Bloc IP accelerador hardware del algoritme d'encriptació AES-256

Reda Chriki, *Estudiant, UAB,* Ricardo Martínez, *Supervisor, CNM-IMB, CSIC,*
i Lluís Terés, *Tutor, CNM-IMB, CSIC i UAB*

Resum—Es presenta un bloc IP (Intellectual Property) per a l'algorisme AES-256 en modes ECB i CBC en un dispositiu Zynq-7000 ZC702. El bloc IP s'ha dissenyat i validat utilitzant eines de disseny de circuits i programació de FPGA per complir els estàndards de seguretat AES. A més, s'ha integrat en un sistema de processament ARM Còrtex-A9 per mostrar-ne l'aplicabilitat en entorns de comunicacions segures. S'ha dut a terme una anàlisi detallada del rendiment en termes de velocitat i temps, comparant els resultats entre una implementació amb AXI4-Lite i AXI4-Full. Els resultats obtinguts demostren la viabilitat d'implementar l'algorisme AES-256 en modes ECB i CBC en un dispositiu Zynq-7000 ZC702 mitjançant un bloc IP dotat amb AXI4-Full, oferint una alternativa d'alta velocitat i eficiència per a aplicacions de comunicacions segures. Aquest treball és útil per a desenvolupadors que busquen implementar l'algorisme AES-256 a un sistema i oferir una solució de seguretat en temps real en sistemes encastats.

Index Terms—AES-256, AMBA-AXI4, SoC, IP, PUF, Accelerador Hardware, Sistemes Encastats.

1 Introducció i Objectius

En els últims anys és innegable el creixement que s'ha produït en el sector IoT (Internet of Things), la previsió a escala mundial són sorprenents, triplicant el nombre de dispositius connectats segons les dades facilitades per Transforma Insights. D'aquesta forma, es passaria d'aproximadament 8.740 milions el 2020 a més de 25.400 milions el 2030 [1]. Per tant, és obvi que s'estan convertint cada cop més en una part integral de la nostra vida diària, amb dispositius connectats que ens permeten controlar i monitoritzar la nostra llar, salut i vehicles des de qualsevol lloc. La IoT, es basa doncs principalment en aplicacions de connectivitat inalàmbrica amb la capacitat de recopilar i transmetre dades a través d'internet.

El maneig d'aquestes dades porta associat diferents reptes, un dels principals i que cada cop preocupa més als directius d'empresa i CEO (Chief Officer Excutive) de les empreses segons un estudi realitzat per PwC [2], és la ciberseguretat, convertint-se en el principal risc per al creixement de negoci de cara als pròxims 3 anys segons l'enquesta realitzada per KPMG [3]. A mesura que avancen les innovacions en tots els dispositius intel·ligents es presenten més i noves amenaces a ciberatacs, per tant, solucions com les que es pretén treballar en aquest projecte posa en el punt de mira aquest repte entre d'altres que es troben associats de forma directa o indirecta al tema en qüestió. El xifratge de clau simètrica, com l'AES-256, és el més utilitzat actualment per a protegir la informació confidencial en convertir les dades en un codi illegible per un receptor que no té la clau de desxifrat adequat.

1.1 Motivació i Justificació

El grup ICAS de l'IMB-CNM (CSIC) està desenvolupant una plataforma per facilitar el procés de disseny de SoC codi obert basat en l'arquitectura RISC-V [4] i orientats principalment a l'àmbit de la IoT. En aquest context, es vol desenvolupar un bloc IP per a l'acceleració per hardware dels algoritmes d'encriptació AES-256 per ampliar el catàleg de blocs IP. La primera versió d'aquesta IP contempla els processos de generació de claus, d'encriptació i de desencriptació a través d'una interfície de comunicacions AXI4-Lite.

El desenvolupament d'un accelerador de xifratge AES-256 es justifica per la necessitat de fer xifratge de dades de manera ràpida i eficient. L'AES (Advanced Encryption Standard) [5] és un algorisme de xifratge simètric que es fa servir àmpliament per protegir la informació confidencial i garantir la privadesa i la seguretat de les dades. L'AES-256 és una variant de l'AES que fa servir una clau de 256 bits i és considerat un dels algorismes de xifratge més segurs disponibles actualment.

Tot i això, l'algorisme AES-256 pot ser computacionalment costós i requerir una gran quantitat de recursos de processament. Això pot ser un problema en aplicacions que requereixen un xifratge d'alta velocitat, com ara el xifratge de grans quantitats de dades (Big Data) en temps real.

Un accelerador per l'algoritme AES-256 és un dispositiu hardware dissenyat específicament per fer xifratge AES-256 de manera més ràpida i eficient que una CPU de propòsit general. Un accelerador de xifratge pot emprar tècniques de paral·lelisme i optimització de hardware per accelerar el procés de xifratge i millorar la seva eficiència. Això pot ser especialment útil en aplicacions que requereixen un alt rendiment de xifratge o en entorns on hi ha una gran quantitat de trànsit de xifratge.

En aquest treball abordem una evolució de la primera versió de la IP per millorar-ne algunes parts més febles, facilitar la seva connexió via AXI4-Full, integrar-lo amb un processador RISC i validar-lo amb una aplicació senzilla.

1.2 Objectius

Aquest treball pretén com a objectiu global disposar d'un bloc IP per l'acceleració hardware de l'AES-256 provat i

testat en FPGA (Field-Programmable Gate Array) i preparat pel seu posterior ús com a bloc o component de sistemes complets que requereixin mòduls d'encriptació i de desencriptació.

El treball que es proposa pretén introduir millores al bloc AES-256, ja desenvolupat en un treball de fi de grau previ "AES256 ECB Hardware Accelerator with AXI4-Lite" [6]. D'aquest projecte, utilitzarem la part del "Device ECB AES-256", que té algunes oportunitats de millores significatives, per tal de convertir-lo en un bloc IP de fàcil ús i millorar-ne les prestacions. Així, doncs, les tasques concretes d'aquest TFG per dur a terme aquesta nova versió de la IP AES-256 són:

1) Assolir i reestructura la IP existent de l'AES-256.
2) Extensió dels usos de registre de control i d'estat.
3) Dotar-lo d'una interfície de bus per a l'estàndard AXI4-Full substituint l'AXI4-Lite actual.
4) Extensió en els modes de treball, afegint CBC, a més de l'ECB existent.
5) Estudi preliminar per la inclusió d'estructures PUF per a la generació de claus.
6) Desenvolupar una llibreria amb funcions en C per l'ús de l'AES-256.

1.3 Estat de l'art

Els acceleradors hardware s'entenen com un maquinari de suport que substitueixen a la CPU a l'hora de realitzar una funció d'aquesta, alliberant-la de cicles de rellotge que pot dedicar a altres funcions i a més aquest hardware ho realitza d'una manera més ràpida, i per tant eficaç respecte l'ús i consum de l'energia.

En els darrers anys, s'ha produït un augment en la demanda d'acceleradors criptogràfics de maquinari a FPGA a causa de la necessitat creixent de seguretat a l'era digital. A més, l'augment de l'ús d'aplicacions al núvol i l'Internet de les coses (IoT) ha portat a una demanda més gran de solucions de seguretat més eficients.

Des que es va publicar l'AES el novembre del 2001, són molts els fabricants i empreses que han desenvolupat blocs IP (Intellectual Property) propietaris que implementen aquest algorisme oferint la seva versió comercial, com Xilinx [7], Intel i AMD, d'entre altres.

Respecte a la ciberseguretat, fins avui dia, algorismes com l'RSA [8] de clau asimètrica d'entre 1024 i 2048 bits normalment, presenten una major seguretat i se solen reservar per a aplicacions que requereixen una seguretat major, ja que el seu ús compromet que les comunicacions siguin més lentes i costoses de processar. És per això, que cercant l'equilibri segur i ràpid de processament de dades, el candidat més idoni és l'AES-256, però, tot i això, a mesura que augmenta la quantitat de dades, cada cop més freqüent per les innovacions tecnològiques d'intel·ligència artificial i big data, juntament amb l'exigència dels usuaris, l'AES-256 en processadors d'ús general perd efectivitat per aquestes aplicacions, per la qual cosa es justifica l'existència d'acceleradors hardware criptogràfics.

2 MARC TEÒRIC

2.1 AES-256

AES-256 és un algorisme de xifratge simètric que utilitza una clau de 256 bits. L'algorisme AES [5] (Advanced Encryption Standard) va ser desenvolupat per l'Institut Nacional d'Estàndards i Tecnologia (NIST) dels Estats Units com un reemplaçament per a l'algorisme de xifratge DES. AES-256 és considerat un dels algorismes de xifratge més segurs disponibles avui dia.

AES fa servir el concepte de blocs de dades de 128 bits i claus de longitud variable (128, 192 o 256 bits). L'algorisme funciona a través de diverses rondes de xifratge, en què s'efectuen operacions matemàtiques complexes amb els blocs de dades i la clau.

L'algorisme AES es basa en diversos processos de xifratge: AddRoundKey, SubBytes, ShiftRows, MixColumns. L'operació AddRoundKey aplica una XOR entre el bloc de dades i la clau de xifratge actual. SubBytes reemplaça cada byte del bloc amb un altre valor específic, segons una taula de substitució. ShiftRows desplaça els bytes a les files del bloc. MixColumns barreja els bytes a les columnes. A continuació, es pot apreciar a la figura 1 l'arquitectura d'aquestes funcions:

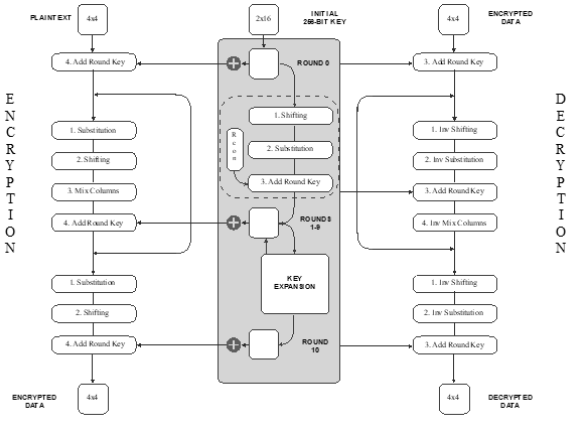

Figura 1. AES-256 Encriptació, Desencriptació i Generació de Claus. [6]

A més d'aquestes operacions, AES-256 utilitza 14 rondes de xifratge addicionals. Cada ronda fa ús d'una clau diferent, generada a partir de la clau original mitjançant un procés d'expansió de clau. El nombre de rondes depèn de la longitud de la clau feta servir, sent més rondes amb una longitud de clau més gran.

Al final, el bloc resultant és el text xifrat. El procés invers es fa servir per desxifrar el text.

2.1.1 AES-256 ECB (Electronic Code Book)

El mode ECB (Electronic Code Book) és la manera d'operació més simple i bàsica que s'utilitza amb AES-256. En aquest mode, es divideix el text pla en blocs de 128 bits i es xifren cada bloc individualment amb la mateixa clau. El problema amb ECB és que si hi ha patrons repetitius al text pla, aquests patrons es mantindran en el text xifrat, cosa que podria permetre a un atacant desxifrar el missatge. A la figura 2 es pot apreciar visualment el mode de funcionament:

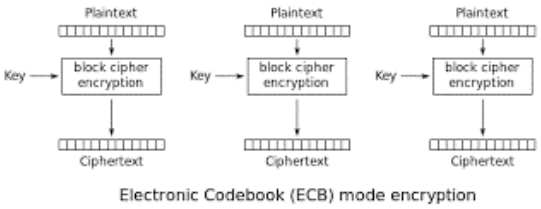

Figura 2. Encriptació en mode ECB.

2.1.2 AES-256 CBC (Cipher Block Chaining)

El mode CBC (Cipher Block Chaining) és una manera d'operació més avançada que s'utilitza amb AES-256. En aquest mode, es divideix el text pla en blocs de 128 bits i es xifra cada bloc fent servir una clau diferent generada a partir de la clau original i el bloc anterior. En aquest mode, el bloc actual es xifra usant una XOR amb el bloc anterior, cosa que significa que cada bloc xifrat és dependent del bloc anterior. Això ajuda a amagar patrons repetitius al text pla i fa que sigui més difícil per a un atacant desxifrar el missatge. A la figura 3 es pot apreciar visualment el mode de funcionament:

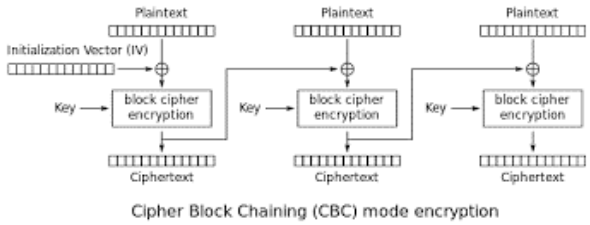

Figura 3. Encriptació en mode CBC.

2.2 PUF

Un PUF (Physical Unclonable Function) [9] és un tipus de dispositiu de seguretat basat en les característiques físiques úniques d'un circuit per generar una resposta única i inimitable. La idea és que, a causa de la variabilitat natural dels processos de fabricació, cada dispositiu físic té característiques úniques que es poden utilitzar com una mena d'empremta dactilar, similar a les característiques físiques d'una persona.

Un PUF pot ser considerat com un circuit que té un comportament aleatori i no reproduïble, i que es fa servir per generar una clau única i privada amb diferents aplicacions [10]. Els PUF es fan servir per proporcionar una protecció robusta contra la reproducció no autoritzada, ja que és molt difícil reproduir les característiques físiques úniques d'un dispositiu.

Hi ha diferents tipus de PUF, com els basats en metalls de pont o els basats en interrupcions de canal. Cadascú es basa en un principi físic diferent per generar la seva resposta única. Per exemple, els basats en canals de difusió usen variacions en la resistència d'un canal de difusió per generar una resposta, mentre que els basats en metalls de pont fan servir variacions en la resistència d'un metall de pont per generar una resposta.

La seguretat de PUF es basa en la dificultat de reproduir les característiques físiques úniques del circuit. Tot i que els científics i els atacants han aconseguit alguns avenços en la reproducció de PUF, encara es considera com una de les formes més segures per protegir els sistemes i dispositius electrònics.

2.3 AMBA: Bus AXI4-Lite i AXI4-Full

AXI (de l'anglés, Advanced eXtensibles Interface) és una especificació d'interfaç de bus de sistema que s'utilitza per connectar components de hardware en un sistema digital. Desenvolupada per ARM Holdings, forma part d'ARM AMBA, una família de busos de microcontroladors introduïda per primera vegada el 1996. La primera versió d'AXI va ser introduida a AMBA 3.0 el 2003. AMBA 4.0 publicat el 2010, inclou la segona versió principal d'AXI, AXI4.

AXI4 és un bus de sistema que permet la comunicació entre components de hardware a través d'un conjunt de senyals de control i dades. Es fa servir per transferir dades i senyals de control entre components de hardware, com processadors, memòria, dispositius d'entrada/sortida (E/S) i altres components hardware.

AXI4 és una especificació d'interfície de bus molt flexible i es pot usar en una àmplia varietat d'aplicacions i sistemes, des de sistemes encastats fins a sistemes de computació d'alt rendiment. És àmpliament utilitzat en la indústria de la tecnologia de la informació i és compatible amb una àmplia gamma de dispositius hardware i sistemes operatius.

Hi ha tres tipus d'interfícies AXI4:

- AXI4-Lite: Per a una comunicació simple i de baix rendiment amb assignació de memòria, és una interfície lleugera, d'una sola transacció i mapejada en memòria.
- AXI4-Stream: Per al flux de dades d'alta velocitat i en temps real. Elimina completament el requisit d'una fase d'adreçament, per tant, no es consideren mapejades en memòria i permet una mida de ràfega de dades il·limitada. Es fa servir principalment per aplicacions de processament de senyals d'imatges i vídeo.
- AXI4-Full: Per a interfícies mapejades en memòria i permet ràfegues d'alt rendiment de fins a 256 cicles de transferència de dades amb una sola fase d'adreçament.

3 METODOLOGÍA

Per a la realització d'aquest projecte s'ha utilitzat una metodologia àgil basada en iteracions incrementals. Això afavoreix l'adaptació als canvis de forma ràpida i començar a treballar directament sobre els objectius establerts d'una forma molt primerenca.

A més, adoptar aquesta metodologia àgil basada en l'aprenentatge continu i l'adaptació a factors fluctuants, ens permet evolucionar el desenvolupament de la solució a través de l'experiència.

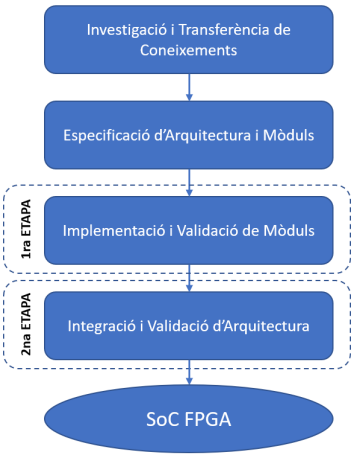

Figura 4. Metodologia.

3.1 Flux de treball

El flux de treball adoptat en aquest projecte es divideix en dues parts ben diferenciades que a la vegada es troben interconnectats on la sortida del primer s'utilitzarà com a entrada del segon flux, per últim, aconseguirem l'objectiu final del qual podrem extreure valors quantitatius, mesurables i comparables que ens serviran per extreure conclusions de valor sobre el treball fet (figura 5).

El fet de dividir en dues parts el flux metodològic ens atorga l'avantatge de poder analitzar i detectar errors en una etapa inicial evitant arrossegar l'error al llarg del flux, això ajuda que el procés d'execució d'aquest projecte sigui iteratiu, afegint millores gràcies a l'experiència i coneixements adquirits per cada iteració consolidada al llarg del procés.

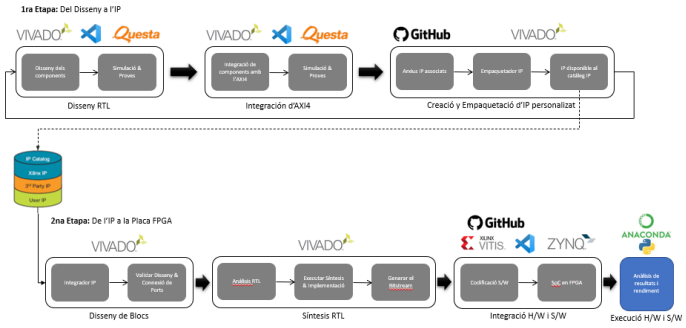

Figura 5. Flux metodològic.

La primera part del flux metodològic (vegeu figura 5, 1a etapa) correspon a la part de portar del disseny sobre el paper al codi, amb la finalitat d'aconseguir produir un bloc IP personalitzat. Podem veure que aquí és on es concentra la dificultat d'aquest projecte i, per tant, podem veure que es tracta d'un procés iterable on a mesura que s'avença en el projecte s'afegiran més prestacions i millores incrementals.
La segona part del flux metodològic (vegeu figura 5, 2a etapa) correspon a la part d'integrar aquest bloc IP personalitzat a la placa, mitjançant l'integrador IP de Vivado Xilinx, amb l'objectiu de poder realitzar execucions per obtenir dades sobre el rendiment de la IP i mostrar els resultats de la feina feta.

3.2 Eines associades al flux de treball

En aquest apartat es detallaran les eines utilitzades per a treballar sobre aquest projecte.

3.2.1 Llenguatge HDL: Verilog

Verilog és un llenguatge de descripció de hardware (HDL). Es diferencia dels llenguatges de programació de propòsit general en el fet que s'usa específicament per modelar el hardware. Així, a Verilog es poden especificar registres, cables, portes, rellotge, etc. És molt útil quan es tenen les especificacions del hardware sobre el paper, i es vol simular i provar primer, abans de sintetitzar el circuit, estalviant així temps i diners. El llenguatge és també força senzill, només requereix coneixements de lògica digital, i té una sintaxi molt similar a la de C.

Per als desenvolupaments d'aquest treball es decideix fer servir el llenguatge HDL Verilog, ja que aquests últims anys ha agafat molta força dins de la indústria del disseny hardware, compta amb una gran comunitat i una extensa documentació, a més, per la seva senzillesa el fan perfecte per a començar. Per profunditzar han estat de gran ajuda els llibres Digital Logic Design Using Verilog de V. Taraate [11] i Digital Design and Computer Architecture de David M. Harris i Sarah L. Harris [12].

3.2.2 Vivado

Vivado és un IDE, eina d'entorn de desenvolupament integrat, desenvolupat pel fabricant Xilinx que s'utilitza per programar FPGAs.

3.2.3 Vitis

La plataforma de software Vitis inclou Vivado Design Suite i funciona amb dissenys de hardware creats en Vivado. La plataforma de software unificada Vitis és un entorn de desenvolupament integrat (IDE) per al desenvolupament d'aplicacions de software embegut dirigit als processadors Xilinx. La plataforma de software Vitis es basa en el codi obert Eclipse.

3.2.4 QuestaSim

QuestaSim és un entorn multillenguatge de Siemens, per a la simulació de llenguatges de descripció de hardware com VHDL, Verilog i SystemC, a més, inclou un depurador de C incorporat. La simulació es realitza mitjançant la interfície gràfica d'usuari (GUI), o de forma automàtica mitjançant scripts.

3.2.5 Xilinx Zynq-7000 SoC ZC702

La característica fonamental dels dispositius Zynq-7000 és que combinen un processador doble nucli ARM Còrtex-A9 amb una lògica corresponent a les familias Artix 7 o Kintex™ de Xilinx. Ambdues parts estan connectades entre si mitjançant interfícies AXI, les quals són un estàndard a la indústria i proporcionen comunicació System-on-Chip amb baixa latència i gran amplada de banda.

4 DESENVOLUPAMENT DEL TREBALL

Després d'introduir en la secció anterior les eines a usar i la metodologia de treball, a continuació es detallaran les etapes d'iteració que seguirem per a la consecució dels objectius.

4.1 Estat Previ

L'estat previ del dispositiu AES-256 programat en Verilog funciona exclusivament en mode ECB. S'ha verificat que el dispositiu realitza la generació de claus, i les operacions d'encriptar i desencriptar dades de manera correcta i efectiva. S'han realitzat una sèrie de proves per assegurar que el dispositiu compleixi els estàndards de seguretat AES-256 mitjançant un programa extern en línia, CrypTool-Online [13], que ha estat configurat de la següent forma (vegeu figura 6):

- Number of rounds: 14
- Permutation: 00010203 05060704 0a0b0809 0f0c0d0e
- Chaining: None

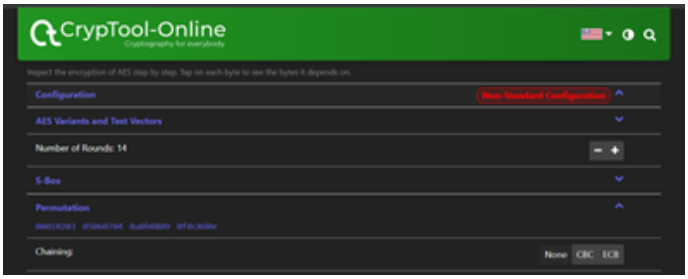

Figura 6. Configuració de l'eina CrypTool-Online. [13]

La implementació de bus en aquest primer dispositiu s'ha realitzat amb AXI4-Lite, el qual presenta certes limitacions a l'hora de connectar-se i treballar amb altres dispositius com s'ha explicat en el marc teòric front l'ús de l'AXI4-Full.

La funció principal del dispositiu, per tant, és rebre dades i processar-les seguint la lògica interna (vegeu figura 7), i retornar dades (xifrats o desxifrats). Tenim tres casos d'ús pel component:

- Primer, a partir d'una llavor (seed, en anglès) de 256 bits el component realitza la generació de claus que s'utilitzen per als passos posteriors.
- Segon, un cop generada la clau, s'agafen dades d'entrada de 128 bits per a la seva encriptació.
- Tercer, un cop encriptada la paraula, es fa el procés de desencriptació, generant paraules en pla de 128 bits.

Partint d'aquest component, es pretén reestructurar la IP amb tots els següents canvis:

- Substituir el bus actual AXI4-Lite per un bus AXI4-Full.
- Ampliar els modes de funcionament ECB, i afegir el CBC.
- Dissenyar i implantar un registre per controlar la nova lògica, amb noves funcionalitats que interactuin amb les FIFO.
- Dissenyar i implementar un registre d'estat per informar de l'estat actual de la IP (inclou les FIFO i Device AES-256).

El resultat final de la distribució del bloc IP pretén cobrir tots els canvis llistats per incrementar la seguretat en les aplicacions que es vulgui fer ús de la IP. Incrementar la seva velocitat en la transferència de dades sobre el bus amb

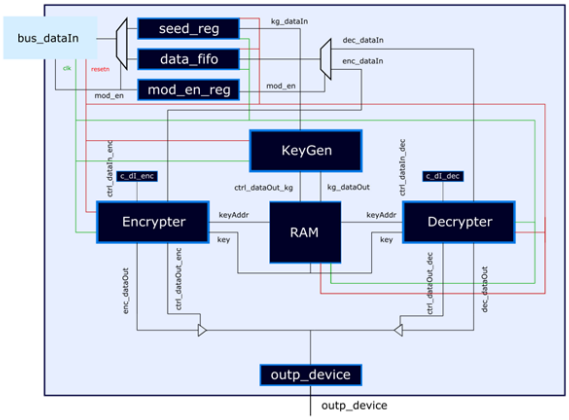

Figura 7. Aproximació de la distribució hardware de l'AES-256 ECB IP Core. [6]

altres dispositius, gràcies a l'AXI4-Full. Incrementar la seva usabilitat amb noves funcionalitats mitjançant el registre de control.

4.2 Disseny d'arquitectura

L'arquitectura d'aquest projecte consta del Device AES-256, el mòdul principal encarregat d'encriptar i desencriptar les dades, que es troba connectat a memòria la qual es troba formada per tres FIFO (FIFO_SEED, FIFO_DATA_IN i FIFO_DATA_OUT), a més d'un registre de control i d'un registre d'estat interconnectats al Device AES-265, i per últim integrats mitjançant el protocol AXI4-Full (figura 8).

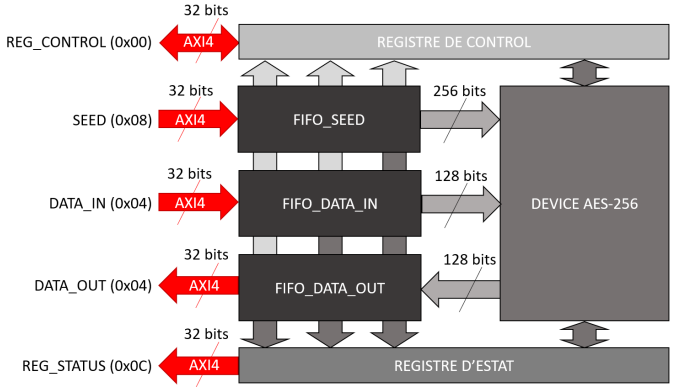

Figura 8. Arquitectura del bloc IP. Paràmetres n i m configurables pel dissenyador.

Aquesta és l'arquitectura que s'ha dissenyat, la qual s'utilitzarà com a referència pel desenvolupament dels components.

4.2.1 Bus: AXI4-Full

L'AXI4-Full és la interfície de bus per la qual es mouran les dades, a continuació es defineixen les adreces per mapejar la interacció (lectura i escriptura), sobre els diferents mòduls de l'arquitectura.

El mapeig d'adreces per lectura s'ha dissenyat de la següent manera:

- 0x00: Dada de 32 bits del REG_CONTROL.

- 0x04: Dada de 32 bits provinent de la FIFO_OUT
- 0x08: n/a
- 0x0C: Dada de 32 bits del REG_STATUS

Mapeig d'adreces per escriptura:

- 0x00: Adreça per escriure en el REG_CONTROL.
- 0x04: Escriure dada de 32 bits sobre la FIFO_IN.
- 0x08: Escriure dades de 32 bits sobre la FIFO_SEED.
- 0x0C: n/a

4.2.2 Memòries

L'emmagatzematge del bloc IP, es distribueix en tres FIFO (First In, First Out), a continuació, una breu explicació del funcionament de cadascuna:

- FIFO_SEED: Aquest mòdul implementa una FIFO per a rebre dades de 32 bits que seran emmagatzemats fins a acumular 256 bits. Aquesta és enviada al Device AES-256 i s'utilitzarà per a generar la clau ("Key") que es farà servir per a encriptar i desencriptar.
- FIFO_IN: La FIFO d'entrada ha de rebre paraules de 32 bits i emmagatzemar-les dins de la FIFO d'entrada fins a acumular paraules de 128 bits, un cop llegida la quarta paraula de 32 bits, s'empaqueta la paraula en una dada de 128 bits i s'envia al Device AES-256 per a processar-la mitjançant l'algoritme AES-256.
- FIFO_OUT: La FIFO de sortida rep paraules de 128 bits del Device AES-256, després del processament d'encriptació o desencriptació, i s'emmagatzema dins de la FIFO en paraules de 32 bits.

4.2.3 Registre d'Estat

El registre d'estat té la funció principal de mantenir al programador informat de l'estat del sistema, per fer-ho llegeix els diferents senyals del Device AES-256 i les FIFO, per fer els canvis pertinents sobre el registre que consta de 32 bits, a continuació s'indica la descripció de cada bit.

0 : Indica estat d'encriptació en el Device. Per 1, indica que el Device està encriptant dades.
1 : Indica estat de desencriptació en el Device. Per 1, indica que el Device està desencriptant dades.
2 : Indica estat de la clau en el Device. Per 1, indica que la clau es troba generada en el Device. Per 0, indica que no hi ha cap clau generada en el Device.
3 : Per 1, indica que la FIFO_IN està plena. Per 0, indica que la FIFO_IN no està plena.
4 : Per 1, indica que la FIFO_OUT està plena. Per 0, indica que la FIFO_OUT no està plena.
5 : Per 1, indica que la FIFO_SEED està plena. Per 0, indica que la FIFO_SEED no està plena.
6 : Per 1, indica que la FIFO_IN està buida. Per 0, indica que la FIFO_IN no està buida.
7 : Per 1, indica que la FIFO_OUT està buida. Per 0, indica que la FIFO_OUT no està buida.
8 : Per 1, indica que la FIFO_SEED està buida. Per 0, indica que la FIFO_SEED no està buida.
9-31 : N/a. Sense assignació.

4.2.4 Registre de Control

Es tracta d'un component que utilitzarem com a màster, s'encarregarà de rebre els ordres i a l'estar interconnectat amb la resta de components, transmetre els senyals en aquests.

0 : Activa l'encriptació en el Device.
1 : Activa la desencriptació en el Device.
2 : Força l'activament de la generació de claus en el Device.
3 : Activa el buidament de la FIFO_IN.
4 : Activa el buidament de la FIFO_OUT.
5 : Activa el buidament de la FIFO_SEED.
6-31 : Sense assignació.

4.2.5 Device AES-256

El principal canvi que s'ha realitzat en aquest mòdul, és l'habilitat de tenir un nou mode de funcionament d'encriptació criptogràfica més segura, el mode CBC. Per això, s'ha reestructat el mòdul amb dos pins d'entrada:

- iv_key: Senyal de 128 bits per on entra l'initial vectorize.
- iv_key_active: Senyal d'un bit, indica que la iv_key és vàlida.

La primera part on afecten aquests canvis, són en la dada que entra l'encriptador, si està treballant en mode ECB, el text pla entra directament a l'encriptador, en canvi, si treballa en mode CBC, primer s'ha d'efectuar una operació XOR entre el text plac i el iv_key. Per fer-ho possible, farem ús d'un multiplexor que mitjançant el senyal iv_key_active, deixarem passar un senyal o un altre (vegeu figura 9).

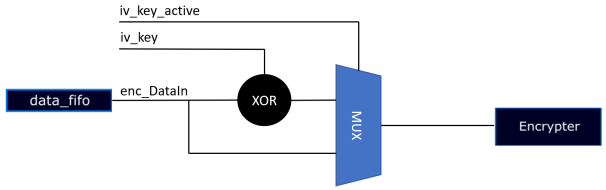

Figura 9. Lògica introduïda en el port d'entrada de dades de l'encriptador.

El mateix passa per la part de desencriptació, si estem treballant en mode CBC, abans de portar la dada desencriptada com la sortida del Deivice AES-256, s'ha passat el senyal per la mateixa lògica, és a dir, mitjançant un multiplexor controlat pel senyal iv_key_active, deixarà passar el senyal amb l'operació XOR, en aquest cas entre la dada de sortida del desencriptador i el iv_key, en cas contrari, deixarà passar la dada sense cap operació (vegeu figura 10).

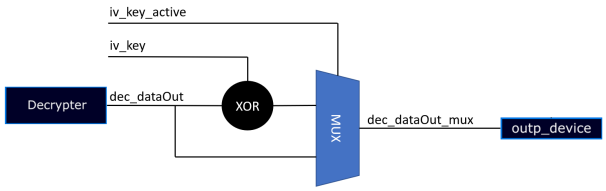

Figura 10. Lògica introduïda en el port de sortida de dades del desencriptador.

Un cop introduït tots aquests canvis en el mòdul del Device AES-256, el resultat final es pot veure en la figura 11.

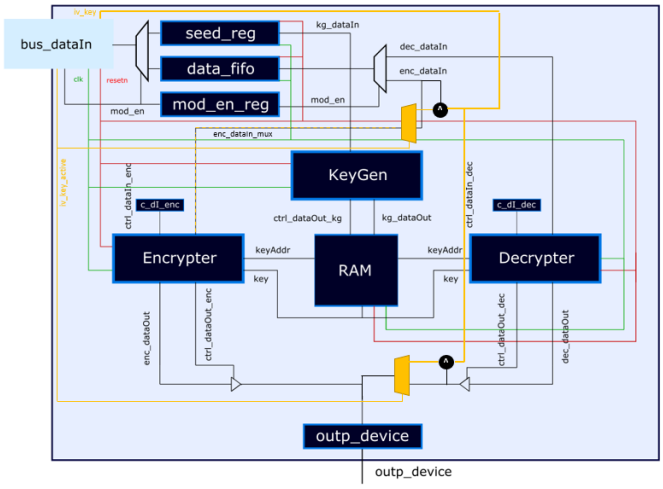

Figura 11. Arquitectura del nou Device AES-256 amb les noves senyals.

4.3 Implementació i Verificació

La feina d'implementació i modelització del hardware necessari s'ha dut a terme mitjançant el llenguatge HDL de Verilog. En aquesta part s'han dut a terme totes les implementacions especificades en l'apartat de Disseny d'Arquitectura.

Un cop realitzat i fets els canvis, passem a realitzar els testbench corresponents. Un Testbench, o banc de proves en català, és un entorn que s'utilitza per a verificar un disseny o model. En el cas d'aquest projecte s'usa per a verificar la funcionalitat d'un mòdul de disseny (verificació a nivell de bloc) o d'un grup de mòduls interconnectats (verificació a nivell de sistema). Per a verificar la funcionalitat de RTL sintetitzable, el programa del banc de proves generalment s'escriu en llenguatge Verilog/VHDL o C/C++.

L'objectiu és poder visualitzar mitjançant un entorn gràfic un waveform que mostri com responen els senyals als diferents impulsos que anteriorment s'han definit en el Testbench. Per a cada mòdul, s'ha portat a cap un testbench per a la seva verificació.

4.4 Generació del bloc IP Accelerador AES-256

El principal resultat fruit d'aquest treball és el bloc IP (Intellectual Property). Actualment, llest per ser afegit a dissenys que precisin fer ús de l'algorisme AES-256. Amb capacitat per a treballar en modes ECB256 i CBC256, en la figura 12 es pot veure les connexions que té habilitades per connectar-se mitjançant la interfície AXI4-Full. A continuació es llisten els ports disponibles:

- s00_axi: Ports per la transmissió de dades per adreçament seguint l'estàndard AMBA AXI4.
- s03_iv_key[127:0]: Port d'entrada per l'Initial Vector que s'usa per la IP quan treballa en mode CBC.
- s00_axi_aclk: Port d'entrada pel senyal de rellotge.
- s00_axi_aresetn: Port d'entrada pel senyal de reinici de la IP.

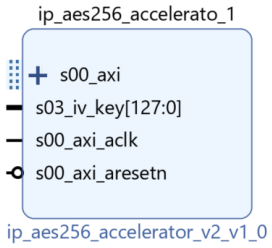

Figura 12. Bloc IP amb les connexions d'entrada i de sortida.

A més, mitjançant la finestra de "Re-customize IP"s'ha habilitat l'opció de poder modificar la mida de les FIFO que s'utilitzen per emmagatzemar les dades d'entrada i de sortida, per defecte, la mida és de 64, per una capacitat de 256 bytes per FIFO, és a dir, de 64*32 bits = 2048 bits.

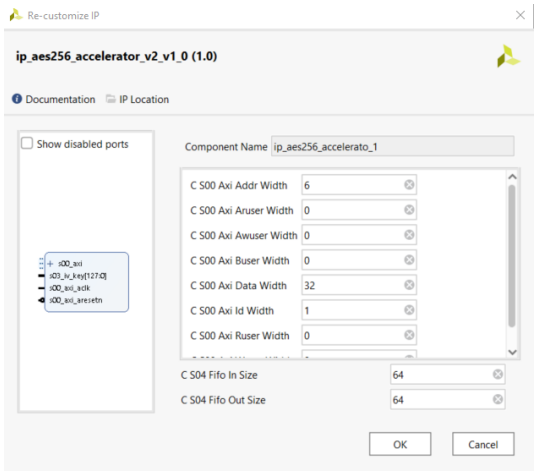

Figura 13. Captura del configurador de paràmetres del Bloc IP.

4.5 Generació d'un sistema amb processador ARM Còrtex-A9

En aquesta etapa es persegueix l'objectiu de passar el treball fet en les etapes anteriors a dins de la FPGA de Xilinx Zynq-7000 SoC ZC702.

Un cop es té el bloc IP generat en l'apartat anterior, es procedeix a generar el sistema embegut en un processador ARM Còrtex-A9 i bus AXI4. Per fer-ho, es fa ús de diversos blocs IP preconstruïts que proporcionen funcionalitats específiques, com l'AXI SmartConnect, que permet la connexió d'un o més dispositius mestres amb memòria AXI amb un o més dispositius esclaus amb memòria. Aquests blocs IP s'utilitzen per accelerar el desenvolupament i millorar l'eficiència del disseny.

La interconnexió de blocs IP es fa en el procés d'IP Integrator de Vivado mitjançant la creació d'un "Block Design". S'hi afegeixen els blocs IP necessaris i es connecten als senyals i ports necessaris del disseny. És important assegurar que els blocs IP estan connectats correctament i configurats de manera adequada per garantir que el disseny funcioni correctament.

En últim lloc, es realitza la validació del "Block design", i consisteix a verificar que les interconnexions i els paràmetres

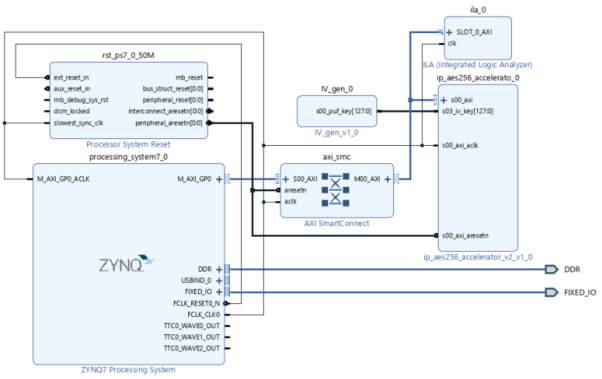

Figura 14. Disseny del bloc AES en l'entorn de Vivado.

de configuració dels diferents blocs IP són correctes i consistents. Per fer-ho, duem a terme la validació mitjançant el motor de Vivado, aquest s'encarrega de validar que totes les connexions són vàlides, i que no ha quedat cap port important d'un bloc IP sense connectar correctament. Un cop s'ha fet la interconnexió de blocs IP i s'ha validat, el disseny estarà llest per passar al procés de síntesi.

4.5.1 Síntesi

La síntesi és el procés de convertir un disseny escrit en un llenguatge HDL (com Verilog o VHDL) en un disseny lògic que es pot implementar en un dispositiu FPGA o ASIC. En aquest cas, es fan ús de tots els fitxers HDL dels blocs IP que s'han interconnectat en l'apartat anterior.

Durant la síntesi, Vivado pren com a entrada un disseny escrit en un llenguatge HDL i genera un fitxer netlist, que és una representació lògica del disseny. El fitxer netlist conté informació descriptiva sobre l'estructura, les portes lògiques, els registres i les connexions electròniques del disseny, i és utilitzat com a entrada per al procés de P&R o síntesi física.

La síntesi també genera informes d'utilització de recursos i anàlisi de temps per ajudar a verificar si el disseny s'ajusta al dispositiu FPGA i si compleix els requisits de temps. Si cal, es poden fer canvis en el disseny HDL abans de continuar amb el P&R.

4.5.2 Implementació

La implementació és un procés complex i que pot consumir molt de temps, però Vivado ofereix diverses eines i opcions per ajudar a optimitzar i automatitzar el procés, accelerant el temps de disseny i millorant l'eficiència. Pren com a entrada el fitxer netlist i genera un fitxer bitstream que pot ser carregat en un dispositiu FPGA. La implementació és el procés final en el flux de disseny de Vivado, i converteix el disseny lògic en una implementació física sobre el dispositiu FPGA. El procés d'implementació comença després de la síntesi i el procés es divideix en tres passos principals:

- Place & Route (P&R): En aquest pas es col·loquen els blocs lògics al dispositiu FPGA i es rutegen les connexions entre ells. S'optimitzen els recursos del dispositiu per assegurar que el disseny s'ajusti a la FPGA i es compleixin els requisits de temps especificats.
- Timing Closure: En aquest pas es verifica que el disseny compleixi els requisits de temps. Si cal, es realitzen canvis per complir els temps de propagació.
- Generació de bitstream: En aquest últim pas es genera el fitxer bitstream, que conté la informació necessària per configurar la FPGA i posar en marxa el disseny.

4.6 Codificació d'un programa

Aquesta última etapa té l'objectiu de posar a prova el sistema mitjançant la codificació d'un programa. Per això, farem ús de l'entorn Vitis, que ja s'ha explicat anteriorment, i del llenguatge de programació C (figura 15).

Un cop codificat aquest programa s'ha executat en la FPGA per emular la IP, connectant aquest dispositiu a la host. Gràcies a l'ILA (Integrated Logic Analyzer) interconnectat en el "Design Block" s'ha pogut monitoritzar i verificar el correcte funcionament dels senyals.

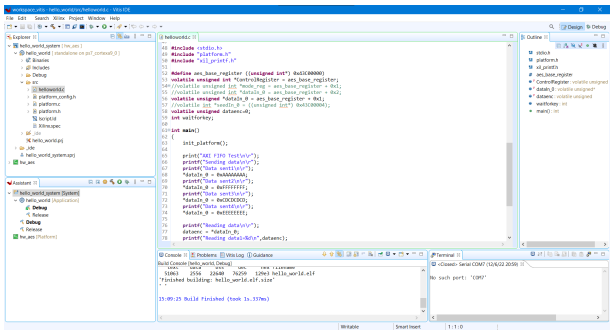

Figura 15. Entorn Vitis amb el codi en C per provar la placa Zynq-7000.

5 RESULTATS

5.1 Análisis de Rendiment

Es fa una anàlisi via simulació per poder quantificar els temps obtinguts després de la integració de la interfície AXI4-Full. Aquests temps inclouran les escriptures i lectures de dades que es produeixen sobre el bus per arribar a les FIFO.

Les dades que es mostren a continuació en la taula corresponen als resultats obtinguts en execucions a freqüència de 100 MHz amb un cicle de treball del 50%:

Procés	Mode	Clock Cycles
Gen. de Clau	Normal	243
Gen. de Clau	Ràfega	219
Encriptació	Normal	348
Encriptació	Ràfega	336
Desencriptació	Normal	387
Desencriptació	Ràfega	375

Com era d'esperar, els temps milloren. Exactament, són 12 cicles de rellotge per a un paquet de 4 bytes, a mesura que augmenta el nombre de paquets en una mateixa fase d'adreçament la millora es fa més patent, per exemple, en el procés de generació de clau, la millora és de 24 cicles de rellotge per l'escriptura d'un paquet de 8 bytes, per tant, tenim que per cada escriptura d'un byte en mode ràfega

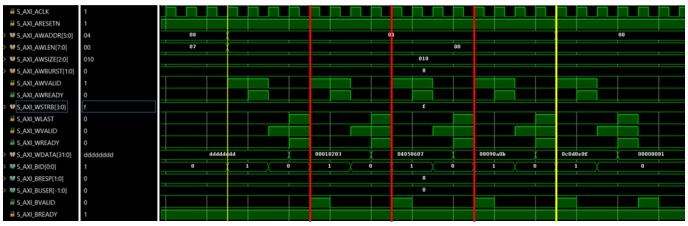

Figura 16. Escriptura en mode normal.

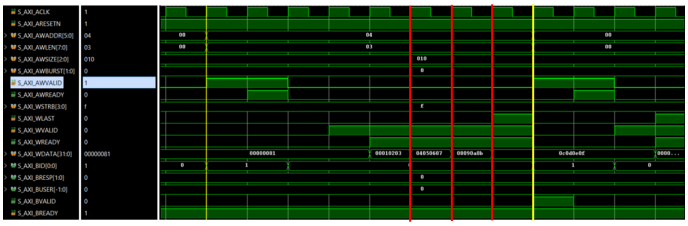

Figura 17. Escriptura en mode ràfega.

es produeix una millora de 3 cicles de rellotge respecte de l'escriptura normal.

En la primera imatge (vegeu figura 16) es pot observar que per cada escriptura es produeix la fase d'adreçament que fa que es consumeixin més cicles de rellotge per a un paquet de 4 bytes. En canvi, l'escriptura en mode ràfega (vegeu figura 17), el mode d'adreçament es realitza només un cop, i a continuació per cada cicle de rellotge es produeix una escriptura.

Les especificacions d'AMBA [14] per l'AXI4-Full, limiten l'ús de la ràfega fins a un màxim de 256 escriptures o lectures per cada fase d'adreçament. Per tant, podem aprofitar aquesta millora per aplicacions de fins a 64 paquets de 4 bytes per carregar la FIFO_IN amb dades, tant per a encriptar com per a desencriptar.

5.2 Slice Logic Distribution

Un cop realitzada la Síntesi i Implementació mitjançant Vivado, podem analitzar el volum que ocupa el component que hem desenvolupat. El nucli IP AES-256 s'ha dissenyat perquè fos el més petit possible, tenint en compte que el Zynq7000 ZC702 compta amb 203.600 components disponibles, només són necessaris per emular la IP Core poc més de 15.000 components.

Com es pot veure a la figura 18, s'usen més Slice Registers que la resta de components en total. Els Slice registers són components lògics que contenen Flip-Flops i Latches, però totes les dades associades als Slice Registers poden entendre's com a Flip-Flops.

Name	Slice LUTs (53200)	Slice Registers (106400)	F7 Muxes (26600)	F8 Muxes (13300)	Slice (13300)	LUT as Logic (53200)
ip_aes256_accelerato_0 (aes_v2_bd_	6829	9257	648	280	3313	6829
inst (aes_v2_bd_ip_aes256_accele	6829	9257	648	280	3313	6829
myip_axififo_v1_S00_AXI_inst (	6829	9257	648	280	3313	6829
DEVICE_AES256 (aes_v2_bc	5504	6268	392	152	2273	5504
FIFO_DATA_IN_1 (aes_v2_b	304	190	0	0	142	304
FIFO_DATA_OUT_1 (aes_v2	853	2188	256	128	820	853
FIFO_SEED (aes_v2_bd_ip_a	123	531	0	0	298	123

Figura 18. Slice Logic Distribution del IP Core AES-256.

Els components Slice Registers i LUT as Logic són els més utilitzats pel bloc IP. No obstant això, quan es tracta de disponibilitat, els dissenyadors han de tenir en compte el marge en l'ús de Slices a l'hora de fer servir aquesta IP Core com un component dels seus projectes.

5.3 Bloc IP Accelerador Hardware AES-256

5.3.1 *Codi IP Accelerador Hardware AES-256*

Tots els elements desenvolupats en aquest treball es poden trobar al següent repositori de GitHub: https://github.com/RedaCB/project_aes256

5.3.2 *Datasheet IP Accelerador Hardware AES-256*

Per a més detall, consultar l'annex.

5.4 Integració de Bloc IP PUF

Un dels objectius d'aquest treball, consistia en la implementació d'una funció física no clonable (PUF) que servís d'entrada per al bloc IP i testar-ho al dispositiu Zynq-7000 ZC702. Un PUF és un circuit físic que genera una resposta única i intrínseca, basada en les variacions aleatòries dels processos de fabricació, i que s'utilitza com una clau secreta per garantir la seguretat del sistema. Per tant, un bon candidat que serveix com a entrada pel port "IV_key" i operar en mode CBC amb la màxima seguretat.

El dispositiu Zynq-7000 ZC702 que combina un processador ARM Còrtex-A9 amb una FPGA programable ofereix una àmplia gamma de recursos de hardware i software, el que és adequat per a una varietat d'aplicacions, inclosa la seguretat.

No obstant això, la integració d'un PUF al dispositiu Zynq-7000 ZC702 va presentar diversos desafiaments i limitacions. En primer lloc, el PUF que es volia dissenyar per aquest projecte, basat en els retards de les interconnexions i que permetia generar un clau únic a causa dels processos de fabricació que fan únics aquestes respostes, estava condicionat a dissenyar un Place&Route manual per fer la Síntesi i Implementació, una qüestió que escapava de l'abast d'aquest projecte. A més, es va descobrir que les restriccions de disseny del dispositiu limitaven la capacitat d'implementar un PUF de forma eficient. Per exemple, el dispositiu Zynq-7000 ZC702 no té una font d'alimentació independent per al PUF, el que podria afectar l'estabilitat i la precisió de la resposta del circuit. També s'ha trobat problemes de compatibilitat amb el sistema operatiu i les llibreries de programari existents, el que dificultava la integració del PUF al sistema, de fet Xilinx no ofereix cap solució comercial de PUF per la placa.

5.5 Llibreria AES256lib

La biblioteca desenvolupada (fitxers AES256lib.h i AES256lib.c) ajuda a proporcionar un conjunt de funcions que poden ser utilitzades per diferents programes per fer tasques específiques sobre el Bloc IP AES-256. Aquestes funcions es compilen i es vinculen amb el programa que les usa permetent al programador evitar la necessitat d'escriure codi per fer tasques comunes.

A continuació es detalla una llista de les funcions que es poden trobar en la llibreria:

- Enviar dada de 32 bits per esriure en la FIFO_IN:

```
int send_data(unsigned int data);
```

- Enviar dada de 32 bits per esriure en la FIFO_SEED:

```
int send_seed(unsigned int data);
```

- Llegir dada de 32 bits procedent de la FIFO_OUT:

```
unsigned int read\_data();
```

- Llegir el resgistre d'estat:

```
unsigned int read_reg_status();
```

- Llegir el resgistre de control:

```
unsigned int read_reg_control();
```

- Activar la encriptació:

```
int active_encryption();
```

- Activar la desencriptació:

```
int active_desencryption();
```

- Activar la generació de clau:

```
int active_key_gen();
```

- Activar el mode de funcionament AES-256 ECB:

```
int acive_mode_ecb()
```

- Activar el mode de funcionament AES-256 CBC:

```
int acive_mode_cbc();
```

Els principals beneficis en desenvolupar i utilitzar biblioteques són la reutilització de codi, millora l'eficiència al programador, reducció d'errors en l'ús, facilitat en el manteniment, i es pot compartir amb altres persones, permetent una major col·laboració i millora a la comunitat de desenvolupament.

6 Conclusions & Treball Futur

En conclusió, en aquest treball s'ha presentat un bloc IP per a l'algorisme AES-256 amb els següents resultats:

- S'ha dotat amb l'interfície de bus AXI4-Full.
- Permet treballar en modes ECB i CBC.
- El bloc IP s'ha dissenyat i validat utilitzant eines de disseny de circuits i programació de FPGA en un dispositiu Zynq-7000 ZC702 amb processament ARM Còrtex A-9, i a més, compleix els estàndards de seguretat AES.
- S'ha desenvolupat una biblioteca en C, per a poder interactuar amb el bloc d'una forma més senzilla.
- S'ha produït un Datasheet funcional i tècnic per a poder fer ús d'aquest bloc IP en altres aplicacions.

Els resultats obtinguts demostren la viabilitat d'implementar l'algorisme AES-256 en modes ECB i CBC en un dispositiu Zynq-7000 ZC702 mitjançant un bloc IP, oferint una alternativa d'alta velocitat per a aplicacions de comunicacions segures.

En el futur, s'espera que el camp dels PUF continuï evolucionant per abordar algunes de les limitacions actuals. Un dels principals objectius serà reduir la variabilitat a les respostes del PUF. Això es podria assolir mitjançant la millora de les tècniques de fabricació per reduir les variacions físiques, o mitjançant l'ús d'algorismes d'aprenentatge automàtic per compensar la variabilitat en les respostes.

D'altra banda, es podria investigar la possibilitat d'implementar una alternativa semblant a un PUF, juntament amb el bloc IP AES-256 per millorar-ne encara més la seguretat del sistema.

Per a treballs futurs relacionats amb el bloc IP, es podria considerar la implementació d'altres modes d'operació AES, com ara el CTR, per ampliar les possibilitats de seguretat del sistema. A més, es podria dur a terme una anàlisi de seguretat més detallada, avaluant la resistència a atacs criptoanalítics, per garantir la seguretat del sistema a llarg termini. També es podria investigar la possibilitat d'integrar el bloc IP en un sistema real, per avaluar-ne el rendiment en un entorn d'aplicació real i comparar-lo amb altres solucions de seguretat existents, tant comercials com de codi obert, obtenint un benchmark que permeti identificar noves oportunitats de millora.

En general, hi ha diverses àrees de recerca per al desenvolupament futur, i el bloc IP presentat en aquest treball és una base sòlida per continuar investigant i millorant la seguretat en sistemes encastats.

Agreïments

L'autor vol agrair a totes les persones que han fet possible aquest treball. Començant per la meva mare, persona que m'ha acompanyat durant tot el meu camí. Al meu pare, per tot l'esforç invisible. Una menció especial al supervisor del treball per tot el sobreesforç, i al tutor per l'oportunitat. Moltes gràcies.

Referències

[1] Rosa Fernández de Statista. Dispositivos conectados (internet de las cosas) a nivel mundial de 2019 a 2030, nov 2021.
[2] PriceWaterhouseCoopers. 25ª encuesta mundial de ceos, dec 2022.
[3] KPMG. Kpmg 2021 ceo outlook pulse survey, mar 2021.
[4] Andrew Waterman, Yunsup Lee, Rimas Avizienis, David A. Patterson, and Krste Asanović. The risc-v instruction set manual volume ii: Privileged architecture version 1.9. Technical Report UCB/EECS-2016-129, EECS Department, University of California, Berkeley, Jul 2016.
[5] Morris Dworkin, Elaine Barker, James Nechvatal, James Foti, Lawrence Bassham, E. Roback, and James Dray. Advanced encryption standard (aes), nov 2001.
[6] A.C. Canut. Aes256 ecb hardware accelerator with axi-lite, jun 2022.
[7] Vivado Design Suite. Advanced encryption standard (aes) engine v1.1 - logicore ip product guide, apr 2022.
[8] Shamir. Adi (Cambridge) Ronald L. (Belmont) and Adleman. Leonard M. (Arlington). The original rsa patent as filed with the u.s. patent office by rivest, dec 1977.
[9] Jorge Guajardo. *Physical Unclonable Functions (PUFs)*, pages 929–934. Springer US, Boston, MA, 2011.
[10] Charles Herder, Meng-Day (Mandel) Yu, Farinaz Koushanfar, and Srinivas Devadas. Physical unclonable functions and applications: A tutorial. *Proceedings of the IEEE*, 102:1126–1141, 08 2014.
[11] Vaibbhav Taraate. *Digital logic design using verilog: Coding and RTL synthesis (1st ed.)*. Springer, Pune - India, 2016.
[12] Sara L. Harris, David M. & Harris. *Digital Design and Computer Architecture*. Morgan Kaufmann, Burlington - USA, 2012.
[13] CrypToolOnline. Aes (step-by-step), 1998.
[14] ARM Limited. Amba® axi™ and ace™ protocol specification - axi3™, axi4™, and axi4-lite™ ace and ace-lite™. oct 2011.

Annex 1.
Datasheet AES-256 IP Core

Centro Nacional de Microelectrónica CSIC

IP AES-256 Accelerator v2.0

December 20, 2022

Introduction

The IP Core AES-256 Accelerator is a versatile hardware that can be used to implementing the Advanced Encryption Standard (AES)[1] with a 256-bit key in five dynamically selectable modes of operation: Electronic Codebook (ECB) [2], Cipher Block Chaining (CBC)

The mode of operation CBC protects data confidentiality and are widely used in numerous security designs and cryptographic protocols. The IP also supports the ECB mode of operation as a building block for other AES modes of operation, but importantly the standalone use of ECB is not recommended for cryptographically secure applications. The design of IP allows for every individual 128-bit data block (plaintext in encryption mode, ciphertext in decryption mode) to use a different key, a different Initialization Vector (IV), and a different mode of operation3.

The IP has been designed for easy integration with FPGA- and ASIC-based designs in a vendor-agnostic design methodology, and the functionality of IP does not rely on any FPGA manufacturer-specific features.

Features

- **Performance**: Key-generation for algorithm AES-256. Encryption and decryption of data, using algorithm AES-256
- **Standard Compliance**: The IP is fully compliant with both the Advanced Encryption Algorithm (AES) standard, as well as with the ECB, and CBC modes of operation.
- **Performance**: Achieves an impressive throughput in the Gbps range, for example 1.00+ Gbps in Xilinx® Zynq-7000®.
- **Versatility**: The key, initialization vector (IV), and the mode of operation can be dynamically updated for every 128-bit data block.

IP Facts Table					
Core Specifics					
Supported Device Family(1)	Zynq™-7000				
Supported User Interfaces	AXI4				
	Resources				**Frequency**
Configuration	**LUTs**	**FFs**	**DSP Slices**	**Block RAMs**	**Max. Freq.**
Unique	6927	9255	928	3	373.0 MHz
Provided with Core					
Documentation	Product Specification				
Design Files	Verilog, System-Verilog				
Example Design	Not Provided				
Test Bench	Verilog				
Constraints File	Not Provided				
Simulation Model	Not Provided				
Supported S/W Driver	Not Provided				
Tested Design Tools					
Design Entry Tools	Vivado v2021.2				
Simulation	Not Provided				
Synthesis Tools(4)	Not Provided				
Support					
Not provided. Academic reasons.					

Applications

The AES-256 Accelerator core is designed to be used in any application that requires encryption or decryption of data using the AES-256 algorithm.

Functional Description

IP AES-256 Accelerator supports two different AES modes of operation: Electronic Codebook (ECB), and Cipher Block Chaining (CBC). The mode of operation CBC use an internal AES256-ECB block as the encryption/decryption engine, but the internal connectivity between 128-bit data block, initialization vector, and the AES256-ECB block inputs and outputs is different; additionally the modes differ in the interdependencies between successive encryption/decryption rounds.

The high-level flow diagrams of CBC in encryption mode are presented in Figures 1.

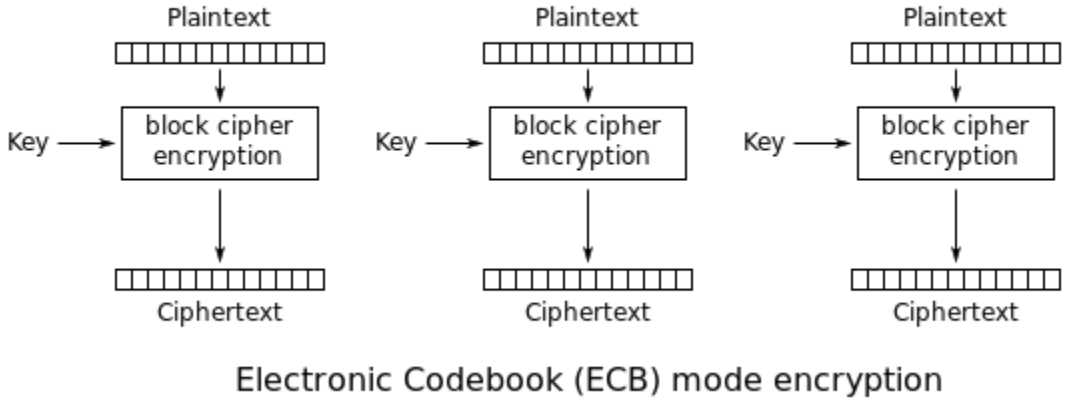

Figure 1. IP in ECB mode of operation, encryption.

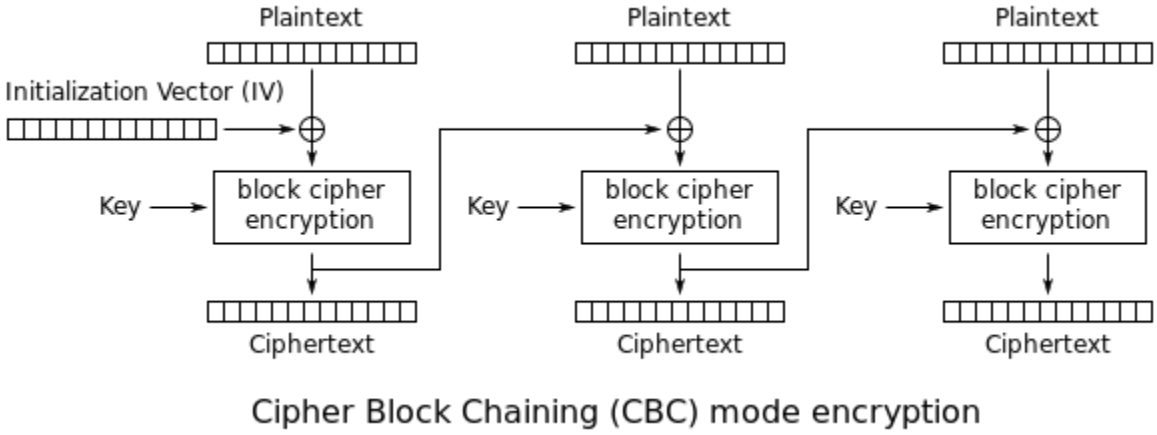

Figure 1. IP in CBC mode of operation, encryption.

IP Core AES-256 Ports and Parameters

Table 1 and Table 2 provide the details about the IP AES-256 Accelerator ports and parameters.

Table 1: **ILA Ports**

Port Name	Direction	Description
CLK	IN	Design clock that clocks all trigger and storage logic.
ARESETN	IN	Design signal that restarts all triggering and storage logic.
IV_KEY	IN	128-bit signal to work in CBC mode.
S_AXI	IN	The bus for transferring data.

IP Core AES-256 Parameters

Table 2: **ILA Parameters**

Parameter Name	Allowable Values	Default Value	Description
component_name	String with A-z, 0-9, and _ (underscore)	aes_v2_0_0	Name of instantiated component
C_FIFO_IN_SIZE	1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072	8192	FIFO IN storage buffer depth. This number represents the maximum number of samples that can be stored at run time for each probe input.
C_FIFO_OUT_SIZE	1024, 2048, 4096, 8192, 16384, 32768, 65536, 131072	8192	FIFO OUT storage buffer depth. This number represents the maximum number of samples that can be stored at run time for each probe input.

IP Core AES-256 Mapping and Register

Table 3 and Table 4 provide the details about the IP AES-256 Accelerator mapping and registers.

Mapping AXI4

Table 2: **ILA Parameters**

Direction	Direction	Description
0x00	WR	Writing data of 32 bits on Control Register.
0x04	WR	Writing data of 32 bits on FIFO IN.
0x08	WR	Writing data of 32 bits on FIFO SEED.
0x00	RD	Reading data of 32 bits on Control Register.
0x04	RD	Reading data of 32 bits on FIFO OUT.
0x0C	RD	Reading data of 32 bits on Status Register.

Control & Status Registers

Bit Position	Register	Description
0	Control	Activate encryption.
1	Control	Activate decryption.
2	Control	Force activation of key generation.
3	Control	Activates the emptying of the FIFO IN.
4	Control	Activates the emptying of the FIFO OUT.
5	Control	Activates the emptying of the FIFO SEED.
6-31	Control	n/a.
0	Status	Indicates encryption status of the Device. By 1, it indicates that the Device is encrypting data.
1	Status	Indicates decryption status on the Device. By 1, indicates the Device is decrypting data.
2	Status	Indicates status of the key to Device. For 1, indicates that the key is generated in the Device. For 0, indicates that no key is generated in Device.
3	Status	By 1, indicates that the FIFO_IN is full. By 0, indicates that the FIFO_IN is not full.
4	Status	By 1, indicates that the FIFO OUT is full. By 0, indicates that the FIFO_OUT is not full.
5	Status	By 1, indicates that the SEED FIFO is full. By 0, indicates that the FIFO_SEED is not full.
6	Status	By 1, indicates that the FIFO_IN is empty. By 0, indicates that the FIFO_IN is not empty.
7	Status	By 1, indicates that the FIFO_OUT is empty. By 0, indicates that the FIFO_OUT is not empty.
8	Status	By 1, indicates that the FIFO_OUT is empty. By 0, indicates that the FIFO_OUT is not empty.
9-31	Status	n/a

Verification

IMB-CNM, CSIC has verified the IP AES-256 Accelerator v2.0 core in a proprietary test environment, using an internally developed bus functional model.

References

[1] Specification for the Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197, 2001.

Support

CNM-IMB (CSIC) not provides technical support for this IP Core product. CNM-IMB (CSIC) cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized or if changes are made to any section of the design.

Revision History

The following table shows the revision history for this document:

Date	Doc Version	Description of Revisions
12/20/2022	1.0	Initial release of the IP AES-256 Accelerator core.