación. No existe evidencia tangible de soluciones desarrolladas a nivel local que aporten al robustecimiento de la seguridad de la información más allá de la utilización de equipos de marcas conocidas que aportan niveles de seguridad, pero no obstante este tema vas más allá debido a la confianza que el proveedor puede generar en la institución o empresa que necesita proteger su información. Se espera que este estudio contribuya al avance en el desarrollo de soluciones de seguridad en sistemas embebidos, ofreciendo información valiosa sobre la implementación de AES en hardware FPGA y su impacto en la protección de datos digitales en entornos sensibles.

En este contexto el escenario sobre el cual se desarrollará esta investigación aplicada propenderá a sentar las bases para el desarrollo de algoritmos de encriptación basados en hardware debido a que aportan con seguridad de primera línea al diferenciarse de los algoritmos de encriptación basados en software ya que no pueden ser fácilmente copiados y sometidos a ingeniería inversa por cuanto su implementación se da en un dispositivo sintetizable (FPGA). Este tema se presenta como una necesidad urgente para el aseguramiento de la información en operaciones donde la evidencia debe ser protegida como

por ejemplo la operación de una aeronave que estando fuera de los límites del área de cobertura de radio VHF/UHF deba reportar información sensible (actividades ilícitas) y necesite hacerlo mediante enlace satelital, este estudio propende a aportar con las bases para el desarrollo de un prototipo de algoritmo propio implementado en hardware.

PROBLEMA CIENTÍFICO:

La principal incógnita científica radica en evaluar cómo la implementación de AES en hardware FPGA afecta la seguridad de datos en sistemas embebidos y si esta solución ofrece una mejora sustancial en términos de rendimiento, consumo de recursos y eficiencia en comparación con otros métodos de implementación de algoritmos criptográficos.

JUSTIFICACIÓN:

Este trabajo de titulación de maestría en Electrónica y Automatización se enfoca en el desarrollo y evaluación de un algoritmo criptográfico AES implementado en hardware FPGA para fortalecer la seguridad de datos en sistemas embebidos. El objetivo principal es aplicar una capa de protección a datos digitales mediante la codificación en VHDL, brindando un análisis exhaustivo de su rendimiento y eficiencia.

Se propone realizar una evaluación detallada utilizando plataformas de software de síntesis de tecnología digital como ISE Project Navigator, Vivado y Quartus, utilizando circuitos integrados FPGA de Intel/Altera y AMD/Xilinx. Esto permitirá comparar y contrastar aspectos clave, incluyendo el rendimiento, la utilización de recursos y otros parámetros relevantes entre las diferentes plataformas.

El trabajo se centrará en la implementación de AES en hardware FPGA mediante VHDL, explorando su viabilidad y optimización para su aplicación en sistemas embebidos. La evaluación exhaustiva considerará aspectos de seguridad, eficiencia de recursos y rendimiento en diferentes entornos FPGA, con el objetivo de proporcionar una comparativa detallada y fundamentada entre las plataformas de síntesis más utilizadas en la industria.

Se espera que este estudio contribuya al avance en el desarrollo de soluciones de seguridad en sistemas embebidos, ofreciendo información valiosa sobre la implementación de AES en hardware FPGA y su impacto en la protección de datos digitales en entornos sensibles.

Formulación del problema de investigación

La Dirección de Innovación y Desarrollo de la Armada del Ecuador está desarrollando un prototipo para la transmisión de datos digitales de información de contactos de interés mediante transmisión satelital desde una unidad de ala rotatoria, no obstante, esto propone el uso de un medio no completamente fiable por el cual pasará esta información de interés ante lo cual el desarrollo de un entorno donde la seguridad de la información es crucial se plantea mediante esta investigación la evaluación exhaustiva del algoritmo simétrico de encriptación AES sobre circuitos integrados FPGA de manera que cualquier intento de ejecución de ingeniería inversa sobre el algoritmo sea una tarea que sin ser considerada imposible sea extremadamente difícil y que conllevaría años de intentos infructuosos y a su vez el requerimiento de conocimientos y necesidades de recursos para ejecutar cada intento de ruptura del modelo de encriptación sean el factor inicial de disuasión.

La hipótesis central de este estudio propone una mejora significativa en la seguridad de trasmisión de datos desde una aeronave garantizando la confidencialidad de la información. Se

espera que este medio de encriptación que será desarrollado mediante codificación VHDL permita una integración efectiva del algoritmo, mientras que la evaluación comparativa a través de las plataformas de sintetización de circuitos integrados FPGA como Vivado y Quartus revelen diferencias significativas en el rendimiento, utilización de recursos y demás parámetros relevantes.

En este contexto y con el afán de abordar esta hipótesis se plantea la siguiente pregunta: ¿Cómo difieren los resultados de rendimiento y uso de recursos al interior de un FPGA al evaluar circuitos integrados de Intel/Altera y AMD/Xilinx utilizando las diferentes plataformas para síntesis de comportamiento digital sobre estos dispositivos electrónicos?

Este trabajo a desarrollarse en la DINDES se orienta a la investigación y experimentación rigurosa para responder esta pregunta con el objeto de contribuir al avance en la implementación de una solución que incremente la seguridad de manera efectiva a la transmisión de información mediante el uso de sistemas embebidos asociando a éstos con una capa de seguridad implementada sobre hardware FPGA y codificado en VHDL.

Objetivo General:

Evaluar un algoritmo de encriptación AES mediante VHDL para la implementación de seguridad sobre Sistemas Embebidos.

Objetivos Específicos:

- Examinar algoritmos de encriptación AES basados en VHDL que aporten robustez a Sistemas Embebidos.

- Sintetizar la operación de un algoritmo de encriptación AES en circuitos integrados FPGA.
- Evaluar el rendimiento y uso de recursos en diferentes plataformas de sintetización de hardware digital.

Planteamiento hipotético

La hipótesis central de este estudio propone que la implementación de AES en hardware FPGA aportará una mejora significativa en la seguridad de sistemas embebidos, garantizando la confidencialidad de los datos. Se espera que la codificación en VHDL permita una integración efectiva del algoritmo, mientras que la evaluación comparativa mediante Vivado y Quartus en circuitos integrados FPGA de Intel/Altera y AMD/Xilinx revelará diferencias significativas en el rendimiento, utilización de recursos y otros parámetros relevantes.

CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

1.1. Fundamentos de criptografía y AES

1.1.1. Principios de criptografía

La criptografía consiste esencialmente en el desarrollo de técnicas matemáticas que permitan ocultar información almacenada en algún medio físico o transmitida por un canal de comunicaciones y propone que estos algoritmos de base matemática sean funciones eficientes y sencillas de evaluar; no obstante, su función inversa sea a su vez muy difícil de obtener mediante el uso de medios computacionales (Jirón, 2022).

1.1.2. AES (Advanced Ecryption Estandar)

En 1997 el Instituto Nacional de Estándares y Tecnología entidad adscrita al Departamento de Comercio de los E.E.U.U. lanzó una convocatoria pública para el desarrollo de un estándar de cifrado avanzado; al que llamaría por su acrónimo AES, esta convocatoria se desarrolló mediante tres conferencias a través de las cuales se llegó a la selección de un algoritmo de cifrado simétrico diseñado por dos criptógrafos belgas; Vincent Rijmen y Joan Daemen, y derivado de los nombres de estos dos autores nació lo que hoy se conoce bajo el nombre de Algortimo Rijndael.

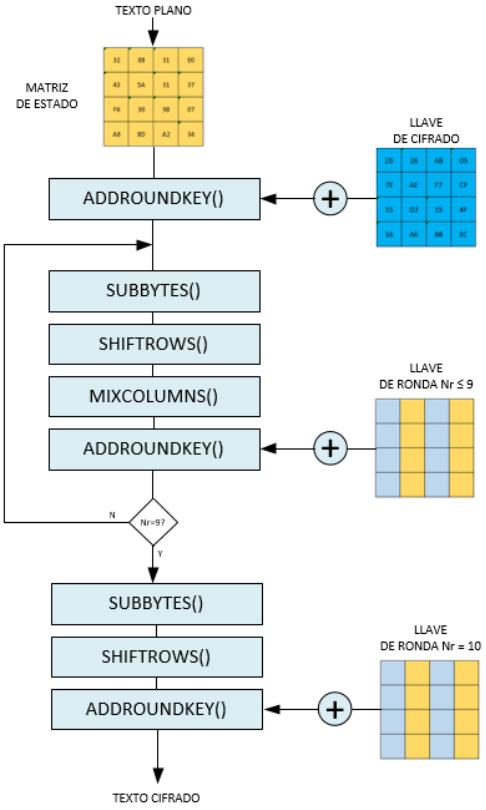
De acuerdo a la descripción desarrollada en el contexto de los Estándares Federales de Procesamiento de la Información, FIPS 197 para AES, el núcleo de este algoritmo de encriptación contiene una secuencia de transformaciones fijas de una matriz de estado llamada ronda, cada ronda requiere una entrada adicional llamada clave de ronda, esta clave de ronda

es un bloque que normalmente se representa como una secuencia de 4 palabras (16 bits); además, contiene una rutina de expansión la cual toma la clave de cifrado del bloque como una entrada y genera las claves redondas como salida (NIST, 2022).

En la figura 1 podemos observar el algoritmo de cifrado AES mismo que toma un texto plano para armar una matriz de estado con el objeto de procesar esta matriz hasta entregar un texto cifrado.

Figura 1

Algoritmo de encriptación AES



Nota. Adaptado de *Proceso de Encriptación AES*, por Nimbus Nijas on WordPress 2012, WordPress (https://nimbusninjas.wordpress.com/wp-content/uploads/2012/01/aes-encryption-process1.png).

Este algoritmo de encriptación AES contiene un núcleo estructurante en el cual se desarrollan una serie de permutaciones y operaciones aritméticas que en términos generales se conocen bajo los nombres de SubBytes, ShiftRows, MixColumns y AddRoundKey. El subproceso SubBytes; del algoritmo, aplica una tabla de sustitución de 16x16 bytes conocida como SBox en la cual se contienen todos los 255 caracteres posibles del código estándar americano para el intercambio de información (ASCII); conocida también como tabla extendida, siendo el uso de esta tabla una de las razones de la robustez del método de encriptación si consideramos que puede ser modificada en términos numéricos en 16^{16} combinaciones posibles, lo que nos da una escalofriante cifra de 160.000.000.000.000.000 de combinaciones de la tabla ASCII extendida, desarrollar un ataque manual o por fuerza bruta requeriría el trabajo continuo de varias generaciones o recursos computacionales no disponibles comercialmente en la actualidad. El subproceso ShiftRows desarrolla diferentes desplazamientos en las filas de la matriz de estado mientras el subproceso MixColumns efectúa una mezcla dentro de cada columna de la matriz de estado y finalmente el subproceso AddRoundKey efectúa una combinación con una clave de ronda de estado, haciendo entonces que el algoritmo sea muy complejo de des-encriptar si no se tiene la llave y/o la configuración de SBox correcta.

El pseudocódigo propuesto para la implementación de este algoritmo extraído de la documentación oficial de la presentación de la norma propone las siguientes acciones para el proceso de cifrado:

Pseudocódigo para el proceso de Cifrado

```
1    procedure CIPHER(in, w, Nr)
2            state ← in
```

```
3                 state ← ADDROUNDKEY(state, w[0..3])
4     for round from 1 to Nr − 1 do
5           state ← SUBBYTES(state)
6           state ← SHIFTROWS(state)
7           state ← MIXCOLUMNS(state)
8           state ← ADDROUNDKEY(state, w[4 ∗ round..4 ∗ round + 3])
9     end for
10          state ← SUBBYTES(state)
11          state ← SHIFTROWS(state)
12          state ← ADDROUNDKEY(state, w[4 ∗ Nr..4 ∗ Nr + 3])
13    return state
14    end procedure
```

El primer paso dispuesto en la línea 2 del pseudocódigo propone copiar la entrada de datos o información que se desea encriptar en la matriz de estados, luego de que se adiciona una ronda de clave (línea 3) la matriz de estado se transforma progresivamente mediante la ejecución del número de rondas; que para AES128 es de 10, y se iteran como se puede observar entre las líneas 4 y 12, la ronda final comprendida entre las líneas 10 y 12 difiere su aplicación en que se omite la última transformación de mezcla de columnas, una vez ejecutado este algoritmo se obtiene la matriz de estado transformada como se observa en la línea 13 (NIST, 2022).

1.2. Desarrollo teórico y conceptual

1.2.1. VHDL y diseño de hardware digital

El estándar IEEE1076-2019 define la sintaxis y normas del lenguaje de descripción de hardware. Actualmente se le considera ya no solo un lenguaje de descripción sino de verificación de hardware lo que implica que la potencialidad y fortalezas del lenguaje de descripción de circuitos y sistemas digitales se ha convertido en una norma por defecto para el diseño de comportamiento digital en la industria de prototipos (IEEE, 2019).

El algoritmo de encriptación AES es un estándar de seguridad de información que puede ser implementado a nivel de software, firmware, hardware o cualquier combinación posible entre estos recursos tecnológicos; no obstante, la propuesta de esta investigación se fundamenta en la necesidad de validar la implementación en hardware FPGA debido a que en esencia el uso de esta tecnología propone un incremento del nivel de seguridad por cuanto desarrollar ingeniería inversa sobre un circuito integrado de éste tipo es extremadamente complejo y costoso tanto en recursos, conocimiento e instrumentación, así como en tiempo.

Estudios anteriores han demostrado la posibilidad de desarrollar encriptación mediante el algoritmo AES sobre circuitos integrados FPGA de la serie Arria 10GX de Intel Altera en el que inclusive se ha podido utilizar el procesador Nios II, uno de los objetivos primordiales es demostrar cuan eficiente y rápido puede ser el algoritmo (Sideris y otros, 2019).

1.2.2. Implementación de algoritmos en VHDL

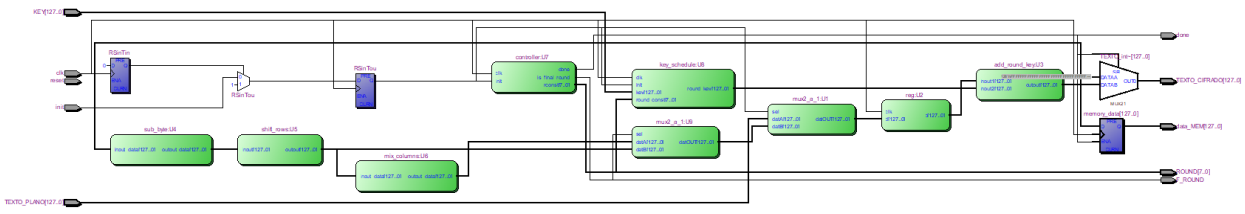
La utilización de VHDL como medio de construcción de algoritmos de cifrado seguro se basa en el aumento de complejidades para el desarrollo de ingeniería inversa inherente al uso de dispositivos FPGA, no obstante para el desarrollador este circuito integrado aporta con gran flexibilidad y reconfigurabilidad (Gupt y otros, 2021).

La implementación del algoritmo de encriptación AES en hardware siempre será más seguro que desarrollarlo en software, incluso se puede buscar mejorar la eficiencia mediante métodos de sumas y desplazamientos que busquen optimizar el uso del área, reducción de consumo de energía y aumento de la velocidad computacional del algoritmo (Divya y otros, 2021).

La propuesta implementa un algoritmo base para la encriptación de datos digitales AES128 descrito en VHDL, en la Figura 2 se puede observar la descripción RTL del conjunto de encriptación AES128 modelado lo más cerca posible a lo que determina la norma.

Figura 2

Descripción RTL de encriptador AES128



Este algoritmo de encriptación está constituido por 9 sub-bloques cuyo propósito es generar una palabra de 128 bits procesada mediante el núcleo del proceso de encriptación descrito en el pseudocódigo anterior.

En primera instancia tenemos un bloque multiplexor de 128 bits que toma el texto plano o texto a cifrar, procedimiento que arranca con un bit de inicio. La señal reg_input de 128 bits es ingresada a un registro para capturar el texto, el registro provee esta trama al módulo add_round_key mediante la señal reg_output; éste módulo agrega la clave de ronda y entrega mediante la señal subbox_input al módulo sub_byte el cual es el encargado de realizar la primera transformación no lineal mediante la aplicación de la SBox, luego de este proceso se entrega mediante señal subbox_output al siguiente proceso en el módulo shift_rows donde se

realiza un procedimiento de desplazamiento cíclico de las filas mediante la señal shiftrows_output se entrega los 128 bits procesados al módulo mix_columns donde se realiza una operación algebraica a la matriz de estado. Aquí se realiza una multiplicación por una matriz fija definida por el estándar, esta multiplicación se realiza dentro del campo de Galois cuidando que los términos de los polinomios no excedan del exponente 7 ya que este desborde saldría del campo de 128 bits con que se tratan los datos; entonces pues se procede en esos casos se aplica el teorema del binomio irreductible para $x^8 = x^4 + x^3 + x + 1$ se reemplaza con el polinomio descrito y continúa la operación. A este nivel se adiciona la clave de ronda, pero cuando se ha cumplido con la última de las rondas se salta el procedimiento de mezcla de columnas como dispone la norma. La señal feedback es la encargada de devolver la trama de 128 bits que ya ha sido procesada por el algoritmo hasta la culminación del procedimiento.

1.3. Sistemas embebidos y seguridad

1.3.1. Arquitectura de Sistemas Embebidos

El horizonte futuro para la implementación del algoritmo de encriptación es su uso en el fortalecimiento de la seguridad de la información procesada por un Sistema Embebido (SE). Estos, actualmente juegan un papel importantísimo en el desarrollo de prototipos para la validación conceptual de grandes sistemas de comunicación; sin embargo, al ser dispositivos de bajo costo y aunque sus prestaciones los califican para el uso en pruebas de concepto es necesario reforzar algunas de sus características, ahí la importancia de esta investigación.

En el contexto del uso de algoritmos de encriptación en sistemas embebidos investigadores; han concluido que, efectivamente AES tiene un rendimiento superior en cifrado

y descifrado debido a que el núcleo de su modelo matemático se fundamenta en sustituciones, permutaciones y transformaciones lineales en bloques de 16 bits (Gonzalez y otros, 2021).

Estructuralmente un SE está constituido por un procesador cuya dedicación no está orientada al soporte de un sistema operativo completamente multitareas como Windows sino que se espera éste administre unas pocas tareas específicas para lo cual tiene una memoria que le permite almacenar los algoritmos computacionales (programas) a ejecutar; normalmente desarrollados en Python, y los datos sujeto del procesamiento. Contiene además el sistema embebido alguno que otro periférico de manera que le sea posible interactuar con el entorno. Es importante tomar en consideración que los SE utilizados están optimizados para obtener requisitos específicos de rendimiento, consumo y administración de energía, tamaño, peso y costo.

Un procesador RISC-V está dotado de un conjunto reducido de instrucciones basadas en código abierto de manera que sea posible diseñar, fabricar, adaptar y comercializar chips con aplicaciones específicas sin tener que recurrir a pago por derechos de autor en el contexto del uso de los circuitos integrados (Jimenez Santiago, 2023)

Las especificaciones técnicas de la Raspberry Pi4 se encuentran en la Tabla 1; a continuación, en la cual podemos resaltar el procesador, memoria y periféricos disponibles.

Tabla 1

Especificaciones Técnicas de Raspberry Pi4

ESPECIFICACIONES TÉCNICAS RASPBERRY PI4

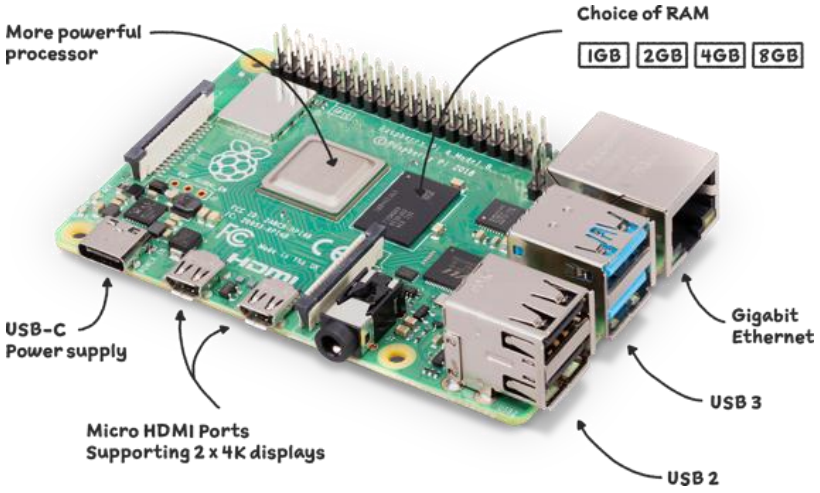
Procesador	Broadcom BCM2711, SoC de cuatro núcleos Cortex-A72 (ARM v8) de 64 bits a 1,8 GHz
Memoria	4GB LPDDR4-3200 SDRAM
Conectividad	IEEE 802.11ac inalámbrico de 2,4 GHz y 5,0 GHz, Bluetooth 5.0, BLE Gigabit Ethernet 2 puertos USB 3.0; 2 puertos USB 2.0.
Pines uso general	Conector GPIO estándar de 40 pines Raspberry Pi (totalmente compatible con placas anteriores)
Audio y Video	2 × puertos micro-HDMI® (se admiten hasta 4kp60) Puerto de pantalla MIPI DSI de 2 carriles Puerto de cámara MIPI CSI de 2 carriles Puerto de audio estéreo y vídeo compuesto de 4 polos H.265 (decodificación 4kp60), H264 (decodificación 1080p60, codificación 1080p30) OpenGL ES 3.1, Vulkan 1.0
Almacenamiento	Ranura para tarjeta Micro-SD para cargar el sistema operativo y almacenamiento de datos
Alimentación	5 V CC a través del conector USB-C (mínimo 3 A*)

	5 V CC a través del conector GPIO (mínimo 3 A*) Alimentación a través de Ethernet (PoE) habilitada (requiere PoE HAT separado)
Temperatura	Temperatura de funcionamiento: 0 – 50 grados C ambiente

Este pequeño computador embebido está constituido por un procesador Cortex-A72 de cuatro núcleos, 4 Gb de RAM, puerto Gigabit Ethernet y un GPIO de 40 pines como características relevantes, en la figura 3 observaremos una Raspberry Pi4.

Figura 3

Raspberry Pi4



Nota. Tarjeta *Raspberry Pi* [Fotografía], por Raspberry, 2023, Raspberry (https://www.raspberrypi.com/products/raspberry-pi-4-model-b/).

1.3.2. Seguridad en Sistemas Embebidos

El uso de sistemas embebidos facilita su aplicación en entornos donde el espacio para la implementación del desarrollo es muy reducido; ante esto, el soporte para estándares de comunicación como WiFi, Bluetooth, Ethernet y Serial aunque aumenta las probabilidades de uso aumenta también el riesgo de que la información que maneja u opera sea capturada para usos no autorizados (Ferreira & Silva, 2020)

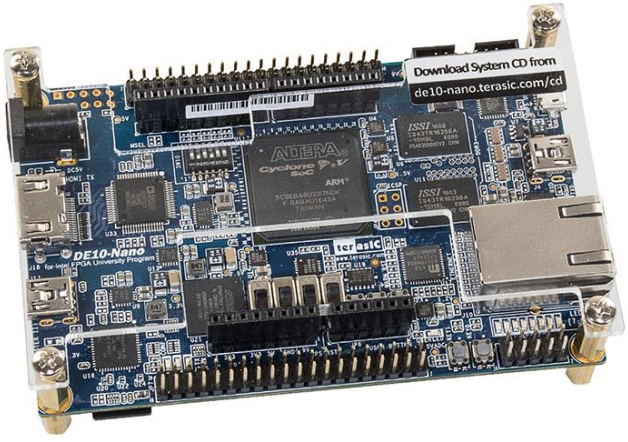
1.4. Plataformas de síntesis y evaluación

1.4.1. Plataformas FPGA y herramientas de síntesis

Las plataformas para síntesis y descripción de circuitos integrados FPGA proponen el uso de herramientas como ISE Porject Navigator, Quartus, Vivado, Vitis (Olivares & Soto, 2024).

Figura 4

DE10 nano



Nota. Tomado de DE10-Nano Kit [Fotografía], por Terasic (https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=167&No=1046).

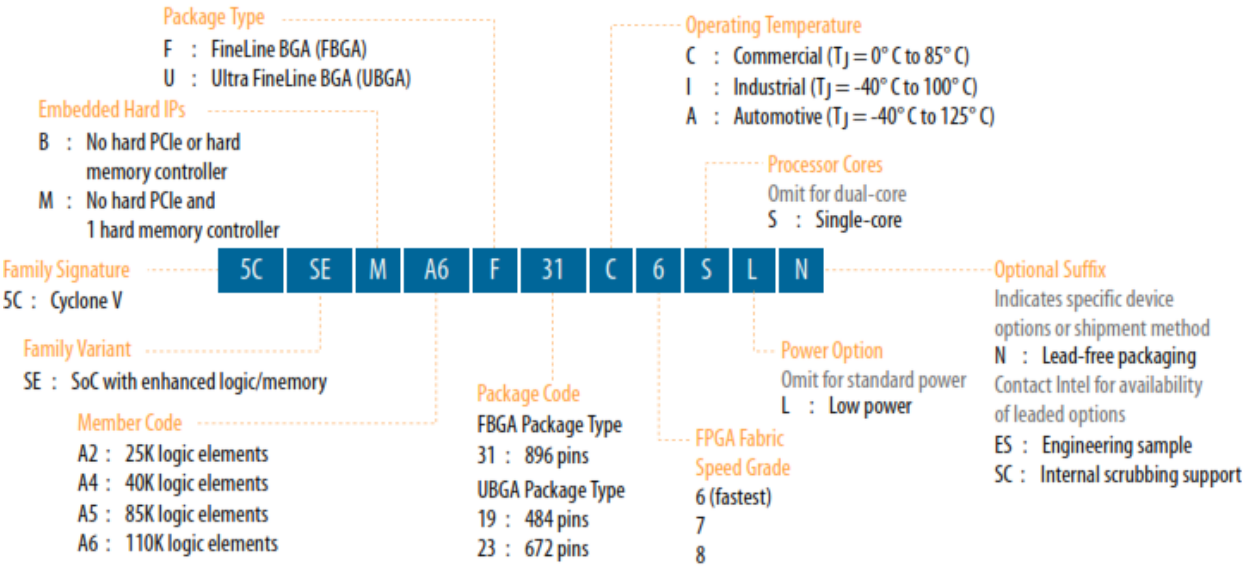
La figura 4 muestra una tarjeta de prototipado DE10-nano fabricada por Terasic, la cual incluye el circuito integrado FPGA SE 5CSEBA6U23I7NDK cuya denominación indica en términos generales que pertenece a la familia Cyclone®V de Intel (antes Altera por ello su logo en el dispositivo), con tecnología litográfica de implementación de 28 nanómetros, de grado industrial con 672 pines de los cuales tiene a su disposición 288 pines que pueden ser configurados a conveniencia por el desarrollador sea como salida o como entrada y en cuanto al procesador embebido es de características Dual Core. La tarjeta de entrenamiento DE10-nano incluye un reloj de 50 MHz conectado al pin V11 del FPGA.

1.4.1.1. El FPGA Intel 5CSEBA6U23I7NDK

Para el análisis de las especificaciones técnicas del circuito integrado FPGA de la familia Cyclone V empezaremos desglosando el significado de su número de parte, mismo que aporta con información relevante, en la figura 5 podemos observar la descripción del número de parte del dispositivo donde revela información respecto de la familia a la que pertenece 5C, refiérase a la familia Cyclone 5, SE indica que corresponde a la variante Sistem-on-Chip con lógica/memoria mejorada, con la letra B especifica que no cuenta con controlador PCI embebido ni controlador de memoria, el término A6 especifica que contiene 110000 elementos lógicos, la letra U indica que éste circuito integrado corresponde al encapsulado UBGA, el número 23 especifica que este dispositivo tiene 672 pines, la letra I determina que es de grado industrial y que soporta temperaturas de entre -40ºC hasta 100ºC, el número 7 especifica el grado de velocidad del dispositivo. Se omiten las letras antepenúltima y penúltima del de la figura lo que indica que en la parte del procesador este está constituido por dos núcleos y es de características estándar respecto de la potencia de disipación.

Figura 5

Descripción de número de parte de 5CSEBA6U23I7N



Nota. Tomado de Cyclone® V Device Overview [Captura], por Intel (https://www.intel.com/content/www/us/en/docs/programmable/683694/current/available-options-64019.html).

Para el propósito de esta investigación nos interesa conocer las especificaciones técnicas del circuito integrado que albergará el código VHDL para la implementación de un algoritmo de encriptación, estas especificaciones técnicas para este componente son:

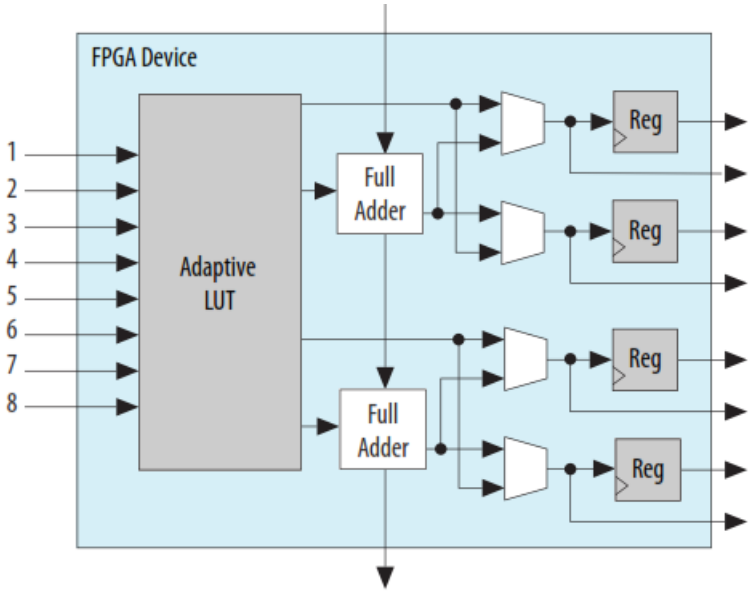
- Elementos lógicos (LE): 110.000
- Módulos de lógica adaptativa (ALM): 41.910
- Registros de módulo de lógica adaptativa (ALM): 166.036
- Entradas/Salidas con PLL para el FPGA: 6
- Entradas/Salidas con PLL para el HPS: 3
- Memoria embebida máxima: 6191 Mb

- Bloques para procesamiento digital de señales (DSP): 112
- Hard Processor System: Dual Core ARM Cortex A9
- Controladores Hard Memory: Si
- Interfaces de memoria externa (EMIF): DDR2, DDR3, LPDDR2

La arquitectura de éste circuito integrado FPGA bajo análisis contempla 41910 módulos de lógica adaptativa que es el término que el fabricante emplea para describir el conjunto de componentes electrónicos conformado por las Look-Up-Tables (LUT), registros, multiplexores y rutas de interconexión, de ésos elementos una LUT es quizás uno de los elementos comunes en toda FPGA que más interés concita ya que argumentan la operación de una tabla de búsqueda que permite la implementación de cualquier función lógica. En principio al ser esta LUT como se puede observar en la figura 5 un recurso que tiene 8 entradas podríamos decir que esta podría albergar el comportamiento de una función lógica con ocho variables, el número de entradas de una LUT varía en función de las familias y marcas del fabricante del circuito integrado FPGA. Estos recursos de tablas de búsqueda se complementan con registros como se puede observar en la figura 6 los cuales permiten construir comportamiento secuencial como contadores, registros de desplazamiento y más útiles e importantes aún este conjunto de elementos nos permitiría instrumentar con ellos máquinas de estados finitos (FSM). Dicho esto, podemos considerar entonces pues que con la inclusión de sumadores completos dentro de esta arquitectura es posible entonces desarrollar operaciones básicas como suma y restas. Para la implementación de cualquier diseño lógico la interconexión que propone entre estos elementos y los conjuntos de estos elementos adyacentes es de vital importancia en el diseño de estos dispositivos FPGA las rutas de interconexión que permiten plasmar los requerimientos del diseñador y para expandir las capacidades según requerimiento de diseño se cuenta con sendos multiplexores.

Figura 6

Módulos ALM para dispositivos Cyclone V



Nota. Tomado de Cyclone® V Device Overview [Captura], por Intel (https://www.intel.com/content/www/us/en/docs/programmable/683694/current/adaptive-logic-module.html).

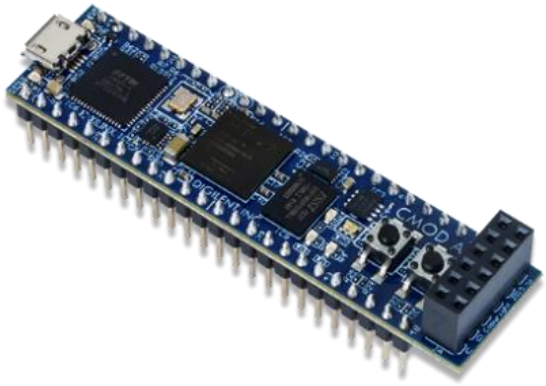
Es importante mencionar que en particular este dispositivo cuenta con un procesador embebido ARM Cortex A9 de doble núcleo; no obstante, para el propósito de esta investigación vamos a prescindir de este recurso.

Otra de las tarjetas de entrenamiento que utilizaremos para el desarrollo de esta investigación es el módulo Cmod-A7. Esta tarjeta de prototipado incluye el circuito integrado FPGA Artix-7 XC7A15T-1CPF236C cuya denominación indica en términos generales que pertenece a la familia Artix®7 de AMD (antes Xilinx por ello su logo en el dispositivo), con tecnología litográfica de implementación de 28 nanómetros, de grado industrial con 250 pines

GPIO de propósito general que pueden ser configurados a conveniencia por el desarrollador sea como salida o como entrada.

Figura 7

Módulo Cmod-A7



***Nota*.** Tomado de Cmod A7 [Captura], por Digilent (https://digilent.com/reference/programmable-logic/cmod-a7/start).

La tarjeta de entrenamiento CMod-A7 incluye un reloj de 12 MHz conectado al pin L17 del FPGA.

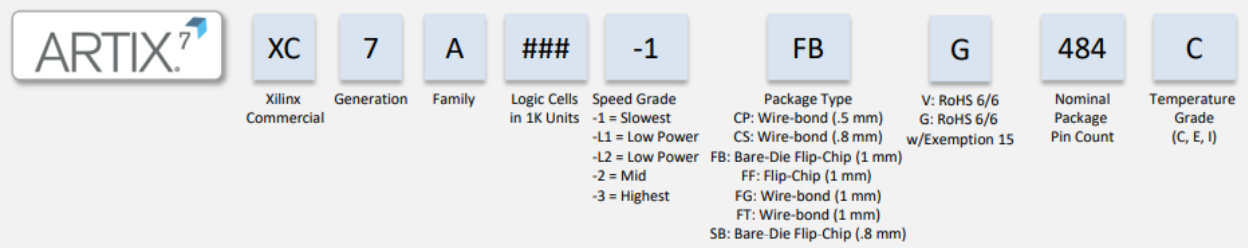
1.4.1.2. El FPGA AMD Artix-7 XC7A15T-1CPF236C

Para el análisis de las especificaciones técnicas de este dispositivo FPGA de la familia Artix7 empezaremos desglosando el significado de su número de parte, mismo que aporta con información relevante, en la figura 8 podemos observar la descripción del número de parte del dispositivo donde revela información respecto de la familia a la que pertenece, para empezar el término XC es la denominación comercial de Xilinx fabricante original del dispositivo, el número 7 se refiere a la generación a la que pertenece, la letra A define a la familia de dispositivos dentro de esa generación, el término 15T especifica el número de celdas lógicas contenidas, el

-1 indica que es el dispositivo más lento en grado de velocidad de entre los dispositivos de la familia a la que pertenece, el término CP se refiere al tipo de encapsulado del circuito integrado, la letra F determina si la construcción de este dispositivo se considera libre de plomo o no. El número 236 identifica el número de pines que tiene el circuito integrado y finalmente la letra C define el grado de temperatura en que puede operar este componente.

Figura 8

Descripción de número de parte de XC7A15T-1CPF236C



Nota. Tomado de Technical Information Portal [Captura], por AMD, (https://docs.amd.com/v/u/en-US/7-series-product-selection-guide).

Del mismo modo que para el circuito integrado analizado anteriormente, para el propósito de esta investigación nos interesa conocer sus especificaciones técnicas, contiene el siguiente arreglo de bloques lógicos configurables, considerando que en esta tecnología cada Slice tiene embebidos cuatro LUT de seis entradas.

- Celdas lógicas: 16.640
- Slices: 2600
- Máxima memoria RAM distribuida: 200Kb
- DSP48E1 Slices: 45
- Block RAM 18Kb: 50
- Block RAM 36Kb: 25

- Block RAM Max (Kb): 900
- CMTs: 5
- PCIe: 1
- GTPs: 4
- XADC Blocks: 1
- Total I/O Banks: 5
- Max User I/O: 250

En la arquitectura de este dispositivo de la serie 7 se reconoce la existencia de celdas lógicas como unidad básica de la lógica programable descrita por este fabricante, estas celdas lógicas contienen una LUT, un flip-flop y un multiplexor para sintetizar cualquier función lógica con posibilidad de almacenar estos estados lógicos para instrumentar diseños de lógica secuencial. Los Slices de AMD en la familia de esta serie contienen cuatro LUTs y 8 Flip-Flops lo que les permite estructurar lógica más compleja.

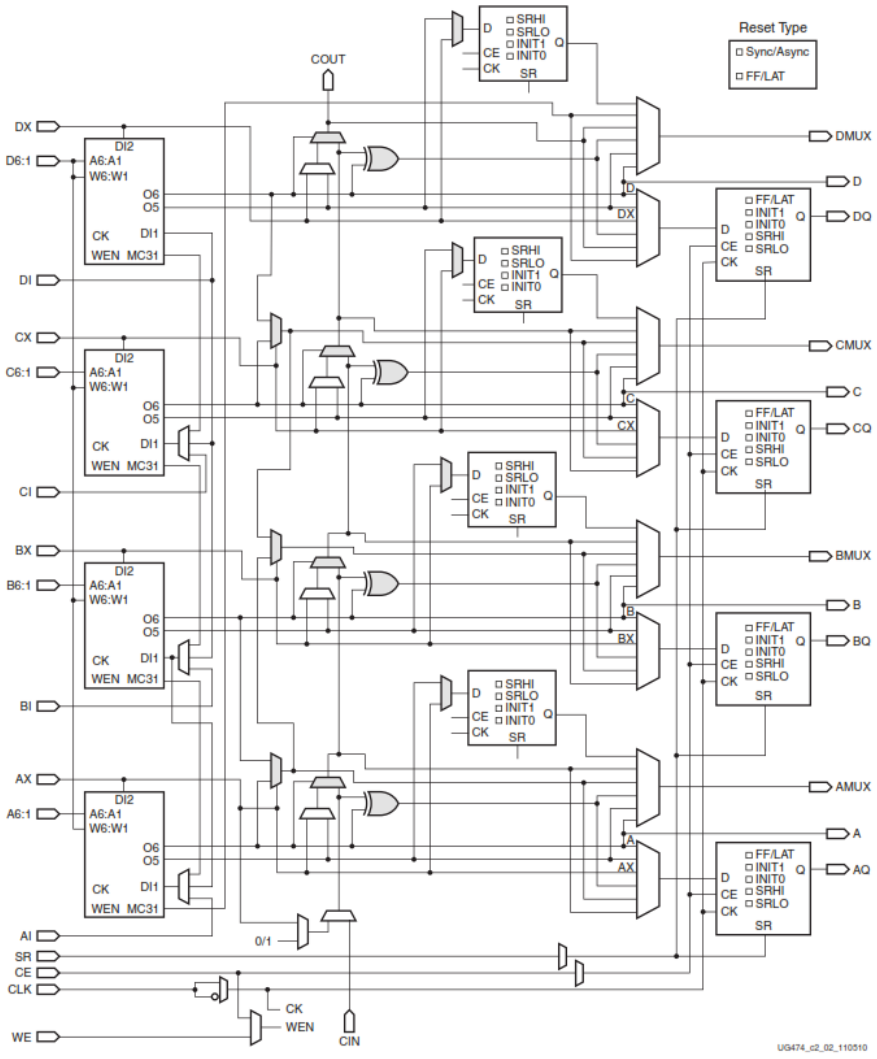
Este dispositivo cuenta también con RAM distribuida, se le conoce así porque es capaz de utilizar las LUT para implementar cantidades pequeñas de memoria RAM de manera distribuida a través de todo el FPGA. El bloque DSP481E se constituye en un bloque especializado para el desarrollo de procesamiento digital, este bloque incluye un multiplicador de 25x18 bits, incluye también recursos de acumulación y pre-adición, con ello se puede desarrollar o implementar soluciones de filtrado digital, transformadas y funciones aritméticas complejas.

Este componente cuenta además con un recurso denominado Clock Management Tiles (CMT) que son recursos destinados a gestionar y generar señales de reloj dentro del dispositivo, con lo cual se pueden multiplicar, dividir y ajustar incluso las fases de las señales de reloj lo que permite sincronizar partes del diseño que así lo requieran.

Otro recurso que tiene este dispositivo es el de los PCIs (Peripheral Component Interconnect Express), esta estructura especializada está optimizada para la construcción de interfaces de alta velocidad para el control de periféricos basados en esta estructura tecnológica tales como controladores de video, tarjetas de red y demás periféricos de alta velocidad.

Figura 9

Diagrama de un SliceM



Nota. Tomado de 7 Series FPGAs Configurable Logic Block, User Guide [Captura], por Xilinx, (https://docs.amd.com/v/u/en-US/ug474_7Series_CLB).

Se puede observar un SLICEM con que cuenta la tecnología de esta familia Artix7, se aprecia la arquitectura constituida por 4 LUTs, ocho elementos de almacenamiento (Flip-Flops), multiplexores y lógica de acarreo. Todos estos elementos permiten constituir circuitos lógicos, aritméticos y funciones de memoria ROM. Algunos Slices soportan funciones adicionales como almacenamiento de datos en RAM distribuida y/o registro de desplazamiento de 32 bits, estos SLICEs que soportan estas funciones adicionales son conocidos como SLICEM, tal como el de la figura 9.

La tarjeta de entrenamiento mostrada en la figura 10 corresponde a la NExys4-DDR la cual soporta facilidades para la implementación de circuitos lógicos combinatoriales sencillos hasta complejos sistemas basados en procesadores embebidos.

Figura 10

Nexys4 DDR

Esta tarjeta de prototipado incluye el circuito integrado FPGA XC7A100T-1CSG324C cuya denominación indica en términos generales que pertenece a la familia Artix®7 de AMD (antes Xilinx por ello su logo en el dispositivo), con tecnología litográfica de implementación de

28 nanómetros, de grado industrial con 250 pines GPIO de propósito general que pueden ser configurados a conveniencia por el desarrollador sea como salida o como entrada.

La tarjeta de entrenamiento Nexys4-DDR incluye un reloj de 100 MHz conectado al pin E3.

1.4.1.3. El FPGA AMD Artix-7 XC7A100T-1CSG324C

Del mismo modo que para el circuito integrado analizado anteriormente las especificaciones en cuanto al número de parte son equivalentes, así como también la descripción funcional de sus bloques elementales para el desarrollo de tecnología digital. Para el propósito de esta investigación nos interesa conocer sus especificaciones técnicas puntuales de las que podemos indicar contiene las siguientes características:

- Celdas lógicas: 101440
- Slices: 15850
- Máxima memoria RAM distribuida: 1188Kb
- DSP48E1 Slices: 240
- Block RAM 18Kb: 270
- Block RAM 36Kb: 135
- Block RAM Max (Kb): 4860
- CMTs: 6
- PCIe: 1
- GTPs: 8
- XADC Blocks: 1
- Total I/O Banks: 6
- Max User I/O: 300

La figura 11 muestra la tarjeta de prototipado DE0-nano la que a diferencia de la DE10-nano cuenta con un FPGA de otra familia, misma que no contiene procesador ARM embebido. El circuito dispositivo instalado en esta placa es el Cyclone IV EP4CE22F17C6N que es un dispositivo cuya familia ha sido fabricada con tecnología de Taiwan Semiconductor Manufacturing Company (TSMC) de 60 nm.

Figura 11

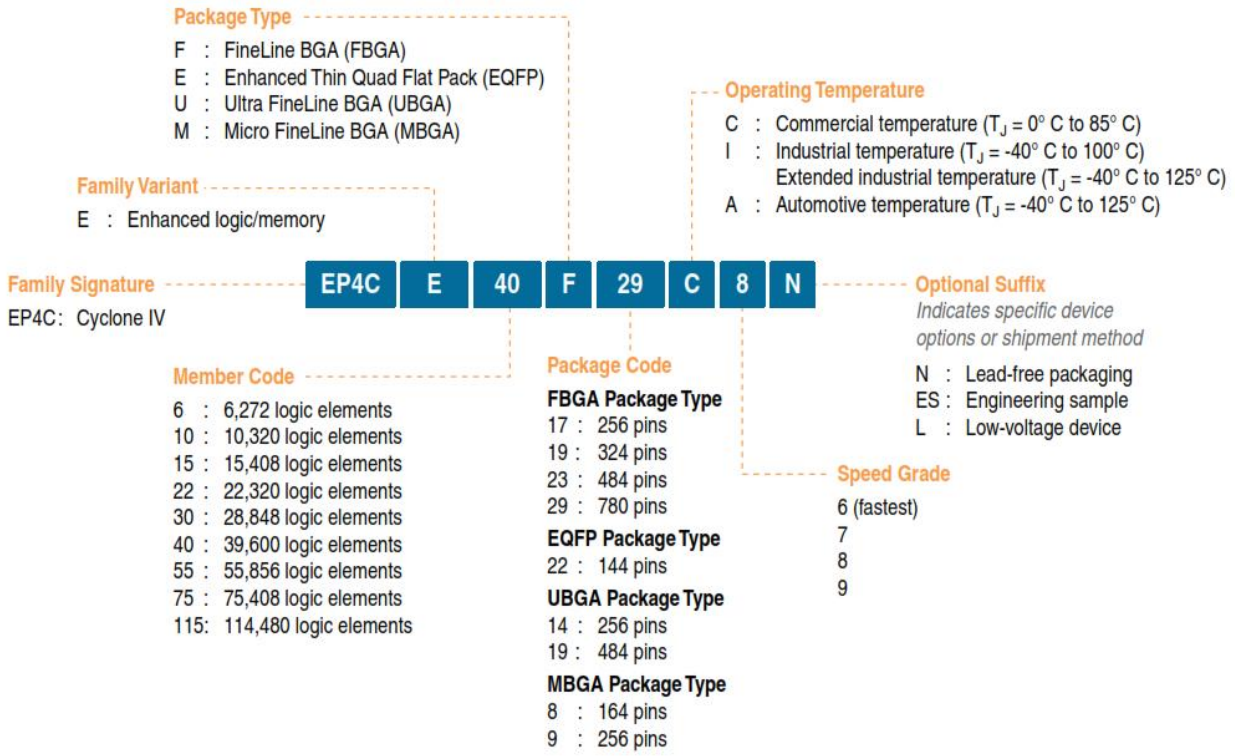
DE0-Nano

Nota. Tomado de DE0-Nano Development and Education Board [Captura], por Terasic (https://www.terasic.com.tw/cgi-bin/page/archive.pl?No=593).

1.4.1.4. El FPGA Intel Cyclone IV EP4CE22F17C6N

Figura 12

Descripción de número de parte de EP4CE22F17C6N



Nota. Tomado de *Manual del dispositivo Cyclone® IV* [Captura], por Intel, (https://www.intel.la/content/www/xl/es/content-details/653974/cyclone-iv-device-handbook.html).

Este dispositivo de la familia Cyclone IV nos indica de acuerdo a su número de parte que es de la variante de familia que contiene lógica y memoria mejorada, el número 22 nos indica que tiene 22320 elementos lógicos, su encapsulado es de tipo FBGA con encapsulado de 256 pines (por el número 17), la letra C nos indica que este dispositivo opera en rangos de temperatura comerciales; entre 0ºC y 85ºC, grado de velocidad 6 que es de lo más rápido en esa familia y la letra final N nos dice que su construcción es Lead-free (libre e plomo).

Las especificaciones técnicas de este circuito integrado son:

- Elementos lógicos (LE): 22320
- Memoria embebida: 594 Kbits
- Multiplicadores embebidos 18x18: 66
- PLL de propósito general: 4
- Redes de relojes globales: 20
- Bancos de I/O de usuario: 8
- Máximo número de I/O disponibles a usuario: 153

En la figura 13 podemos observar una descripción de los Elementos Lógicos que constituyen la lógica fundamental de esta familia tecnológica la cual cuenta con LUTs de 4 entradas, un registro programable, registro y lógica de acarreo.

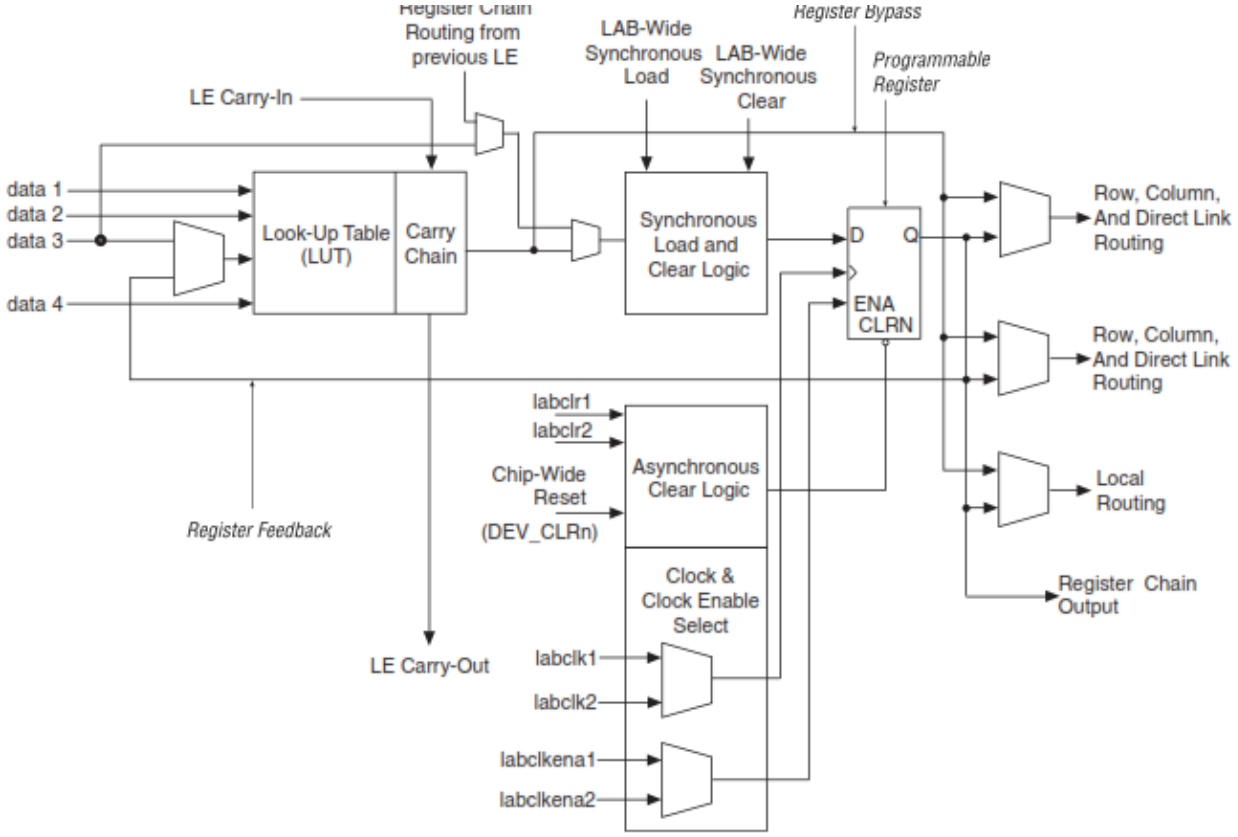
Esta LUT a través de su lógica de acarreo permite configurar funciones lógicas de mayor cantidad de entradas, pudieren sus salidas salir directamente sin pasar por los registros si así se necesitare; no obstante, también es posible configurar y activar el bloque de registros.

1.4.2. Métricas de evaluación

Uno de los aspectos que concita mayor interés a la hora de evaluar sistemas desarrollados sobre FPGAs es el consumo y la eficiencia del uso de la energía (Naiouf y otros, 2020).

Figura 13

Elementos Lógicos en la familia Cyclone IV



Nota. Tomado de *Manual del dispositivo Cyclone® IV* [Captura], por Intel, (https://www.intel.la/content/www/xl/es/content-details/653974/cyclone-iv-device-handbook.html).

1.4.3. La fiabilidad de la información y su importancia

El presente trabajo presenta un modelo probabilístico sobre estructuras lógicas propias de las FPGAs, esto es sobre los distintos CLBs (Configurable Logic Blocks), usualmente constituidos por LUTs (Look-up Tables) y Biestables (Flip-Flop) interconectados vía redes

programables PSM (Programmable Switch Matrix) y bloques de entrada / salida: IOBs (Input-output Blocks). Estas interconexiones y la descripción de lógica de transferencia de registros obedecen a configuraciones de la herramienta de diseño y aunque puede ser afinada se debe tomar en consideración que la robustez de la plataforma genera reportes con muy buena precisión de lo que va a ocurrir en el momento en que el flujo de bits sea descargado en el componente.

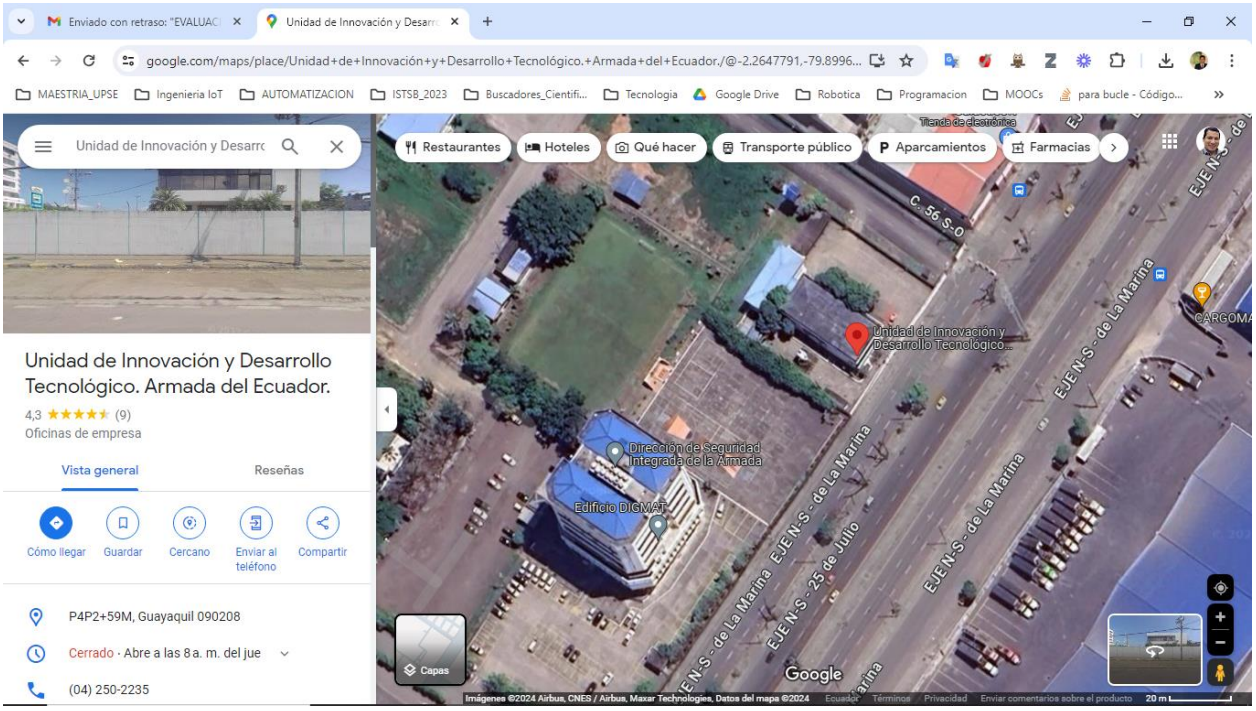
CAPÍTULO 2. METODOLOGÍA

2.1. Contexto de la investigación

La investigación se llevó a efecto en las instalaciones de la Dirección de Innovación y Desarrollo de la Armada del Ecuador; en el laboratorio de Desarrollo de Prototipos, Av de la Marina – Base Naval Sur (-2.26444, -79.89899), Guayaquil – Ecuador.

Figura 14

Dirección de Innovación y Desarrollo de la Armada



Nota. Adaptado *mapa de Guayaquil*, Ecuador en Google maps, Recuperado el 8 de Mayo, 2024, de: https://www.google.com.co/maps/@ -2.264571398357879, -79.89894400802832

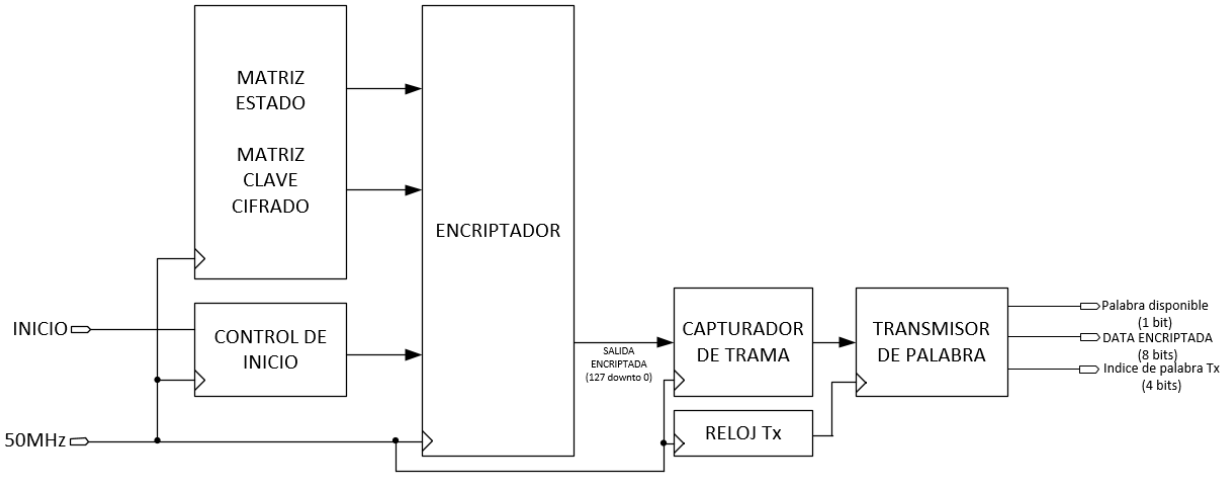
2.2. Diseño y alcance de la investigación

El planteamiento de esta investigación es de carácter experimental en la cual se propone desarrollar pruebas controladas a nivel de laboratorio efectuando manipulaciones directas en el diseño e implementación del algoritmo AES en VHDL, así como la variación de parámetros de optimización de recursos y pruebas de rendimiento en diferentes entornos de FPGA.

El alcance de la investigación implica analizar las características, operatividad y eficacia del algoritmo AES implementado en VHDL con el objeto de describir su funcionamiento para el aprovechamiento de la encriptación a futuro en sistemas embebidos quienes recibirán la trama encriptada de un mensaje cuyo procesamiento de encriptación se desarrollará enteramente en FPGA.

Figura 15

Propuesta de módulo de encriptación AES para modelamiento en FPGA



En la figura 15 podemos observar la propuesta inicial a desarrollar en el contexto de la configuración controlada del circuito integrado FPGA mediante el uso del lenguaje VHDL en la cual se propone la construcción de un bloque donde se cargarán los datos de la matriz de

estado y los datos de la matriz de clave de cifrado, todo esto partiendo de un reloj de sincronización de 50 MHz; es decir, que cada pulsación de sincronización se realizará a razón de 20 nanosegundos. En ese contexto los datos generados por las matrices ingresan de modo paralelo al módulo de encriptación en el cual hay que considerar que el proceso de permutaciones relacionadas al cifrado toma un tiempo evidentemente superior ya que cada paso es ejecutado a 20nS y el algoritmo requiere varias iteraciones de modo que se prevé el uso de un par de señales de reloj adicionales generadas por el módulo de división de frecuencia para sincronizar la trama encriptada. Todo el procesamiento entregará como respuesta una trama de 128 bits los cuales serán analizados mediante la herramienta de simulación provista en el software de sintetización de los FPGA.

2.3. Tipo y métodos de investigación

En este trabajo de investigación se recopilarán datos numéricos y estadísticos cuantificables sobre el rendimiento del algoritmo AES escrito en VHDL tales como velocidades de respuesta del algoritmo a una trama de datos ingresada y el tiempo que toma en encriptar, así como el uso de recursos del FPGA, por tanto, esta investigación será de carácter cuantitativa.

Respecto del método a utilizar durante el proceso de investigación de este tema será de características analítico sintético debido a que se propenderá a descomponer partes específicas del algoritmo AES y analizar su relación con la seguridad en sistemas embebidos, todo esto mediante la segmentación de partes específicas tributantes al módulo de encriptación y el módulo de cifrado mismo para analizar individualmente las respuestas a las modificaciones a ejecutar en el código de manera de poder obtener los datos en los terminales de salida para su evaluación con las herramientas del fabricante del circuito integrado FPGA bajo prueba.

2.4. Población y muestra

Respecto de la población de estudio para este proyecto de investigación, se estima hacer uso de las siguientes tecnologías basadas en Plataformas FPGA para evaluación y entrenamiento: DE0-nano, DE10-nano, Nexys4-DDR y CmodA7.

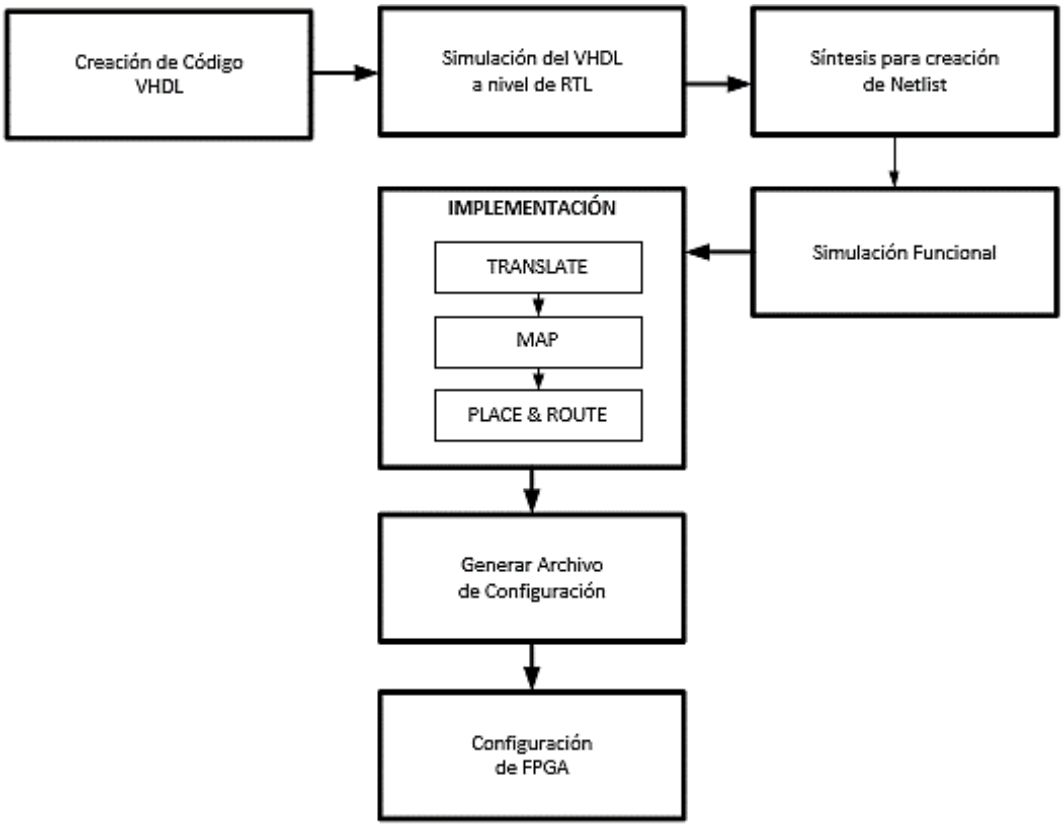
Debido a que el universo de opciones de marcas y modelos de plataformas FPGA es tremendamente amplio, y además muchas de estas opciones no están disponibles en el mercado local y su acceso supone un costo en tiempo y recursos elevado se efectuará un muestreo no probabilístico sobre algunas familias específicas de FPGA mediante la sintetización del algoritmo de encriptación de manera de poder determinar comparativamente la eficiencia del algoritmo en diversas plataformas tecnológicas.

2.5. Técnicas e instrumentos de recolección de datos

El desarrollo de los procesos y actividades para la sintetización de comportamiento digital requiere del uso de técnicas inherentes a la operación en las plataformas de desarrollo de los circuitos integrados que están contenidos en las tarjetas de prototipado a utilizar en esta investigación experimental, ello supone que pudieren existir diferentes formas de ejecutar el proceso de desarrollo desde la construcción del código VHDL hasta la carga del archivo de configuración en el FPGA; no obstante, es posible generalizar los pasos más importantes de donde se obtendrán los reportes respectivos que sirvan para el análisis propuesto en esta investigación, esta descripción general la podemos observar en la figura 16.

Figura 16

Flujo de diseño del FPGA



En lo referente a la creación del código VHDL partiremos de un diseño de módulo de encriptación que contiene todos los subprocesos que la norma AES128 propone. Una vez ingresado el código y el proceso de síntesis de donde se obtendrán los primeros reportes se procederá a desarrollar una simulación funcional para verificar paso a paso que el proceso de encriptación esté desarrollando las conversiones y permutaciones que corresponden. Luego de la simulación funcional que es lo más próximo al comportamiento del código en el componente se desarrollará el proceso de implementación, mismo que está conformado por tres pasos fundamentales conocidos como Translate, Map y Place & Route, con los cuales se lleva ese

comportamiento descrito en VHDL a los recursos que tiene el FPGA para poder sintetizar el comportamiento del algoritmo de encriptación además de desarrollar su simulación funcional.

Dadas las características de este tema de investigación se utilizará un enfoque mixto donde la parte cuantitativa permitirá recolectar datos numéricos y estadísticos sobre el rendimiento del algoritmo AES en VHDL, considerando lo siguiente:

- **Reportes de generados por las herramientas de síntesis e implementación**, con lo que se recolectarán datos cuantitativos generados por herramientas como Vivado y Quartus. Estos proporcionarán datos numéricos sobre el rendimiento, uso de recursos, tiempos de ejecución, entre otros.

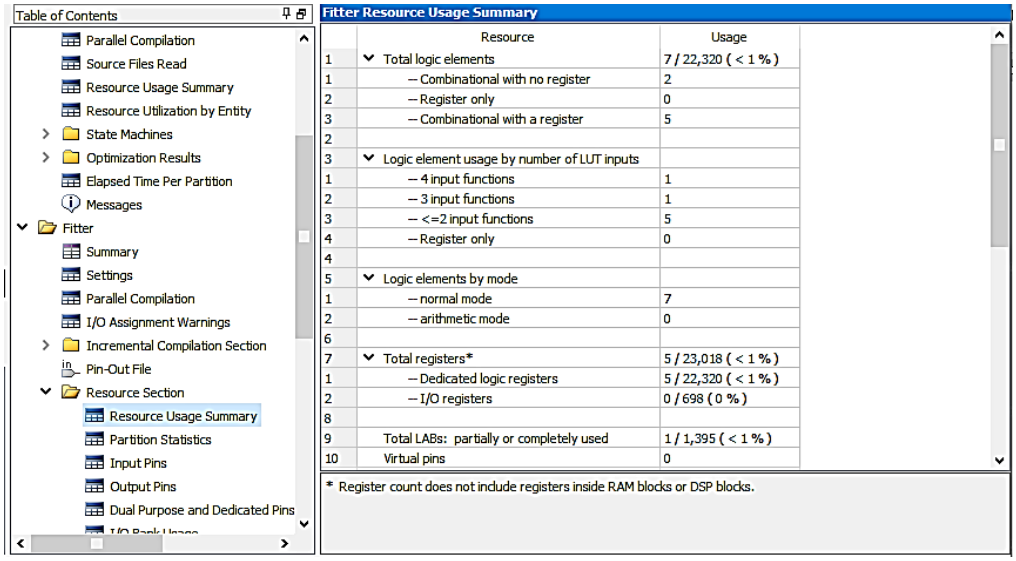
No obstante, se complementará esta información con observaciones cualitativas sobre la eficacia, usabilidad o desafíos identificados durante el proceso de investigación. Estos reportes detallados son de vital importancia ya que son el reflejo exacto de lo que ocurre con el diseño desde la primera hasta la última etapa.

Al igual que las herramientas de AMD/Xilinix las contenidas en Quartus II de Intel/Altera permiten obtener información detallada de los recursos y características del diseño implementado, en la figura 17 podemos observar en la columna de la izquierda (Tabla de contenidos) una muestra de los diferentes reportes que se generan del uso de la plataforma en el momento de la ejecución de los procesos de síntesis e implementación.

Dicho esto, es menester conocer algunos términos utilizados en los reportes para con ello tener un relacionamiento adecuado entre las características técnicas de los diversos circuitos integrados a analizar ya que al ser de marcas diferentes y aun dentro de la misma marca tenemos familias diferentes pudiere haber ligeras diferenciaciones en los términos que describen los recursos de cada FPGA.

Figura 17

Quartus II, Reportes



En la línea de Xilinx y los reportes en Vivado tenemos términos como Slice LUT, las Look-Up Table son los componentes fundamentales sobre los que se soporta el FPGA, dentro de cada Slice se puede implementar cualquier descripción lógica. Los Slice Registers son flip-flop que están contenidos dentro de los Slices con el objeto de registrar los estados lógicos obtenidos por las LUT si fuere necesario. Los F7 Muxes permiten a través de la multiplexión realizar una conmutación entre varias entradas, en este caso hasta 8 entradas, con esto es posible combinar los resultados obtenidos por varias LUT generando entonces funciones lógicas cada vez más complejas.

Los Slices son agrupaciones lógicas entre varios recursos contenidos en la FPGA, entre esos recursos podemos contar las LUT, los registros, multiplexores y lógica de acarreo.

Se pueden utilizar las LUT para implementar cualquier tipo de comportamiento combinatorial, incluso pueden configurarse como memoria RAM. Estos recursos pueden ser utilizados solos o emparejados con los Flip-Flops de ahí los términos LUT as Logic y LUT Flip-

Flop Pairs. El término Block RAM Tile se refiere a los bloques de memoria RAM dedicados dentro del dispositivo FPGA.

El término BFGCTRL se refiere a un buffer de control global que permite la distribución eficiente de la señal de reloj a través del circuito integrado FPGA.

La tecnología FPGA admite emulación de comportamiento mediante aproximación teórica del diseño amparado en las herramientas de simulación de código VHDL propias del fabricante como ISIM (ISE Simulator) de AMD/Xilinx, o de terceros como ModelSim antes de Mentor Graphic y a partir del 2017 propiedad de Siemens EDA, Simics de Intel FPGA. También es posible echar mano de la metodología de evaluación de diseños conocida como Hardware in the Loop donde de preferencia mediante el uso de las herramientas del fabricante del circuito integrado que se haya seleccionado para el diseño (sea Intel o AMD) se puede utilizar herramientas especializadas como ChipScope o SignalTap, mismas que permiten insertar un Analizador Lógico embebido en el diseño de manera que a través de él se determinen puntos de prueba específicos que sea necesario controlar dentro el diseño mediante una interfaz gráfica de usuario que permite visualizar la señal o conjunto de señales luego de descargar el archivo de configuración del circuito integrado FPGA. No obstante, muchas de ellas no son de acceso gratuito y si lo son tienen algún limitante respecto del número de canales a utilizar o en su defecto las especificaciones del circuito integrado.

2.6. Validez y confiabilidad de los instrumentos aplicados.

El Instituto Nacional de Estándares y Tecnología – NIST propone a través de su programa de validación de algoritmos criptográficos (CAVP) la posibilidad de desarrollar pruebas de validación de algoritmos desarrollados para hardware o software como requisito previo para la validación del módulo criptográfico. En ese sentido pone a disponibilidad de los desarrolladores

un protocolo de validación de los modos de operación para el cifrado de bloques. Este protocolo dispone de un conjunto de palabras de 16 caracteres tanto para texto plano como para llave de cifrado y su respectiva respuesta en texto cifrado que el algoritmo desarrollado debe cumplir; el protocolo completo con los resultados parciales de toda la acción del algoritmo a el banco de pruebas se encuentra en el anexo 1. Para efecto de resumir e identificar nuestro método de validación someteremos a la fórmula propuesta por el originador del estándar, tal como aprecia en la tabla 2:

Tabla 2

Protocolo de pruebas para AES128

TIPO DE DATO	PALABRA EN HEXADECIMAL
TEXTO PLANO:	2A179373117E3DE9969F402EE2BEC16B
LLAVE DE CIFRADO:	3C4FCF098814F7ABA6D2AE2816157E2B
TEXTO ENCRIPTADO:	97EF6624F3CA9EA860367A0DB47BD73A

Considerando que las herramientas de síntesis generan reportes, se verificará que los reportes generados abarquen los aspectos técnicos esenciales del rendimiento del algoritmo implementado extrayendo la información relevante y específica sobre el uso de recursos y tiempos de respuesta.

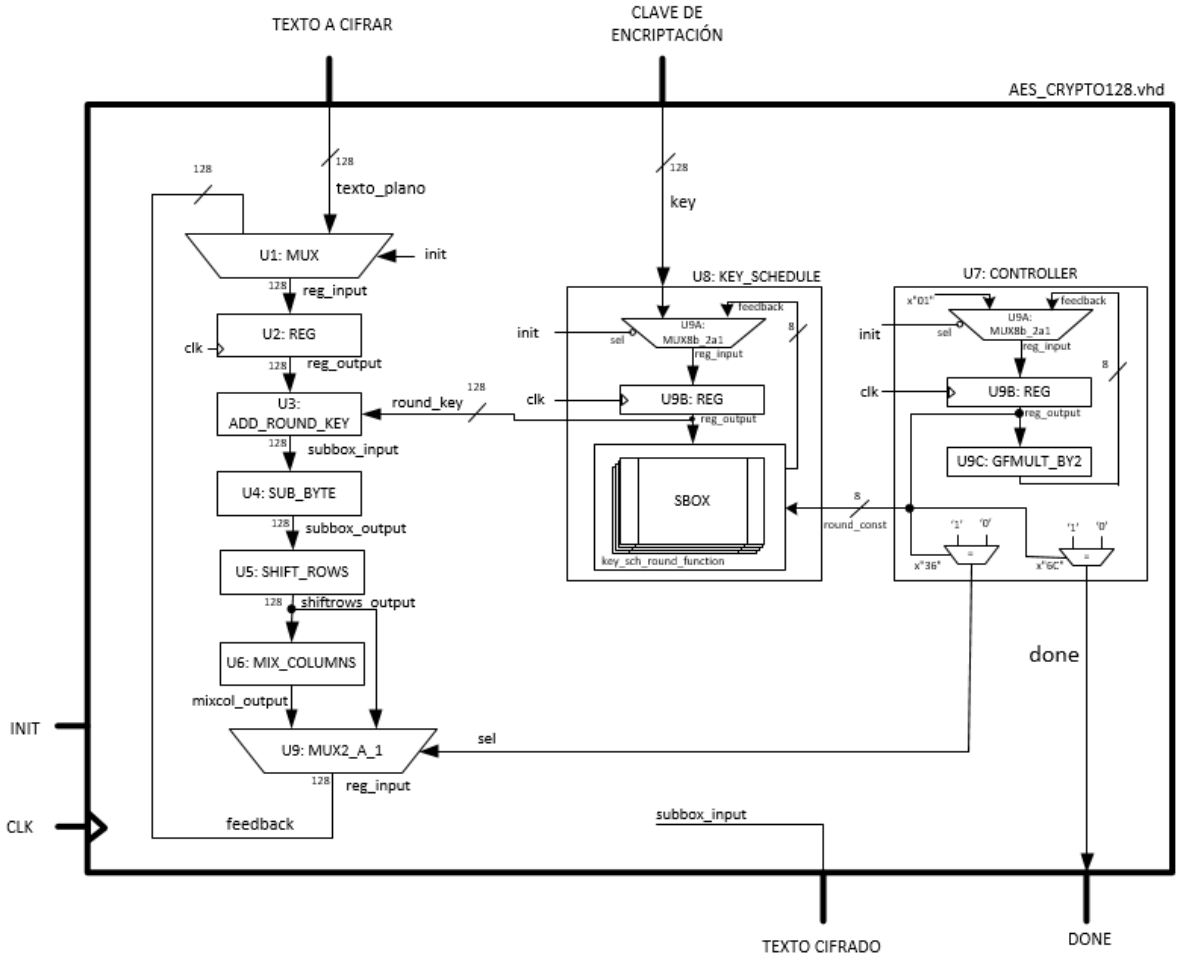
CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

3.1. Examinación de algoritmos de encriptación AES basados en VHDL

Estudiando la norma y revisando algoritmos de encriptación sugeridos por desarrolladores en VHDL de diversos portales de divulgación como GitHub y OpenCores para propósito del desarrollo de esta investigación se analizaron códigos que se acerquen a la descripción modular del estándar para de manera lo más didáctica posible desarrollar un algoritmo de encriptación basado en VHDL que pudiere ser reproducible en diversas plataformas de prueba.

Figura 18

Descripción teórica de los bloques del Algoritmo de Encriptación AES128



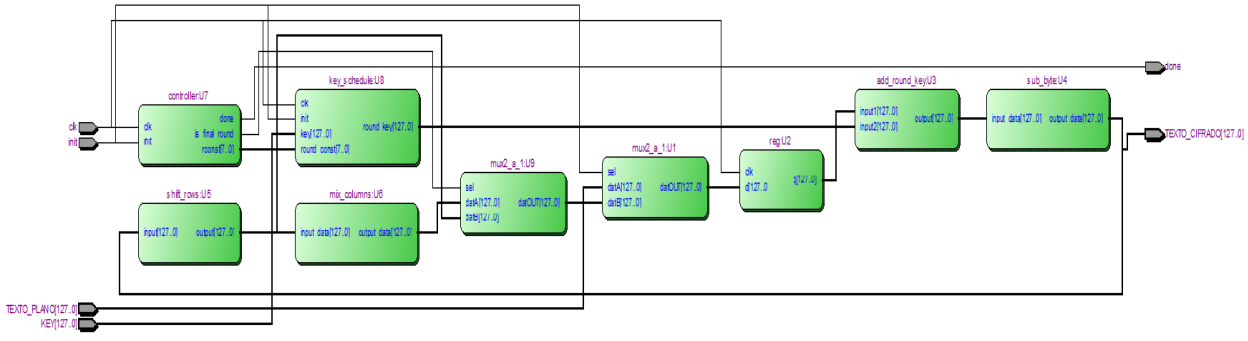
Uno de los primeros escenarios que se tuvo que considerar es que el algoritmo debe estar encaminado a ser implementado en una plataforma de desarrollo y que los diseños que generalmente se plantean en los repositorios visitados o están incompletos o son muy teóricos a la hora de pensar en su implementación, por lo que se debió considerar desarrollar módulos que permitan utilizar la menor cantidad de pines del FPGA posibles de manera que se pueda migrar el resultado de la encriptación a un sistema embebido, en ese contexto lo ideal es mantenernos a nivel de bit evitando el uso de protocolos de comunicación para reducir la necesidad de desarrollar módulos de sincronización según las velocidades de operación de los interfaces de comunicación existentes. En la figura 18 se observa la arquitectura base del algoritmo fundamentado en el estándar NIST FIPS197 el cual propone la utilización de una clave de cifrado sobre un texto plano a encriptar; ambos de 128 bits, mismos que son sometidos a una transformación no lineal invertible de cada uno de los bytes tanto de la palabra de estado (mensaje a encriptar) como de la clave de cifrado, basado en una tabla de sustitución denominada SBox misma que consiste en un arreglo de 16 x 16 bytes donde están contenidos todos los 256 caracteres ASCII, aunque esta tabla de sustitución puede ser modificada se debe considerar fija debido a que se ha diseñado para cumplir con propiedades criptográficas relevantes y su modificación pudiere comprometerlas y habilitar vulnerabilidades. No obstante, con el conocimiento y experiencia adecuadas se pudiere hacer un cambio en su configuración, lo que matemáticamente indicaría que considerando que cada una de las posiciones de la tabla; son 256 posibles combinaciones, puede contener cualquier carácter (esto sería cualquier valor de byte desde 0 hasta 255) nos daría la astronómica cifra de 256, por tanto $8578x10^{506}$ configuraciones posibles. Esta primera acción de sustitución aporta no linealidad, propiedad de confusión, resistencia a ataques diferenciales y ataques lineales. Luego de este primer procedimiento de sustitución se realiza un desplazamiento de las tres últimas filas de manera

cíclica cuyo desplazamiento depende del número de fila, una vez realizado esto el algoritmo es sometido a una mezcla de columnas, lo que implica una operación algebraica sobre la matriz de estado (lo que se viene modificando), lo cual es multiplicado con una matriz fija cuidando que el procedimiento matemático esté dentro del Campo de Galois denotado como GF(p^n). Este campo propone el uso de un polinomio conocido como polinomio irreducible de grado 8 [GF(2^8)], lo que implica que si dentro de la operación polinómica llegamos a obtener un polinomio de grado 8, este es sustituido por su solución basada en los postulados de Évariste Galois cuyo polinomio de grado 8 se reemplazaría con: $x^4 + x^3 + x + 1$, este procedimiento para el algoritmo AES128 se realiza 9 veces, en la décima ronda no se efectúa la mezcla de columnas sino que el resultado de este procedimiento durante las 10 rondas en que se efectúa el mismo procedimiento anterior es sometido a una combinación con la clave de ronda mediante una operación OR exclusiva bit a bit.

En la figura 19 podemos observar la descripción de transferencia de registros para el módulo de encriptación desarrollado para la presente propuesta donde podemos evidenciar el desarrollo de la codificación que equipara el esquema teórico propuesto en la figura anterior ya descrito en VHDL.

Figura 19

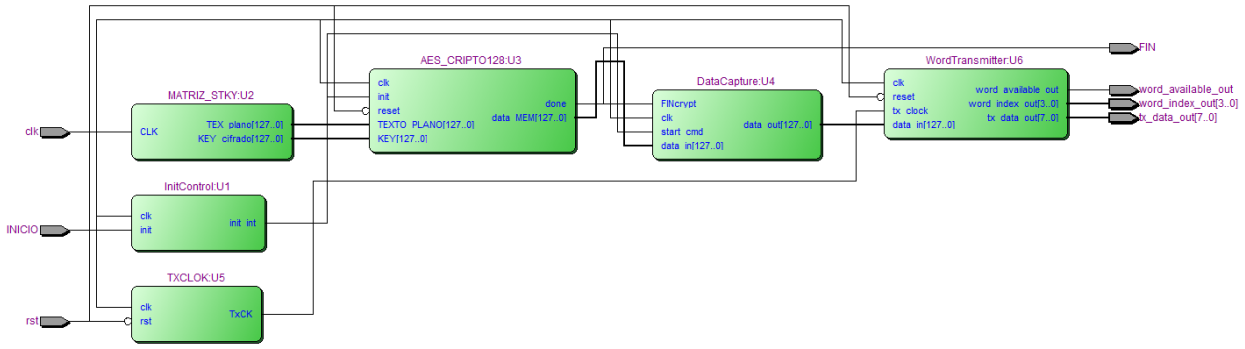
Descripción RTL de módulo de encriptación AES_CRYPTO128.vhd



En la figura 20 se observan los módulos que corresponden al desarrollo completo propuesto para esta investigación, esta implementación está comprendida por 6 módulos, mismos que tienen sub-módulos en su interior de acuerdo a la función que cumple. Cada uno de los módulos cumple una función específica que argumenta la operación del módulo de encriptación AES128, la descripción general de los módulos la tenemos a continuación:

Figura 20

Descripción RTL de AES_ENCRYPTION_SYSTEM



- **U1: InitControl**, es el encargado de dar la orden de inicialización para el proceso de encriptación; este módulo recibirá a futuro el comando de inicialización gestionado por un sistema embebido a manera de bit desde un puerto GPIO.
- **U2: MATRIZ_STKY**, este módulo constituye un banco de memorias donde se encuentran registrados la palabra de estado o texto plano que va a ser encriptado y la respectiva clave de cifrado. Este módulo debe ser modificado para recibir los vectores tanto de texto plano y llave de cifrado de manera externa desde un sistema embebido, aquí hay que considerar que se utilice la menor cantidad de pines posibles. La propuesta sería que se construyan dos puertos digitales uno para dirección y otro para dato además de un bit adicional para selección con lo que se pueden actualizar las memorias de estado y de

llave de cifrado de manera multiplexada con el bit de selección reduciendo el número de canales a utilizar en los puertos GPIO del sistema embebido y la FPGA.

- **U3: AES_CRIPTO128**, este módulo desarrolla el procedimiento de encriptación basado en el estándar FIPS 197.
- **U4: DATACapture**, este módulo está constituido por una máquina de estados que audita el cumplimiento de las rondas de encriptación del algoritmo para que cuando la bandera de finalización de ronda y bandera de finalización de proceso de encriptación aparezcan consecutivamente se tome la palabra de 128 bits correspondientes a la palabra encriptada a partir de los datos inicialmente ingresados.
- **U5: TXTCLOK**, considerando que las plataformas de desarrollo FPGA utilizadas para esta investigación pueden trabajar con relojes de hasta 100 MHz se propone la existencia de este módulo que construye un nuevo reloj para que el proceso de transmisión de la palabra encriptada pueda ser tomado a otra velocidad considerando se va a fragmentar la palabra para propósitos de reducción de número de pines a utilizar, así mismo esto deja abierta la opción para que a través de este módulo se pueda implementar el reloj externo de algún protocolo de comunicación particular que desee utilizar, tal como un protocolo de comunicación serial mismo que puede ser configurado en diferentes tasas de transmisión de bits.
- **U6: WordTransmitter**, este módulo constituye una máquina de estados que fragmenta la palabra encriptada de 128 bits para transmitir 16 palabras de 8 bits de manera ordenada, para lograr esto es habilitó un puerto de 4 bits adicional al puerto de 8 bits de manera que este bloque provee al mundo exterior el carácter los 16 caracteres encriptados de manera ordenada y a través del puerto de 4 bits se envía al mismo tiempo el número de la palabra que se encuentra en el puerto. Todo esto pensado para que

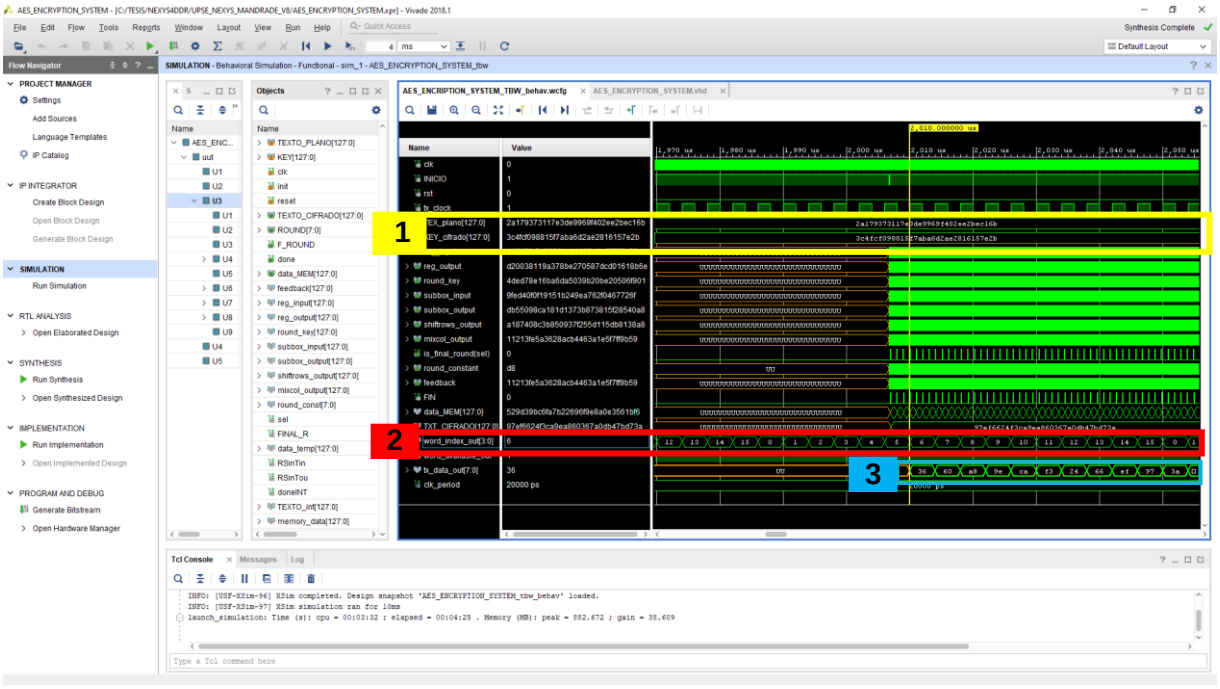
pueda ser tomada esta información en un puerto GPIO de un sistema embebido y sea reconstruida la palabra de 128 bits a través de una llave de desencriptación que puede ser desarrollada al igual que el propósito de esta investigación; es decir, en FPGA o a nivel de software dentro del sistema embebido.

3.2. Sintetización y Evaluación de algoritmo de encriptación AES en VHDL

La figura 21 muestra la evaluación del comportamiento del algoritmo de encriptación AES128 sometido a la herramienta de desarrollo XSim del programa Vivado 2018.1 en la que se puede evidenciar que ante los datos de prueba que dispone el banco de pruebas de la norma FIPS127 el texto cifrado corresponde a lo que se espera de la ejecución del algoritmo. Luego de ello podemos evidenciar que la palabra digital de 128 bits ha sido fragmentada en 16 palabras de 8 bits con el propósito de transmitirla a un sistema embebido mediante pocos pines GPIO donde se podrá obtener la palabra encriptada, el índice de ésta y un flag de indicación de que la palabra está disponible en el puerto.

Figura 21

Evaluación funcional del algoritmo de encriptación AES128



En la gráfica anterior podemos observar en el recuadro 1 las palabras de 128bit (descritas en hexadecimal) tanto para el texto plano como de la llave de cifrado, en el recuadro 2 podemos observar la bandera de señalización que corresponde al número de palabra que se está transmitiendo en ese mismo instante en el recuadro 3. En el recuadro 3 se puede observar ya la segmentación por octetos del texto cifrado con el objeto de poder ser transmitidos a un sistema embebido tipo Raspberry o Jetson nano a nivel de bit de manera de utilizar la menor cantidad de pines posibles. Se observa la palabra 6 a la que le corresponde el valor 36, la palabra 7 a la que le corresponde el valor 60 y la palabra 8 a la que le corresponde el valor A8 dentro del texto cifrado han sido ya obtenidos, a nivel simulación y correspondería que el vector de la palabra encriptada (8 bits) puedan ser ingresados a 8 pines GPIO de un sistema embebido y como señalización de lo que está ingresando al sistema embebido tendríamos al número de índice de palabra lo cual representarían 4 bits adicionales. Esto significa que con 12 bits o lo que es lo mismo 12 pines GPIO de un sistema embebido podríamos obtener cada transmisión de byte encriptado y por cada 16 transmisiones podríamos recomponer el texto encriptado completo.

A continuación, realizaremos la transcripción de los resultados obtenidos por cada una de las herramientas software de diseño y sintetización de comportamiento digital aplicado a cada uno de los circuitos integrados que se encuentran disponibles en las diferentes tarjetas de entrenamiento dispuestas para esta investigación. Iniciaremos listando los perfiles de los componentes sobre los que se realizará el análisis:

Tabla 3

Dispositivos FPGA utilizados para la prueba

TARJETA DE ENTRENAMIENTO	CIRCUITO INTEGRADO FPGA	FAMILIA FPGA	SOFTWARE UTILIZADO
Nexys4 DDR	XC7A100T-1CSG324C	Artix-7	Vivado 2018.1
CmodA7	XC7A15T-1CPG236C	Artix-7	Vivado 2018.1
DE10-Nano	5CSEBA6U23I7NDK	Cyclone V SE	Quartus II v 13.0
DE0-Nano	EP4CE22F17C6N	Cyclone IV CE	Quartus II v 13.0

Una vez realizada la codificación, síntesis e implementación de código VHDL procedemos a hacer un levantamiento de los reportes generados por las herramientas Vivado 2018.1 sobre el código implementado en la tarjeta Nexys4-DDR, cuyos resultados podemos apreciar en la tabla 4, donde se han transcrito los recursos utilizados por el algoritmo de manera jerárquica indicando lo que correspondió en uso de recursos a cada uno de los módulos que componen en diseño del Sistema de Encriptación AES128 diseñado para esta investigación.

Tabla 4

Reporte de implementación por jerarquía en Nexys4-DDR

NEXYS4-DDR	REPORTE DE IMPLEMENTACIÓN POR JERARQUÍA									
RECURSOS	Slice LUTs	Slice Registers	F7 Muxes	F8 Muxes	Slice	LUT as Logic	LUT FlipFlop Pairs	Block RAM Tile	Bonded IOB	BUFGCTRL
Disponibles	63400	126800	31700	15850	15850	63400	63400	135	210	32
AES_ENCRYPTION_SYSTEM	1030	818	272	136	349	1030	175	1	17	1
U1 (Init Control)	4	6	0	0	3	4	3	0	0	0
U5 (TxClock)	10	9	0	0	3	10	5	0	0	0
U6 (WordTransmitter)	42	280	16	8	65	42	16	0	0	0
U4 (DataCapture)	46	130	0	0	41	46	2	0	0	0
U3 (AES_CRIPTO128)	928	393	256	128	259	928	149	1	0	0

En la tabla 5 se han transcrito los resultados obtenidos por la misma herramienta Vivado 2018.1 sobre el módulo Cmod-A7, en esta tabla se han transcrito los resultados obtenidos por la herramienta de acuerdo a lo que cada componente del diseño VHDL del proyecto requirió.

Tabla 5

Reporte de implementación por jerarquía en Cmod-A7

ARTIX7 - CMOD	REPORTE DE IMPLEMENTACIÓN POR JERARQUÍA									
RECURSOS	Slice LUTs	Slice Registers	F7 Muxes	F8 Muxes	Slice	LUT as Logic	LUT FlipFlop Pairs	Block RAM Tile	Bonded IOB	BUFGCTRL
Disponibles	10400	20800	16300	8150	8150	10400	10400	25	106	32
AES_ENCRYPTION_SYSTEM	1019	817	272	136	331	1019	178	1	17	1
U1 (Init Control)	4	6	0	0	3	4	3	0	0	0
U3 (AES_CRIPTO128)	918	393	256	128	254	918	152	1	0	0
U4 (DataCapture)	46	130	0	0	53	46	2	0	0	0
U5 (TxClock)	9	8	0	0	3	9	5	0	0	0
U6 (WordTransmitter)	42	280	16	8	63	42	16	0	0	0

Utilizando la herramienta de diseño Quartus II v13.0 sobre la tarjeta de prototipado DE10-nano se obtuvieron los resultados transcritos en la tabla 6, donde se puede apreciar los recursos utilizados por el diseño de la propuesta de investigación sobre dispositivos Intel.

Tabla 6

Reporte de implementación por jerarquía en DE10-Nano

DE10-nano	REPORTE DE IMPLEMENTACIÓN POR JERARQUÍA					
RECURSOS	LC Combinatorials	LC Registers	Block Memory Bits	DSP Blocks	Pins	Virtual Pins
Disponibles	110000	166036	5570	112		
AES_ENCRYPTION_SYSTEM	1354	819	0	0	17	0
U3 (AES_CRIPTO128)	1225	393	0	0	0	0
U4 (DataCapture)	5	132	0	0	0	0
U1 (Init Control)	6	6	0	0	0	0
U5 (TxClock)	11	8	0	0	0	0
U6 (WordTransmitter)	106	280	0	0	0	0

Es importante considerar que los recursos entre dispositivos de diferentes fabricantes difieren en la concepción tecnológica utilizada por cada uno; no obstante, la tecnología FPGA requiere del empleo de lenguaje de descripción de hardware (VHDL) para la descripción del diseño y la propuesta de esta investigación se la ha realizado de manera que el código diseñado sea completamente transportable; es decir, prescindiendo de los recursos especializados que

tiene cada circuito integrado con el objeto de que no sean requeridas modificaciones mayores al código previo a su implementación. En la tabla 7 podemos observar los resultados obtenidos de la implementación del algoritmo sobre la tarjeta de prototipado DE0-nano.

Tabla 7

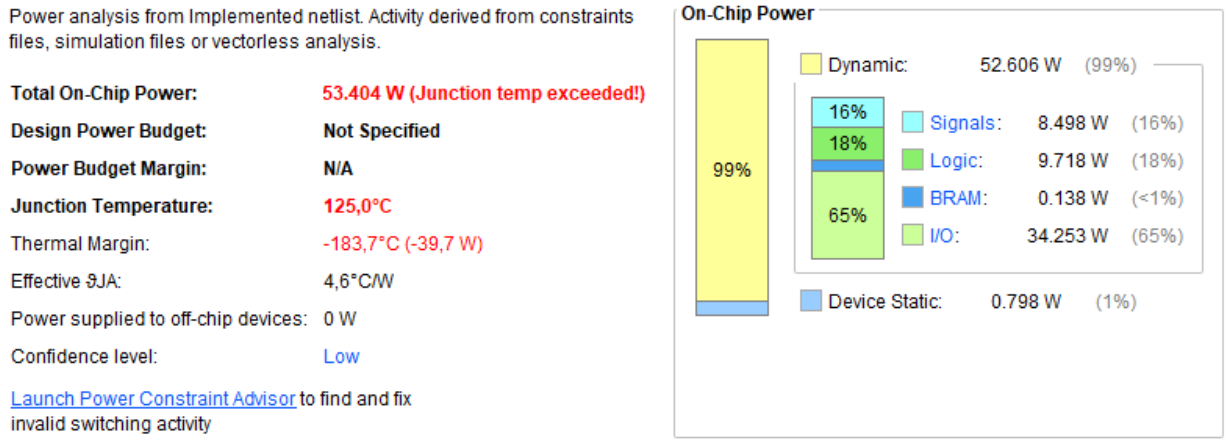
Reporte de implementación por jerarquía en DE0-Nano

DE0-nano	REPORTE DE IMPLEMENTACIÓN POR JERARQUÍA							
RECURSOS	LC Combinatorials	LC Registers	Block Memory Bits	DSP Blocks	DSP 9x9	DSP 18x18	Pins	Virtual Pins
Disponibles								
AES_ENCRYPTION_SYSTEM	5017	819	0	0	0	0	17	0
U3 (AES_CRIPTO128)	4808	393	0	0	0	0	0	0
U4 (DataCapture)	8	132	0	0	0	0	0	0
U1 (Init Control)	6	6	0	0	0	0	0	0
U5 (TxClock)	16	6	0	0	0	0	0	0
U6 (WordTransmitter)	178	280	0	0	0	0	0	0

El Software Vivado V2018.1 en función del diseño del proyecto activo admite con la licencia disponible la generación de reporte de consumo de energía de manera que se obtuvieron las proyecciones de disipación de calor del diseño, en la figura 22 se pueden observar los resultados obtenidos para el circuito integrado FPGA instalado en la tarjeta Nexys4-DDR.

Figura 22

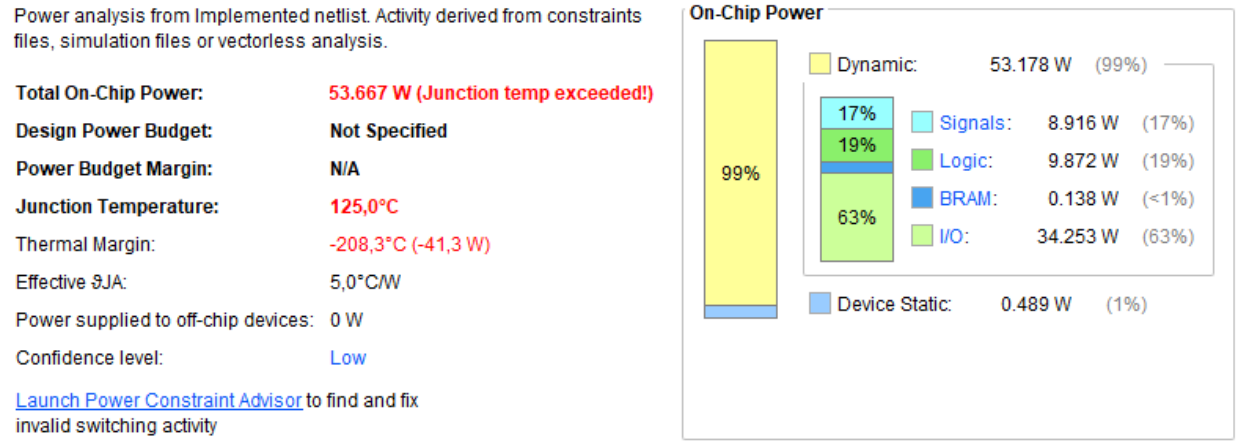
Análisis de consumo de energía en Nexys4-DDR



En la figura 23 podemos observar las proyecciones de generación de calor debido al trabajo realizado por el algoritmo en el FPGA instalado en el módulo Cmod-A7.

Figura 23

Análisis de consumo de energía en Cmod-A7



Otros reportes generados por las herramientas que se circunscriben a los mismos proyectos, en los mismos circuitos integrados muestran algunos datos adicionales y a la vez consistentes con el reporte por jerarquía, motivo por el cual son considerados también para efectos de la evaluación propuesta en esta investigación. En las tablas 8-11 se muestran los resúmenes de los recursos utilizados por el diseño en el FPGA transcritos desde Vivado 2018.1 y Quartus II V13.0.

Tabla 8

Resumen de implementación en Nexys4-DDR

NEXYS4-DDR	**RESUMEN**		
Recurso	Utilización	Disponible	% Utilizacion
LUT	1030	63400	1,62%
FF	818	126800	0,65%
BRAM	1	135	0,74%
IO	17	210	8,10%

Tabla 9

Resumen de implementación en DE0-Nano

DE0-nano		**RESUMEN**	
Recurso	Utilización	Disponible	% Utilizacion
Total logic elements	5386	22320	24,13%
Combinational with no register	4567		
Register only	369		
Combinational with a register	450		
Total combinational functions			
Logic element usage by number of			
LUT inputs			
4 input functions	4687		
3 input functions	98		
<= 2 input functions	232		
Register only	369		
Logic elements by mode	5005		
normal mode	12		
arithmetic mode			
Total registers	819	23018	3,56%
Dedicated logic registers	819	22320	3,67%
I/O registers	0	698	0,00%
Total LABs: partially or completely use	358	23018	1,56%
Virtual pins	0		
I/O pins	17	154	11,04%
Clock pins	2	7	28,57%
Dedicated input pins	0	9	0,00%
Global signals	2		
M9Ks	0	66	0,00%
Total block memory bits	0	608256	0,00%
Total block memory implementation bi	0	608256	0,00%
Embedded Multiplier 9-bit elements	0	132	0,00%
PLLs	0	4	0,00%
Global clocks	2	20	10,00%
JTAGs	0	1	0,00%
CRC blocks	0	1	0,00%
ASMI blocks	0	1	0,00%
Impedance control blocks	0	4	0,00%
Average interconnect usage (total/H/V	6%	6%	6,00%
Peak interconnect usage (total/H/V)	40%	37%	44,00%
Maximum fan-out	795		
Highest non-global fan-out	265		
Total fan-out	22008		
Average fan-out	3.55		

Tabla 10

Resumen de implementación en Cmod-A7

ARTIX7 - CMOD	**RESUMEN**		
Recurso	Utilización	Disponible	% Utilizacion
LUT	1019	10400	9,80%
FF	817	20800	3,93%
BRAM	1	25	4,00%
IO	17	106	16,04%

Tabla 11

Resumen de implementación en DE10-Nano

RESUMEN DE RECURSOS UTILIZADOS	USADO	DISPONIBLE
Logic utilization (ALMs needed / total ALMs on device)	1158	41910
ALMs needed [=A-B+C]	1158	
[A] ALMs used in final placement [=a+b+c+d]	1288	41910
[a] ALMs used for LUT logic and registers	192	
[b] ALMs used for LUT logic	881	
[c] ALMs used for registers	215	
[d] ALMs used for memory (up to half of total ALMs)	0	
[B] Estimate of ALMs recoverable by dense packing	140	41910
[C] Estimate of ALMs unavailable [=a+b+c+d]	10	41910
[a] Due to location constrained logic	0	
[b] Due to LAB-wide signal conflicts	1	
[c] Due to LAB input limits	9	
[d] Due to virtual I/Os	0	
Difficulty packing design	Low	
Total LABs: partially or completely used	152	4191
Logic LABs	152	
Memory LABs (up to half of total LABs)	0	
Combinational ALUT usage for logic	1355	
7 input functions	40	
6 input functions	913	
5 input functions	126	
4 input functions	62	
<= 3 input functions	214	
Combinational ALUT usage for route-throughs	118	
Dedicated logic registers	893	
By type:		
Primary logic registers	814	839820
Secondary logic registers	79	83820
By function:		
Design implementation registers	819	
Routing optimization registers	74	
Virtual pins	0	
I/O pins		
Clock pins	17	314
Dedicated input pins	2	8

3.3. Evaluación del rendimiento y uso de recursos

Con el objeto de evaluar apropiadamente el uso de los recursos debemos hacer primero una comparación entre las tecnologías que se están validando dado que las definiciones y diseños entre un fabricante y otro, incluso dentro del mismo fabricante y sus diversas familias de FPGA pudieren tener conceptualmente diferencias en diseño sin embargo pudiere existir alguna semejanza operacional de cara a la construcción de comportamiento de un mismo diseño VHDL. La tabla 12 se ha preparado con el objeto de abordar estas equivalencias entre términos.

Tabla 12

Equivalencia entre términos utilizados por Intel y AMD

	VIVADO	QUARTUS	DESCRIPCIÓN
SLICE LUTs	LUTs	LE (Logic Elements) o ALUTs (Adaptive Logic)	Tablas de búsqueda para implementar funciones lógicas
SLICE Registers	Flip-Flops	LE (Logic Elements) o Dedicated Logic Registers	Registros (FF) utilizados en el diseño.
F7 Muxes **F8 Muxes**	Número de multiplexores F7 y F8	No hay equivalencia (los MUXs están implícitos dentro de la lógica)	Multiplpexores usados dentro de Slices en el diseño.
Slices	Número de Slices utilizados	No hay equivalencia (Los Slices son una analogía de Logic Array Blocks (LABs)	Bloques que contienen LUTs y Registros
LUT as Logic	LUTs utilizados como lógica	Incluidos en ALUTs o Logic Elements	Utilización de LUTs para implementación específica de lógica digital (no memorias, etc)
LUT Flip-Flop pairs	Número de parejas de LUT y Flip-Flops	Logic Elements	LUTs y Flip-Flop que están emparejadas
Block RAM Tile	Reporta el número de bloques de RAM utilizados	Equivalente a M9 Block o M20 Blocks según la generación del FPGA	Memoria Embebida en el FPGA
Bonded IOB	Número de IOBs utilizados	Número de IOBs utilizados	Bloques que gestionan los pines como entrada/salida conectados
BUFGCTRL	Número de Buffers Globales utilizados	Global Clock Buffers Global Clock Control Blocks	Buffers de reloj global

La tabla 13 nos muestra los recursos utilizados por los diversos FPGAs considerando los elementos que tienen funcionalidad equivalente.

Tabla 13

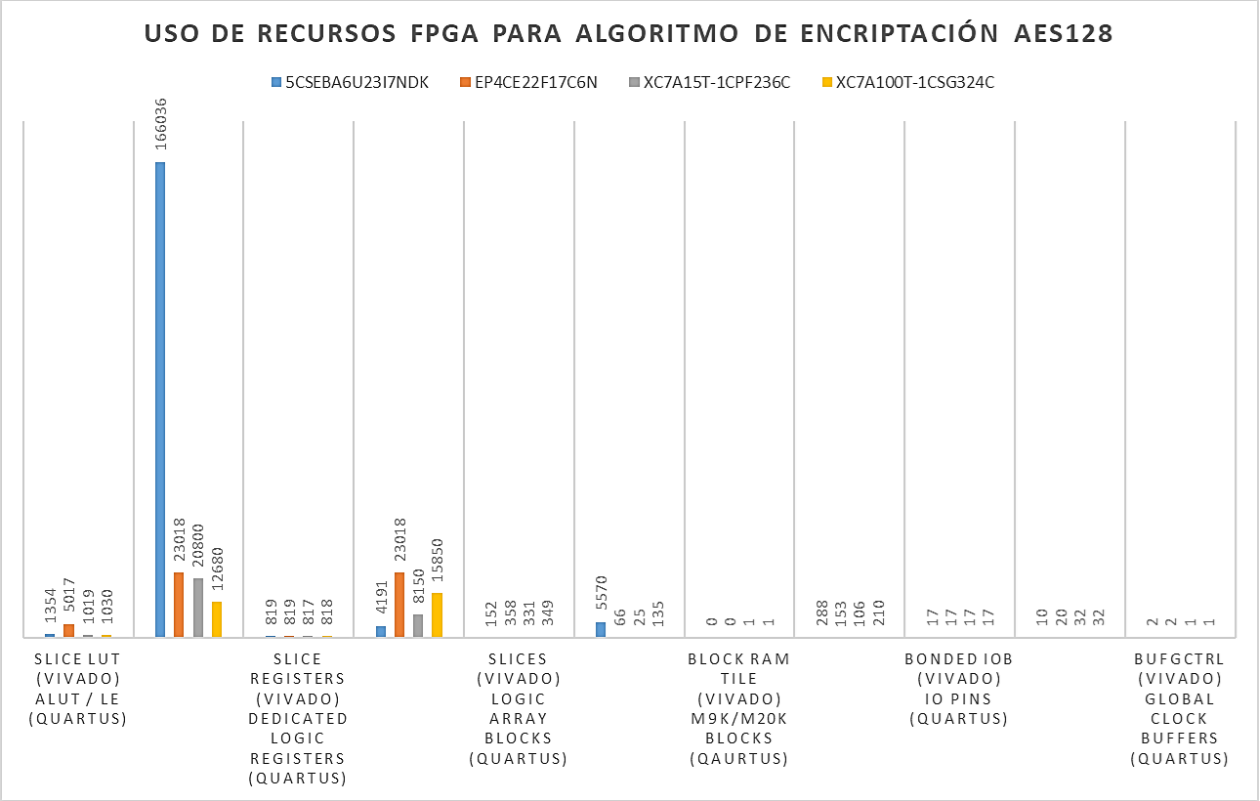
Análisis comparativo de recursos utilizados

	DE10-Nano **5CSEBA6U23I7NDK**	DE0-Nano **EP4CE22F17C6N**	Cmod A7 **XC7A15T-1CPF236C**	Nexys4-DDR **XC7A100T-1CSG324C**
Slice LUT (Vivado) ALUT / LE (Quartus)	1354	5017	1019	1030
Slice Registers (Vivado) Dedicated Logic Registers (Quartus)	819	819	817	818
Slices (Vivado) Logic Array Blocks (Quartus)	152	358	331	349
Block RAM Tile (Vivado) M9K/M20K Blocks (Qaurtus)	0	0	1	1
Bonded IOB (Vivado) IO Pins (Quartus)	17	17	17	17
BUFGCTRL (Vivado) Global Clock Buffers (Quartus)	2	2	1	1

Esta tabla, así como la figura 24 nos muestran un resumen de recursos utilizados con la implementación del algoritmo de encriptación AES128 preparado para esta investigación, se debe tomar en consideración que la comparación efectuada refleja información inherente tan solo a la cantidad de elementos utilizados en cada dispositivo y que, desde el enfoque de las diferencias existentes en las tecnologías de cada circuito integrado propuestas por los fabricantes no podríamos decir aun en este punto que algún análisis pudiere ser determinante o concluyente todavía debido a que cada dispositivo ha sido desarrollado con procesos tecnológicos diferentes y cuentan con una cantidad de recursos diversos entre sí, por lo tanto no pueden ser comparados tan superficialmente. No obstante, esta comparación ya nos va arrojando información relevante respecto de la forma como se utilizan recursos, especialmente las LUT para sintetizar comportamiento digital aplicando el mismo algoritmo en las diversas familias de circuitos integrados seleccionados para la prueba.

Figura 24

Uso de recursos FPGA para AES128

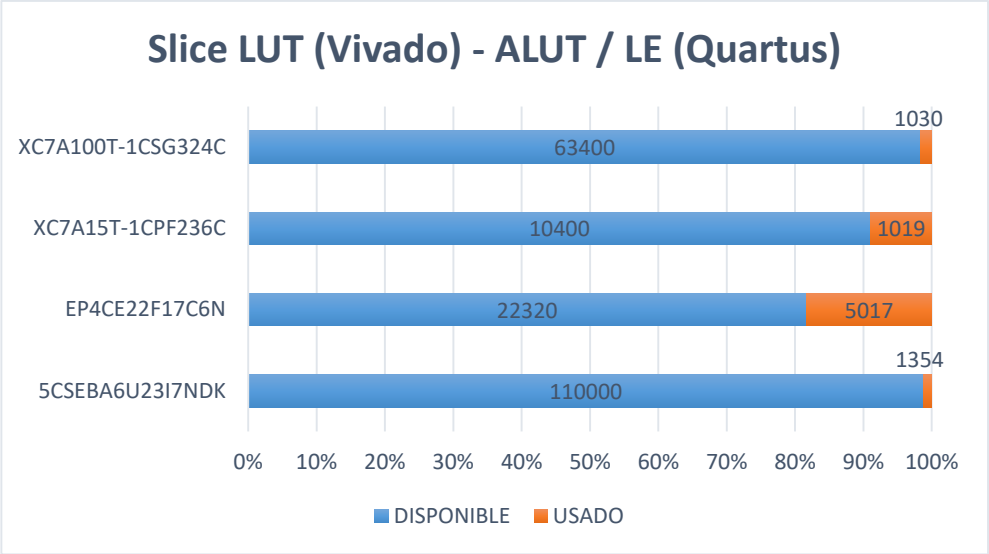


La figura 25 nos revela el nivel de utilización de los elementos Slice LUT (ALUT), los que sometidos a la implementación del mismo algoritmo requieren del 1% de este recurso en el circuito integrado 5CSEBA6U23I7NDK (Cyclone V), del 22% de este recurso en el circuito integrado EP4CE22F17C6N (Cyclone IV), del 10% de este recurso en el circuito integrado XC7A15T-1CPF236C (Artix7, 15T) y del 2% de este recurso en el circuito integrado XC7A100T-1CSG324C (Artix7, 100T). Estos datos son relevantes ya que nos dan cuenta del bajo nivel de utilización de este recurso una vez aplicado el algoritmo en el circuito integrado Cyclone V lo que sugiere que este dispositivo tiene una gran capacidad de Slice LUT disponibles con lo que podríamos interpretar que este dispositivo puede ser considerado altamente eficiente y con un

amplio margen de maniobra para la implementación de más requerimientos de circuitos dentro del mismo encapsulado.

Figura 25

Nivel de utilización de Slice LUT / ALUT vs disponibilidad



Respecto de la utilización del 22% de este recurso en el circuito integrado Cyclone IV sugiere que este dispositivo es menos eficiente ya que compromete la disponibilidad de Slices LUT para el desarrollo de tecnología complementaria que se requiera dentro del FPGA además del algoritmo de encriptación. El 10% de utilización de este recurso en el Artix7, 15T revela que, aunque no aparenta ser tan eficiente como el Cyclone V tiene mejores prestaciones para la implementación de lógica adicional dentro de su encapsulado, finalmente el circuito integrado Artix7, 100T presenta las mejores prestaciones posibles de cara a la implementación del algoritmo de encriptación AES128 debido a que cuenta a su favor un 98% de disponibilidad de este recurso para la implementación de lógica adicional.

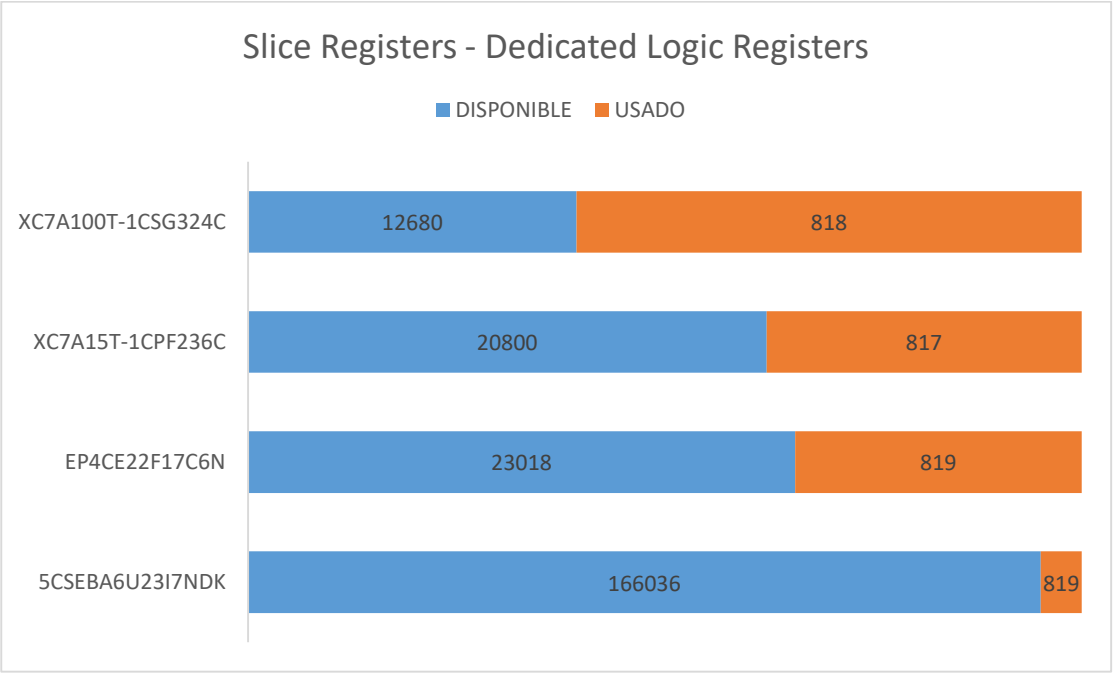
En este contexto se evidencia que tanto el circuito integrado Cyclone V y el Artix7, 100T son los más eficientes para este algoritmo, no obstante, se abre un nuevo debate respecto del

costo en función del rendimiento y necesidades de inserción o no de más código en el circuito integrado a elegir.

Los datos que revela la figura 26 dan cuenta del nivel de utilización de Slices con registros para la implementación del algoritmo de encriptación AES128, dado que el diseño requiere de la implementación de sendos registros para capturas de tramas de 128 bits este recurso es esencial y podría ser determinante a la hora de elegir el componente adecuado para el desarrollo de un producto de este tipo a larga escala. Respecto de los recursos disponibles en cada circuito integrado y los requeridos para la implementación de este algoritmo podemos indicar que se requieren del 0.493% de este recurso en el circuito integrado 5CSEBA6U23I7NDK (Cyclone V), del 3.558% de este recurso en el circuito integrado EP4CE22F17C6N (Cyclone IV), del 3.928% de este recurso en el circuito integrado XC7A15T-1CPF236C (Artix7, 15T) y del 6.451% de este recurso en el circuito integrado XC7A100T-1CSG324C (Artix7, 100T).

Figura 26

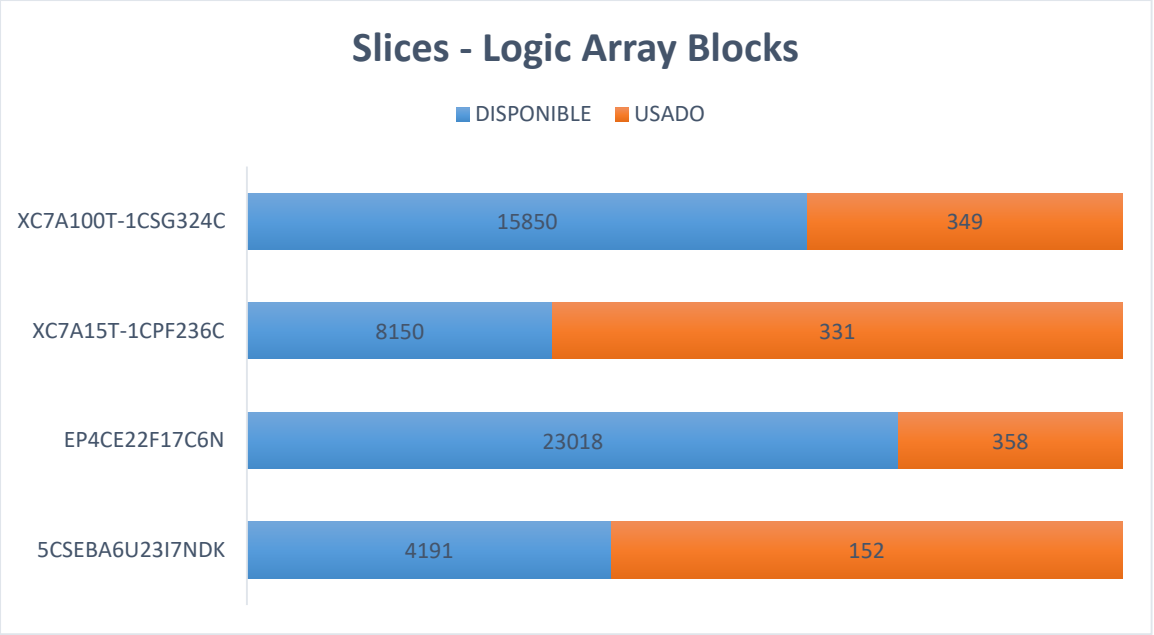
Nivel de utilización de Slice Registers / DLR vs disponibilidad



Es relevante indicar que considerando la infraestructura tecnológica que maneja el circuito integrado Cyclone V lo hace más adecuado para la implementación de este algoritmo debido a que aparenta una mejor eficiencia en la construcción de algoritmos que requieren una gran cantidad de registros y, aun así; en este caso, está en capacidad de implementar mucha más lógica.

Figura 27

Nivel de utilización de Slice / LAB vs disponibilidad



Respecto de los recursos Slices (Logic Array Blocks) disponibles en cada circuito integrado y los requeridos para la implementación mostrados en la figura 27 podemos indicar que se requieren del 3.627% de este recurso en el circuito integrado 5CSEBA6U23I7NDK (Cyclone V), del 1.555% de este recurso en el circuito integrado EP4CE22F17C6N (Cyclone IV), del 4.061% de este recurso en el circuito integrado XC7A15T-1CPF236C (Artix7, 15T) y del 2.202% de este recurso en el circuito integrado XC7A100T-1CSG324C (Artix7, 100T).

Bajo este análisis podríamos indicar que el circuito integrado Cyclon IV es el apropiado respecto del manejo del recurso Slice (LAB) y que su relación uso vs disponibilidad le dan una gran ventaja respecto de los otros dispositivos; no obstante, el Artix7, 100T está muy cerca del rendimiento, pero la cantidad de recursos disponibles es ligeramente inferior que el Cyclone IV.

Estos tres análisis son los más relevantes de cara a determinar la eficiencia en el uso de los recursos entre estos dispositivos FPGA para la implementación de un algoritmo de encriptación AES128. El diseño propuesto no contempla el uso de memorias dedicadas por tanto se hace irrelevante el análisis de este parámetro, así como también se debe considerar que se ha utilizado el mismo código para todas las pruebas en las diferentes plataformas en consecuencia todas indican utilizar el mismo número de pines por tanto se hace irrelevante su análisis, además respecto de los usos de buffers de reloj globales es pues innecesario el análisis a fondo del impacto que ocasiona el algoritmo a cada componente por tanto se hace innecesario su análisis detallado.

Sin embargo, es importante señalar también que de acuerdo al análisis de temperatura ambos circuitos integrados AMD consideran que la temperatura de la juntura estará cerca de los 53Watts, estando este margen de temperatura por encima de lo sugerido por tanto se debe prestar atención en los ajustes que pudieren hacerse a la codificación o a las reglas de sintetización de manera de mejorar la eficiencia al respecto.

Del análisis efectuado se puede concluir que tanto la tecnología aplicada en el desarrollo del circuito integrado 5CSEBA6U23I7NDK y en el XC7A100T-1CSG324C tienen grandes prestaciones para el desarrollo de un algoritmo que contempla un gran número de iteraciones, uso de registros y computo de grandes cantidades de dato. No obstante se debe considerar situaciones adicionales como el costo beneficio debido a que el uso de un dispositivo de gran capacidad sin tener la necesidad de hacerlo pudiere generar costos innecesarios ya que estos

circuitos integrados FPGA requieren de alimentación de buena calidad, un oscilador estable, generalmente el uso de un dispositivo de memoria externo para almacenar el archivo de configuración y un puerto de carga JTAG para su configuración inicial o actualizaciones futuras posibles. Dicho esto, los resultados de este análisis sugieren que los dispositivos desarrollados por AMD [XC7A15T-1CPF236C y XC7A100T-1CSG324C] presentan excelentes prestaciones y muy buen manejo de recursos si el desarrollo del producto se lo requiere hacer a nivel de lógica directa reduciendo al máximo el uso de los recursos; es decir, el uso de LUT y/o multiplexores para este algoritmo en particular de manera de poder inferir más comportamiento asociado al algoritmo de encriptación dentro del mismo FPGA.

Este estudio revelo datos comparativos que aportan importante información respecto de la eficiencia de los dispositivos FPGA y su adaptación al desarrollo de algoritmos complejos. Debido a que los circuitos integrados de la familia Cyclone V y Artix7-100T son de características superiores muestran un rendimiento más balanceado y es destacable su eficiencia y cantidad de recursos disponibles post construcción del algoritmo de encriptación. Sin embargo debido a su tamaño y prestaciones en la construcción de este algoritmo pudiere ser una extraordinaria opción el uso del circuito integrado XC7A15T-1CPF236C debido a su versatilidad si lo que se requiere es construir un módulo de encriptación sin mayores necesidades de inserción de lógica adicional dentro del mismo componente.

CONCLUSIONES

La evaluación de un algoritmo de encriptación AES mediante VHDL se llevó a efecto bajo ambiente controlado para lo cual debió desarrollarse una codificación que describa apropiadamente la norma que lo sustenta. Es menester considerar que se cumplió el objetivo al evidenciar la eficiencia del algoritmo en hardware FPGA de cara cubrir la necesidad de incrementar una capa de seguridad sobre las comunicaciones digitales establecidas en ambiente operacional por unidades militares mismas que no pueden estar sujetas a protocolos de aseguramiento de la información de terceros.

Se examinó la norma FIPS 197 para AES128 y se analizaron sendos repositorios que proponen algoritmos de encriptación; no obstante, las técnicas de codificación utilizadas o la inadecuada conceptualización abordada revelaron la imposibilidad de su utilización para propósitos de análisis, por lo que se desarrolló un proyecto nuevo basado en una licencia GNU de Hosein Hadipour del que se utilizó el modelo conceptual para abordar el procedimiento de encriptación amparado en la norma.

Se sintetizó el algoritmo de encriptación en cuatro circuitos integrados FPGA diferentes utilizando dos herramientas de diseño y sintetización basado en VHDL, Quartus II V13.0 y Vivado 2018.1 de los que se pudo levantar la información relevante de uso de recursos que da sustento a este análisis. El uso de circuitos integrados FPGA aporta con una capa difícilmente reversible de seguridad a las comunicaciones a efectuarse mediante sistemas embebidos ya que el desarrollo de ingeniería inversa dificulta enormemente la intensión de romper las seguridades.

La evaluación del rendimiento y uso de recursos en diferentes plataformas de sintetización de hardware digital dejó información relevante respecto de la idoneidad de uno u otro circuito integrado FPGA, donde el costo beneficio se hacen determinantes a la hora de

seleccionar el dispositivo apropiado. Sin embargo, es relevante considerar que el ejercicio desarrollado en esta investigación propuso el desarrollo de un proyecto completamente escrito en VHDL sin opción de utilización de recursos especializados para que este algoritmo sea fácilmente transportable entre los diversos circuitos integrados, esto deja abierta la opción de mejoras futuras al algoritmo especializando algunos de sus bloques para obtener un rendimiento superior.

Desde el enfoque de la construcción de un dispositivo de encriptación que cumpla tan solo esa función y que se pueda integrar a un sistema embebido mediante conexión a nivel bit entre puertos de entrada – salida digital de propósito general una muy buena opción es el circuito integrado FPGA XC7A15T-1CPF236C [instalado en el módulo Cmod-A7], pero si el enfoque es que este algoritmo de encriptación se asocie con más recursos tecnológicos, entiéndase por ello más código VHDL dentro del mismo dispositivo una gran alternativa son en este orden los circuitos integrados XC7A100T-1CSG324C [instalado en la tarjeta de entrenamiento Nexys4-DDR] si de uso optimizado de recursos lógicos se requiere y el circuito integrado 5CSEBA6U23I7NDK [instalado en la tarjeta de entrenamiento DE10-Nano] si de optimización y alto uso de recursos de memoria embebida se refiere el diseño a implementar, tal como este algoritmo de encriptación.

RECOMENDACIONES

Se recomienda afirmar este estudio como línea de base para el desarrollo de soluciones basadas en la encriptación de datos digitales como soporte al desarrollo de equipos de uso militar, ya que su empleo permite incrementar una capa de seguridad nativa frente a los escenarios de enlaces de comunicaciones sensibles.

La evaluación de nuevos códigos permitirá mejorar la eficiencia en el uso de los recursos de los circuitos integrados FPGA y su inserción como parte de un sistema embebido a las más altas tasas de transmisión posibles. La evaluación de código fuente bajo licencia pública permitiría reducir la curva de aprendizaje y permite encaminar los esfuerzos hacia el desarrollo de nuevos productos. El estudio de la norma y el afianzamiento de los modelamientos matemáticos adscritos a ella, pero pensado en VHDL permitirá el desarrollo de algoritmos más eficientes y con mejores prestaciones.

La información relevante obtenida debe constituirse en la piedra angular de nuevos desarrollos tendientes a la sintetización de un algoritmo de encriptación cada vez más eficiente y con un apropiado uso de recursos.

REFERENCIAS

Divya, B. N., Davana, N., Dhanushree, K., Apoorva, V., & Anusha, V. (2021). Standard, Design and Implementation of Advanced Encryption. *Embedded-Systems and Signal-Processing* , 68-72.

Ferreira, A. C., & Silva, N. B. (2020). Comparison of secure communication with AES between embedded system and general purpose computer. *IEEE Symposium on Computers and Communications (ISCC).* Rennes, France: IEEE.

Gonzalez, M. S., Roldan, G., D'Angiolo, G., & Asteasuain, F. (2021). Implementación y Análisis de rendimiento de Algoritmos Criptográficos para aplicaciones en Sistemas Embebidos. *IV Simposio Argentino de Informática Industrial e Investigación Operativa (SIIIO 2021) - JAIIO 50 (Modalidad virtual).* Sociedad Argentina de Informática e Investigación Operativa. Obtenido de https://sedici.unlp.edu.ar/handle/10915/141782

Gupt, K. K., Kshirsagar, M., Sullivan, J. P., & Ryan, C. (2021). Automatic Test Case Generation for Prime Field Elliptic Curve Cryptographic Circuits. *17th International Colloquium on Signal Processing & Its Applications (CSPA).* Langkawi, Malaysia: IEEE. doi:978-0-7381-4397-2/21

IEEE. (2023 de 12 de 2019). *IEEE 1076-2019*. Obtenido de IEEE Standard for VHDL Language Reference Manual: https://standards.ieee.org/ieee/1076/5179/

Jimenez Santiago, L. (2023). *Evaluación de SOCs con procesadores RISC-V sobre FPGAs de bajo consumo.* Universidad Politécnica de Madrid.

Naiouf, M., De Giusti, A., De Giusti, L., Chichizola, F., Sanz, V., Pousa, A., . . . Bagsali, M. J. (2020). Fundamentos, algoritmos y evaluación de rendimiento en diferentes plataformas de HPC. *XXII Workshop de Investigadores en Ciencias de la Computación.* El Calafate, Santa Cruz: Repositorio Instituciona de la UNLP.

NIST. (19 de 12 de 2022). *FIPS 197.* Obtenido de Advanced Encryption Standard (AES): https://csrc.nist.gov/pubs/fips/197/ipd

Olivares, J., & Soto, J. M. (2024). *Laboratorio Remoto para Prácticas sobre FPGA .* Cordova: Universidad de Córdova.

Sideris, A., Sanida, T., & Dasygenis, M. (2019). Hardware Acceleration of the AES Algorithm using Nios-II Processor. *Panhellenic Conference on Electronics and Telecommunications (PACET).* Volos, Greece: IEEE. doi:10.1109/PACET48583.2019.8956285

ANEXOS

Anexo 1: MODOS DE OPERACIÓN DE CIFRADO DE BLOQUES

Libro electrónico, banco de pruebas propuesto por NIST

```
############################################################

  Block Cipher Modes of Operation

      Electronic Codebook (ECB)

Plaintext is
    6BC1BEE2 2E409F96 E93D7E11 7393172A
    AE2D8A57 1E03AC9C 9EB76FAC 45AF8E51
    30C81C46 A35CE411 E5FBC119 1A0A52EF
    F69F2445 DF4F9B17 AD2B417B E66C3710

############################################################

ECB-AES128 (Encryption)
------------------------------------------------------------

Key is
    2B7E1516 28AED2A6 ABF71588 09CF4F3C

Plaintext is
    6BC1BEE2 2E409F96 E93D7E11 7393172A
    AE2D8A57 1E03AC9C 9EB76FAC 45AF8E51
    30C81C46 A35CE411 E5FBC119 1A0A52EF
    F69F2445 DF4F9B17 AD2B417B E66C3710

KeyAddition   40BFABF4 06EE4D30 42CA6B99 7A5C5816

Round = 1
  Substitution  090862BF 6F28E304 2C747FEE DA4A6A47
  ShiftRow      09287F47 6F746ABF 2C4A6204 DA08E3EE
  MixColumn     529F16C2 978615CA E01AAE54 BA1A2659
  KeyAddition   F265E8D5 1FD2397B C3B9976D 9076505C

Round = 2
  Substitution  894D9B03 C0B51221 2E56883C 6038534A
  ShiftRow      89B5884A C0565303 2E389B21 604D123C
  MixColumn     0F31E929 319A3558 AEC95893 39F04D87
  KeyAddition   FDF37CDB 4B0C8C1B F7FCD8E9 4AA9BBF8

Round = 3
  Substitution  540D10B9 B3FE64AF 68B0611E D6D3EA41
  ShiftRow      54FE6141 B3B0EAB9 68D310AF D60D641E
  MixColumn     9151ABE1 E5541CFD 014A713E DA7E3134
  KeyAddition   ACD1EC9C A242E2C3 1F690F7A B704B90F
```

```
Round = 4
  Substitution  913ECEDE 3A2C982E C0F976DA A9F25676
  ShiftRow      912C7676 3AF956DE C0F2CE2E A93E98DA
  MixColumn     4D25CB1E ECF71646 7658C73B 49BCC9E9
  KeyAddition   A2616E5F 44A54D39 C029E200 92B764E9

Round = 5
  Substitution  3AEF9FCF 1B06E312 BAA59863 4FA9431E
  ShiftRow      3A06981E 1BA543CF BAA99F12 4FEFE363
  MixColumn     F89B35EC 4E40724E 025B00C7 34D7D81B
  KeyAddition   2C4AF314 32C3EFC9 C8A9B87B 252ECDA7

Round = 6
  Substitution  71D60DFA 232EDFDD E8D36C21 3F31BD5C
  ShiftRow      712E6C5C 23D3BDFA E8310DDD 3FD6DF21
  MixColumn     A0C56369 6FB884E4 4840BFBE E1D32F0A
  KeyAddition   CD4DC013 7EB3BA19 93B939FF 2BD3BCF7

Round = 7
  Substitution  BDE3BA7D F36DF4D4 DC561216 F1666568
  ShiftRow      BD6D1268 F356657D DC66BAD4 F1E3F416
  MixColumn     AC394C73 1F8DE8C7 6711B210 253DDB33
  KeyAddition   E26DBB7D 40D22134 E3B7FDA2 6B9B077C

Round = 8
  Substitution  983CEAFF 09B5FD18 11A9543A 7F14C510
  ShiftRow      98B55410 09A9C5FF 1114EA18 7F3CFD3A
  MixColumn     AB05B572 C8EB2B92 EC04E2FD 7D21EC34
  KeyAddition   41D7C653 7D669140 DD2F179D 02ACC51B

Round = 9
  Substitution  830EB4ED FF338109 C115F05E 7791A6AF
  ShiftRow      8333F0AF FF15A6ED C191B409 770E815E
  MixColumn     1741A118 91C99168 8C36386F 23AD82AA
  KeyAddition   BB36C7EB 88334D49 A4E7112E 74F182C4

Substitution  EA05C6E9 C4C3E33B 49948231 92A1131C
ShiftRow      EAC3821C C49413E9 49A1C63B 9205E331
KeyAddition   3AD77BB4 0D7A3660 A89ECAF3 2466EF97

KeyAddition   85539F41 36AD7E3A 35407A24 4C60C16D

Round = 1
  Substitution  97EDDB83 0595F380 9609DA36 29D0783C
  ShiftRow      9795DA3C 05097883 96D0DB80 29EDF336
  MixColumn     77EFE995 EA1C6263 07DB70B1 BBD06309
```

```
  KeyAddition   D7151782 62484ED2 24784988 91BC150C

Round = 2
  Substitution  0E59F013 AA522FB5 36BC3BC4 816559FE
  ShiftRow      0E523BFE AABC5913 3665F0B5 81592FC4
  MixColumn     2F19339C DA319126 86426CBE 1986D17D
  KeyAddition   DDDBA66E A0A72865 DF77ECC4 6ADF2702

Round = 3
  Substitution  C1B9249F E05C344D 9EF5CE1C 029ECC77
  ShiftRow      C15CCE77 E0F5CC9F 9E9E244D 02B9341C
  MixColumn     C4478324 8CC12C27 F7989F99 FC2BF7B3
  KeyAddition   F9C7C459 CBD7D219 E9BBE1DD 91517F88

Round = 4
  Substitution  99C61CCB 1F0EB5D4 1EEAF8C1 81D1D2C4
  ShiftRow      990EF8C4 1FEAD2CB 1ED11CD4 81C6B5C1
  MixColumn     07522BD5 02760C94 9C57905C 3C136E72
  KeyAddition   E8168E94 AA2457EB 2A26B567 E718C372

Round = 5
  Substitution  9B471922 AC365BE9 E5F7D585 94AD2E40
  ShiftRow      9B36D540 ACF72E22 E5AD19E9 94475B85
  MixColumn     E2D3DCD5 4D096172 CD665A49 2472F1AA
  KeyAddition   36021A2D 318AFCF5 0794E2F5 358BE416

Round = 6
  Substitution  0577A2D8 C77EB0E6 C52298E6 963D6947
  ShiftRow      057E9847 C72269D8 C53DA2E6 9677B0E6
  MixColumn     570D9967 42E044B2 92A4961C F855ABB1
  KeyAddition   3A853A1D 53EB7A4F 495D105D 3255384C

Round = 7
  Substitution  809780A4 EDE9DA84 3B4CCA4C 23FC0729
  ShiftRow      80E9CA29 ED4C07A4 3BFC8084 2397DA4C
  MixColumn     D8259DEA B6D85834 6DC74B22 722FCFB0
  KeyAddition   96716AE4 E98791C7 E9610490 3C8913FF

Round = 8
  Substitution  90A30269 1E1781C6 1EEFF260 EBA77D16
  ShiftRow      9017F216 1EEF7D69 1EA702C6 EBA38160
  MixColumn     E6A54262 0235B062 0A8BEC10 D24EF1C4
  KeyAddition   0C773143 B7B80AB0 3BA01970 ADC3D8EB

Round = 9
  Substitution  FEF5C71A A96C67E7 E2E0D451 952E61E9
```

```
  ShiftRow      FE6CD4E9 A9E0611A E22EC7E7 95F56751
  MixColumn     6EA80168 09CBA555 8D0B6B01 039C5D94
  KeyAddition   C2DF679B 10317974 A5DA4240 54C05DFA

Substitution  259E8514 CAC7B692 06572C09 20BA4C2D
ShiftRow      25C72C2D CA574C14 06BA8592 209EB609
KeyAddition   F5D3D585 03B9699D E785895A 96FDBAAF

KeyAddition   1BB60950 8BF236B7 4E0CD491 13C51DD3

Round = 1
  Substitution  AF4E0153 3D8905A9 2FFE4881 7DA6A466
  ShiftRow      AF894866 3DFEA453 2FA601A9 7D4E0581
  MixColumn     EB181CE7 947E65BB 07D26B9F AC6FA1D5
  KeyAddition   4BE2E2F0 1C2A490A 247152A6 8603D7D0

Round = 2
  Substitution  B398988C 9CE53B67 36A30024 447B0E70
  ShiftRow      B3E50070 9CA30E8C 367B9867 44983B24
  MixColumn     3912C6CB 5F5FAC11 1E14CF77 2406C627
  KeyAddition   CBD05339 25C91552 47214F0D 575F3058

Round = 3
  Substitution  1F70ED12 3FDD5900 A0FD84D7 5BCF046A
  ShiftRow      1FDD846A 3FFD0412 A0CFED00 5B7059D7
  MixColumn     AC436FAC 74C0FC9C FC09AED9 A887FB71
  KeyAddition   91C328D1 33D602A2 E22AD09D C5FD734A

Round = 4
  Substitution  812E343E C3F6773A 98E5705E A6548FD6
  ShiftRow      81F670D6 C3E58F3E 9854343A A62E775E
  MixColumn     BE30F6A9 18A66148 D956EAA7 0C3D8414
  KeyAddition   517453E8 B0F43A37 6F27CF9C D7362914

Round = 5
  Substitution  D192ED9B E7BF809A A8CC8ADE 0E05A5FA
  ShiftRow      D1BF8AFA E7CCA59B A805ED9A 0E9280DE
  MixColumn     13CB74B2 A40BCC76 3314D924 EF74FEA7
  KeyAddition   C71AB24A D88851F1 F9E66198 FE8DEB1B

Round = 6
  Substitution  C6A237D6 61C4D1A1 998EEF46 BB5DE9AF
  ShiftRow      C6C4EFAF 618EE9D6 995D37A1 BBA2D146
  MixColumn     80D02D3F 74904773 58DB5283 07CA6A29
  KeyAddition   ED588E45 659B798E 8322D4C2 CDCAF9D4
```

```
Round = 7
  Substitution  556A196E 4D14B619 EC934825 BD749948
  ShiftRow      55144848 4D93996E EC741919 BD6AB625
  MixColumn     96ED0933 C3AE4501 5F368170 4C8DCF4A
  KeyAddition   D8B9FE3D 9CF18CF2 DB90CEC2 022B1305

Round = 8
  Substitution  6156BB27 DEA16489 B9608B25 77F17D6B
  ShiftRow      61A18B6B DE607D27 B9F1BB89 77566425
  MixColumn     DAD5705F 5DBE2D2A 531FA593 555286E1
  KeyAddition   3007037E E83397F8 623450F3 2ADFAFCE

Round = 9
  Substitution  04C57BF3 9BC38841 AA18530D E59E798B
  ShiftRow      04C3538B 9B1879F3 AA9E7B41 E5C5880D
  MixColumn     8EE7E791 8FD37F2A CC410182 00FA3C63
  KeyAddition   22908162 9629A30B E49028C3 57A63C0D

Substitution  93600CAA 90A50A2B 6960342E 5B24EBD7
ShiftRow      93A534D7 9060EBAA 69240C2B 5B600A2E
KeyAddition   43B1CD7F 598ECE23 881B00E3 ED030688

KeyAddition   DDE13153 F7E149B1 06DC54F3 EFA3782C

Round = 1
  Substitution  C1F8C7ED 68F83BC8 6F86200D DF0ABC71
  ShiftRow      C1F82071 6886BCED 6F0AC7C8 DFF83B0D
  MixColumn     DB3BEA62 104DA143 CFE1B3F7 807446A3
  KeyAddition   7BC11475 98198DF2 EC428ACE AA1830A6

Round = 2
  Substitution  2178FA9D 46D45D89 CE2C7E8B ACAD0424
  ShiftRow      21D47E24 462C049D CEADFA89 AC785D8B
  MixColumn     7F346581 618FDEC3 18130C17 1D30E8C7
  KeyAddition   8DF6F073 1B196780 41268C6D 6E691EB8

Round = 3
  Substitution  5D428C8F AFD485CD 83F7643C 9FF9726C
  ShiftRow      5DD4646C AFF7728F 83F98CCD 9F42853C
  MixColumn     D52EF58F BA43366A 4C28356A 5AB38805
  KeyAddition   E8AEB2F2 FD55C854 520B4B2E 37C9003E

Round = 4
  Substitution  9BE43789 54FCE820 002BB331 9ADD63B2
  ShiftRow      9BFCB3B2 542B6389 00DD3720 9AE4E831
  MixColumn     3304D786 3F2E39BD 6BD8D3AA C15BE6DB
```

```
  KeyAddition   DC4072C7 977C62C2 DDA9F691 1A504BDB

Round = 5
  Substitution  860940C6 8810AA25 C1D34281 A253B3B9
  ShiftRow      861042B9 88D3B3C6 C1534025 A209AA81
  MixColumn     DCD9C2AA 103D7774 09827D01 6FD47C47
  KeyAddition   08080452 6CBEEAF3 C370C5BD 7E2D69FB

Round = 6
  Substitution  3030F200 50AE870D 2E51A67A F3D8F90F
  ShiftRow      30AEA60F 5051F900 2ED8F20D F330877A
  MixColumn     2089D846 AAE2E858 D0851E42 507B584D
  KeyAddition   4D017B3C BBE9D6A5 0B7C9803 9A7BCBB0

Round = 7
  Substitution  E37C21EB EA1EF606 2B10467B B8211FE7
  ShiftRow      E31E46E7 EA101FEB 2B212106 B87CF67B
  MixColumn     5EF243B3 0B00E2E7 120C4271 623ABEAF
  KeyAddition   10A6B4BD 545F2B14 96AA0DC3 2C9C62E0

Round = 8
  Substitution  CA248D7A 20CFF1FA 90ACD72E 71DEAAE1
  ShiftRow      CACFD7E1 20ACAA7A 90DE8DFA 7124F12E
  MixColumn     F3CC8884 7FFC4D92 35415A17 511FDE1A
  KeyAddition   191EFBA5 CA71F740 046AAF77 2E92F735

Round = 9
  Substitution  D4720F06 74A36809 F20279F5 314F6896
  ShiftRow      D4A37996 74026806 F24F0F09 317268F5
  MixColumn     A294248A 80CEACFA 2874B85F 699897B8
  KeyAddition   0EE34279 993470DB 00A5911E 3EC497D6

Substitution  AB112CB6 EE1851B9 63068172 B21C88F6
ShiftRow      AB1881F6 EE0688B6 631C2CB9 B2115172
KeyAddition   7B0C785E 27E8AD3F 82232071 04725DD4


Ciphertext is
    3AD77BB4 0D7A3660 A89ECAF3 2466EF97
    F5D3D585 03B9699D E785895A 96FDBAAF
    43B1CD7F 598ECE23 881B00E3 ED030688
    7B0C785E 27E8AD3F 82232071 04725DD4

================================================================
```