RISC-V.br

Embedded - UFCG

Advanced Encryption Standard - IP

Documentation

Contents

1 Revision History

Date	Version	Description	Author
01/03/2017	1.0	Initial Release	Gabriel Villanova

2 Introduction

O Advanced Encryption Standard (**AES**), é um algoritmo de criptografia de chave simétrica, desenvolvido pelo governo dos EUA e anunciado pelo NIST (Instituto Nacional de Padrões e Tecnologia dos EUA) como U.S. FIPS PUB (FIPS 197). O AES cifra/decifra informações de 128 bits, com chaves de tamanhos variados, podendo ser de 128, 192 ou 256 bits.

O **IP** aqui apresentado, implementa esse algoritmo, oferecendo um hardware dedicado para transmitir e receber dados com segurança, porém, só permite chaves com tamanhos de 128 bits. O IP ainda pode ser configurado para processar vários modos de operação, sendo eles: **ECB, CBC, PCBC, CFB, OFB e CTR**. A comunicação é feita atráves da interface **AMBA AXI4-Lite**. A figura 1 mostra a arquitetura proposta.

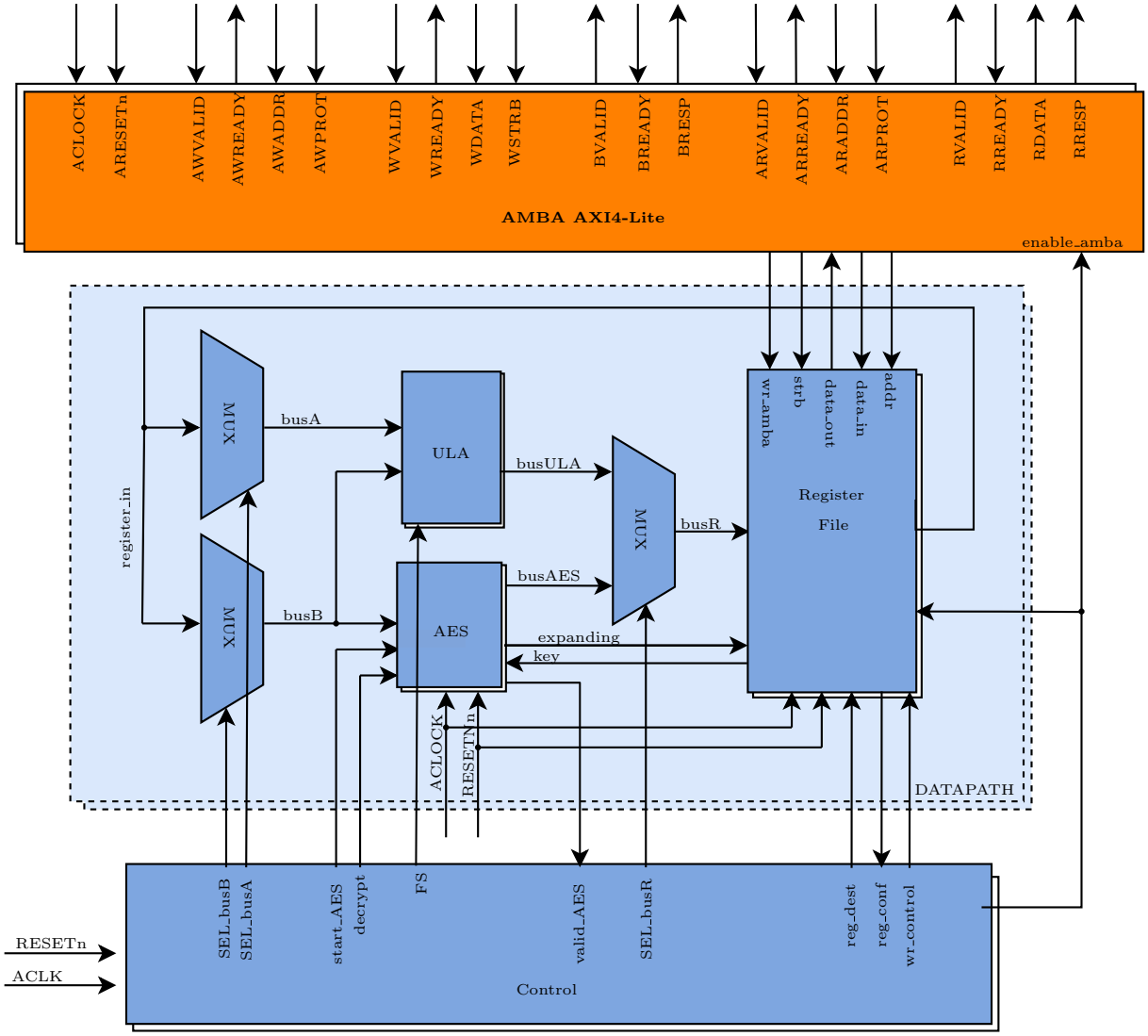

Figure 1: Proposed Architecture of IP

2.1 Basic Concept of the Algorithm

O algoritmo foi desenvolvido seguindo a documentação U.S. FIPS PUB (FIPS 197). A estrutura do algoritmo, em modo de encriptação, pode ser vista na figura 2.

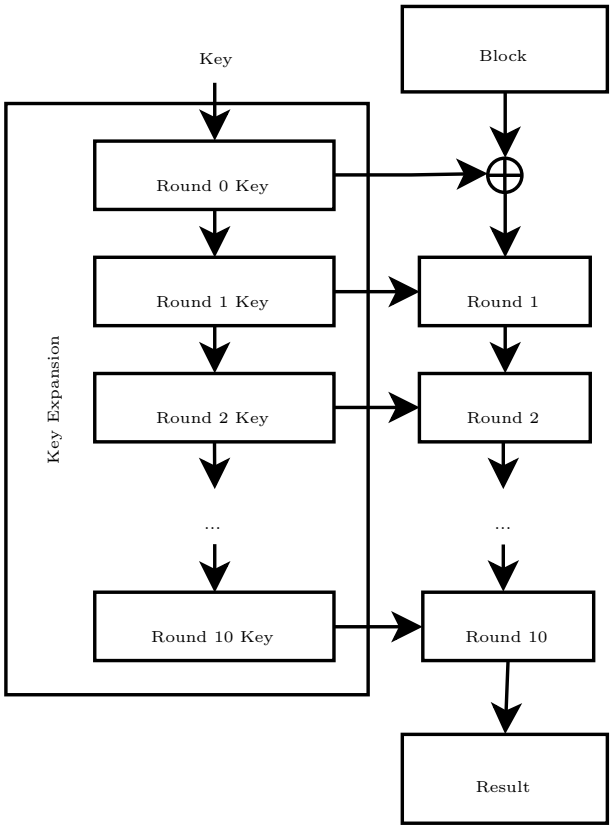

Figure 2: AES Algorithm

Resumidamente, tem-se uma operação de xor com a primeira parte da chave e o bloco de entrada, após isso, os **Rounds** seguintes fazem as operações **Substitute bytes, Shift rows, Mix columns, Add round key**. O processo de decriptação é semelhante, resumindo-se a realizar as operações inversas do algoritmo apresentado. As operações feitas nos Rounds estão detalhadas na na documentação oficial (FIPS 197).

2.2 Modes of Operations

Esse IP contempla 6 modos de operações, **ECB, CBC, PCBC, CFB, OFB e CTR**. Esses modos realizam algumas operações simples, como por exemplo operação de xor com o resultado da encriptação/decriptação com algum outro valor. O objetivo é proteger a informação de possíveis ataques hackers. Cada modo tem suas peculiaridades em relação ao tipo de proteção, ficando ao critério do programador escolher o mais adequado para sua aplicação. O desenvolvimento desses modos foi baseado no material Block Cipher Mode of Operation da Wikipédia.

3 Internal Signals

Para um maior entendimento do processamento do IP, os sinais internos estão todos descritos nas tabelas 1, 2 e 3.

Name	Description	Type	Bits
register_in.r0	Register for **Block**	Unsigned logic	128 bits
register_in.r1	Register for **Initialization Vector** (IV)	Unsigned logic	128 bits
register_in.r2	Register for **Result**	Unsigned logic	128 bits
register_in.r3	Register for **Counter**	Unsigned logic	64 bits
addr	Receives ADDR read from AMBA	Unsigned logic	32 bits
data_in	Receives WDATA read from AMBA	Unsigned logic	32 bits
data_out	Return a value the Regiser File to RDATA	Unsigned logic	32 bits
strb	Receives WSTRB read from AMBA	Unsigned logic	32 bits
wr_amba	Enable or disable write in Regiser File	Bool	1 bit

Table 1: Description of Internal Signals in Register File

Name	Description	Type	Bits
sel_busA	Selects a register_in in busA	Unsigned logic	2 bits
sel_busB	Selects a register_in in busB	Unsigned logic	2 bits
sel_busR	Selects the busULA or busAES in busR	Bool	1 bit
start_AES	Start block AES when 1	Bool	1 bit
valid_AES	Indicates result AES done when 1	Bool	1 bit
decrypt	Indicates decrypt operation when 1, encrypt when 0	Bool	1 bit
expanding	Indicates key expansion operation when 1	Bool	1 bit
key	Register for **Key**	Unsigned logic	128 bits
FS	Function Select of block ULA	Unsigned logic	2 bits
enable_amba	Enable or disable write in Register File by AMBA	Bool	1 bit

Table 2: Description of Internal Signals in Control Block

Name	Description	Type	Bits
busA	Receives a register_in	Unsigned logic	128 bits
busB	Receives a register_in	Unsigned logic	128 bits
busR	Receives result from block ULA or AES	Unsigned logic	128 bits
busULA	Receives result from block ULA	Unsigned logic	128 bits
busAES	Receives result from block AES	Unsigned logic	128 bits
reg_dest	Select destination register to save in busR	Unsigned logic	2 bits
reg_conf	Contains the configuration register	Unsigned logic	32 bits
wr_control	Enable or disable write in Register File	Bool	1 bit

Table 3: Description of Internal Signals in Datapath

4 Memory Map

Nessa seção seram apresentados os registradores do IP.

4.1 Introduction

Os registradores, seus respectivos endereços, usualidade e tipos de acesso podem ser visualizados na tabela 4.

Address	Pseudonym	Use	Access Type	bit done
0x00	key[0]	First part of key (less significant)	Read/Write	1
0x04	key[1]	Second part of key	Read/Write	1
0x08	key[2]	Third part of key	Read/Write	1
0x0C	key[3]	Fourth part of key (most significant)	Read/Write	1
0x10	r0[0]	First part of block (less significant)	Read/Write	1
0x14	r0[1]	Second part of block	Read/Write	1
0x18	r0[2]	Third part of block	Read/Write	1
0x1C	r0[3]	Fourth part of block (most significant)	Read/Write	1
0x20	r1[0]	First part of IV (less significant)	Read/Write	1
0x24	r1[1]	Second part of IV	Read/Write	1
0x28	r1[2]	Third part of IV	Read/Write	1
0x2C	r1[3]	Fourth part of IV (most significant)	Read/Write	1
0x30	r2[0]	First part of result (less significant)	Read/Write	1
0x34	r2[1]	Second part of result	Read/Write	1
0x38	r2[2]	Third part of result	Read/Write	1
0x3C	r2[3]	Fourth part of result (most significant)	Read/Write	1
0x40	r3[0]	First part of Counter (less significant)	Read/Write	1
0x44	r3[1]	Second part of Counter (most significant)	Read/Write	1
0x48	reg_cmd	Command register	Read/Write	1
0x4C	reg_status	Status register	Read Only	X

Table 4: Memory Map AES IP

Todas as palavras binárias dos registradores são organizados como **Little-Endian**, ou seja, extremidade menor primeiro. Para contextualizar, suponha que seja desejado a escrita da chave 00010203_04050607_08090A0B_0C0D0E0F. Nesse caso, deve-se fazer como mostrado nas equações abaixo.

$$key[0] = 03020100 \quad (1)$$

$$key[1] = 07060504 \quad (2)$$

$$key[2] = 0B0A0908 \quad (3)$$

$$key[3] = 0F0E0D0C \quad (4)$$

O acesso para ler a chave é possível, porém, deve-se ter o mais alto nível de privilégio, que segundo o AMBA AXI4-Lite, $ARPROT = 3'bX11$ (veja AMBA AXI and ACE Protocol Specification).

4.2 Command Register

A tabela 5 mostra o registrador que possui a função de definir o modo de operação que ele deve processar e iniciar os cálculos no IP. A tabela 6 detalha o signigicado de cada bit do Command Register.

R/W	[31]	[30:5]	[4]	[3]	[2:0]
R			counter_zero	decrypt	mode
W	start		counter_zero	decrypt	mode

Table 5: Command Register

Command	Modes of Configuration	Operation
start	0	Not start processing in IP
	1	Start processing in IP
counter_zero	0	Maintains value of r3
	1	Reset to zero register r3
decrypt	0	Encryption mode
	1	Decryption mode
mode	000	**ECB**: Eletronic Cipher Block
	001	**CBC**: Cipher Block Chaining
	010	**PCBC**: Propagating Cipher Block Chaining
	011	**CFB**: Cipher Feedback
	100	**OFB**: Output Feedback
	101	**CTR**: Counter
	110	PCBC
	111	CFB

Table 6: Description Bits Commands

4.3 Status Register

O registrador de status retorna informações sobre o processamento do IP, sendo somente leitura (read only). A tabela 8 detalha o significado de cada bit desse registrador.

R/W	[31]	[30:6]	[5:3]	[2]	[1]	[0]
R	done		mode	key_exp	decrypt	encrypt
W						

Table 7: Status Register

5 Examples of Results

Na tabela 9 mostra os resultados de encriptação para cada modo considerando os dois dados de entrada, Key, IV e o valor de Counter, como segue:

Command	Modes of Configuration	Operation
done	0 1	Can't read/write in registers Can read/write in registers
mode	000 001 010 011 100 101 110 111	**ECB**: Eletronic Cipher Block **CBC**: Cipher Block Chaining **PCBC**: Propagating Cipher Block Chaining **CFB**: Cipher Feedback **OFB**: Output Feedback **CTR**: Counter PCBC CFB
key$_exp$	0 1	Key expansion is done Calculating key expansion
decrypt	0 1	- Decryption operation in progress
encrypt	0 1	- Encryption operation in progress

Table 8: Description Bits Commands

$$block\_in1 = 00112233445566778899AABBCCDDEEFF \quad (5)$$

$$block\_in2 = FFEEDDCCBBAA99887766554433221100 \quad (6)$$

$$key = 000102030405060708090A0B0C0D0E0F \quad (7)$$

$$IV = CCCCCCCCAAAAAAAAFFFFFFFFEEEEEEEE \quad (8)$$

$$counter = 0 \quad (9)$$

MODE	First Cipher Block	Second Cipher Block
ECB	69C4E0D86A7B0430D8CDB78070B4C55A	1B872378795F4FFD772855FC87CA964D
CBC	F6217F61B2D50A6AE6791f8C384B1E07	00E6F7C3F089B33AF8D701BEB170AF82
PCBC	F6217F61B2D50A6AE6791f8C384B1E07	1766EA65883AE0BE5DE323B1431CBD54
CFB	79C571F93E6E91590576009881A3AB8E	3F778AF1BA19E580C751137195BCFF23
OFB	79C571F93E6E91590576009881A3AB8E	03219503dd5973187FADC74474E3F047
CTR	79C571F93E6E91590576009881A3AB8E	833441D22AD2BAAEEfAB5EA586AC0DF2

Table 9: Example of Results

6 Programming Model

A FSM da figura 3 ilustra a lógica de funcionamento do IP. Pode-se ver que, após o estado de reset (idle), é possível escrever e ler os registradores enquato o bit de start não for 1. Quando o bit start finalmente contiver o valor 1, o processamento se inicial, o bit done vai para 0 e impossibilida escrita ou leitura nos registradores, salvo o registrador

de status. Acabado o processamento, o IP volta para o estado de bit done igual 1, em que pode-se ler ou escrever dos registradores.

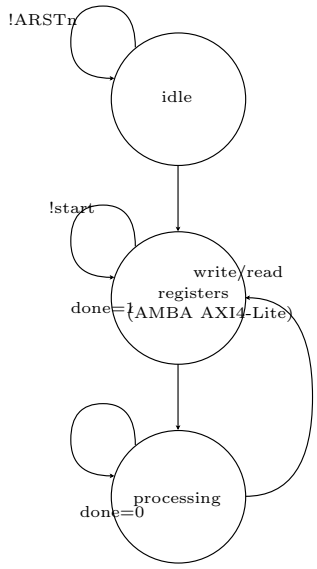

Figure 3: Programming Model in FSM Representation

Considerando os endereços na tabela 4, será mostrado nessa seção, o exemplo de programação em assembly de como usar o modo de operação CFB.

```
1 OUT 0x00,0x0C0D0E0F              ; Escrita dos bits [31 : 0] da chave
2 OUT 0x04,0x08090A0B              ; Escrita dos bits [63 :32] da chave
3 OUT 0x08,0x04050607              ; Escrita dos bits [95 :64] da chave
4 OUT 0x0C,0x00010203              ; Escrita dos bits [127:96] da chave
5
6 OUT 0x10,0x00112233              ; Escrita dos bits [31 : 0] do bloco
7 OUT 0x14,0x44556677              ; Escrita dos bits [63 :32] do bloco
8 OUT 0x18,0x8899AABB              ; Escrita dos bits [95 :64] do bloco
9 OUT 0x1C,0xCCDDEEFF              ; Escrita dos bits [127:96] do bloco
10
11 OUT 0x20,0xCCCCCCCC             ; Escrita dos bits [31 : 0] do IV
12 OUT 0x24,0xAAAAAAAA             ; Escrita dos bits [63 :32] do IV
13 OUT 0x28,0xFFFFFFFF             ; Escrita dos bits [95 :64] do IV
14 OUT 0x2C,0xEEEEEEEE             ; Escrita dos bits [127:96] do IV
15
16 OUT 0x48,0x80000003
17 ; Escrita do bit de start, do modo CFB
18 ; encripta  o no registrador de comando
```

7 Parameters

A tabela 10 contém os valores referenciados no texto.

Parameter	Bits	Value (in hex)
AES_OFFSET	24	0x000000

Table 10: Parameters

8 Reference

[**1**] AMBA AXI and ACE Protocol Specification. ARM Copyright©;
[**2**] FIPS PUB 197, Advanced Encryption Standard (AES)
[**3**] WIKIPEDIA, Block cipher mode of operation