Universitas Indonesia

AES ENCRYPTION HARDWARE ACCELERATOR

Kelompok PA-16

- Fathan Yazid Satriani 2306250560
- Mutia Casella 2306202870
- Muhammad Rafli 2306250730
- Ahmad Malik Ramadhino 2206059370

Daftar Isi

Latar Belakang

Keamanan data menjadi isu penting di era digital, mengingat risiko akses oleh pihak tidak bertanggung jawab. Advanced Encryption Standard (AES) adalah algoritma enkripsi simetris yang banyak digunakan karena keamanan dan efisiensinya. AES mengenkripsi data dalam blok 128-bit melalui beberapa tahap, seperti SubBytes, ShiftRows, MixColumns, dan AddRoundKey, yang diulang berdasarkan panjang kunci (128-bit)

Peforma AES pada perangkat lunak sering terbatas oleh kemampuan perangkat keras. Implementasi AES pada Field-Programmable Gate Array (FPGA) adalah sulusi dengan meningkatkan kecepatan melalui paralelisme dan pipelining.

Deskripsi Proyek

Proyek ini bertujuan untuk mengimplementasikan akselerator enkripsi AES berbasis VHDL yang dijalankan pada FPGA untuk memproses enkripsi dan dekripsi data secara cepat dan efisien. Akselerator ini akan menggunakan teknik paralelisme dan pipelining untuk meningkatkan performa dari setiap operasi yang dilakukan oleh algoritma AES, yakni SubBytes, ShiftRows, MixColumns, dan AddRoundKey. Algoritma ini akan dijalankan dalam 10 ronde untuk kunci 128-bit, dengan input berupa blok plaintext 128-bit dan kunci enkripsi, serta output berupa ciphertext yang terenkripsi.

Desain akselerator ini bertujuan untuk mencapai kinerja yang lebih tinggi dibandingkan implementasi perangkat lunak, serta menyediakan kemampuan untuk melakukan dekripsi menggunakan algoritma invers dari langkah enkripsi.

Tujuan Proyek

- Mendesain dan mengimplementasikan algoritma AES-128 berbasis VHDL pada FPGA.
- Memanfaatkan paralelisme dan pipeline untuk meningkatkan kinerja pada proses enkripsi dan dekripsi..
- Menguji sistem dalam memproses data enkripsi dan dekripsi menggunakan FPGA.
- Meningkatkan pemahaman tentang implementasi algoritma kriptografi pada perangkat keras.

Pembagian Tugas

Fathan Yazid Satriani

- Membuat program bagian Encryption dan Testbench untuk Encryption.
- Membuat program Encryption dalam Bahasa C juga untuk menguji kebenaran hasil dengan program VHDL.
- Membantu menulis dokumentasi pada laporan, powerpoint, dan readme, khususnya untuk penjelasan Encryption.

Mutia Casella

- Membuat program VHDL bagian Decryption dan Testbench untuk Decryption.
- Membantu menulis dokumentasi pada laporan, powerpoint, dan readme.

Muhammad Rafli

- Membuat program VHDL bagian Decryption dan Testbench untuk Decryption.
- Membantu menulis dokumentasi pada laporan, powerpoint, dan readme.

Aplikasi Pendukung

- GitHub
- ModelSim
- Visual Studio Code
- Quartus Prime

Implementasi

Addroundkey

```
for col in 0 to 3 loop
    for row in 0 to 3 loop
        state(row, col) <= state(row, col) xor
                           current_round_key(127-8*(4*col+row) downto 120-8*(4*col+row));
    end loop;
end loop;
```

SubByte

Deskripsi

```
function inv_sub_bytes(state_in: state_array) return state_array is
    variable result : state_array;
begin
    for i in 0 to 3 loop
        for j in 0 to 3 loop
            result(i, j) := INV_SBOX(to_integer(unsigned(state_in(i, j))));
        end loop;
    end loop;
    return result;
end function;
```

Enkripsi

```
function sub_bytes(state_in: state_array) return state_array is
    variable result : state_array;
begin
    for i in 0 to 3 loop
        for j in 0 to 3 loop
            result(i, j) := SBOX(to_integer(unsigned(state_in(i, j))));
        end loop;
    end loop;
    return result;
end function;
```

ShiftRows

Enkripsi

```
function shift_rows(state_in: state_array) return state_array is
    variable result : state_array;
begin
    -- Row 0: tidak bergeser
    result(0, 0) := state_in(0, 0);
    result(0, 1) := state_in(0, 1);
    result(0, 2) := state_in(0, 2);
    result(0, 3) := state_in(0, 3);

    -- Row 1: geser kiri 1
    result(1, 0) := state_in(1, 1);
    result(1, 1) := state_in(1, 2);
    result(1, 2) := state_in(1, 3);
    result(1, 3) := state_in(1, 0);

    -- Row 2: geser kiri 2
    result(2, 0) := state_in(2, 2);
    result(2, 1) := state_in(2, 3);
    result(2, 2) := state_in(2, 0);
    result(2, 3) := state_in(2, 1);

    -- Row 3: geser kiri 3
    result(3, 0) := state_in(3, 3);
    result(3, 1) := state_in(3, 0);
    result(3, 2) := state_in(3, 1);
    result(3, 3) := state_in(3, 2);

    return result;
end function;
```

Deskripsi

```
function inv_shift_rows(state_in: state_array) return state_array is
    variable result : state_array;
begin
    -- Row 0: tidak bergeser
    result(0, 0) := state_in(0, 0);
    result(0, 1) := state_in(0, 1);
    result(0, 2) := state_in(0, 2);
    result(0, 3) := state_in(0, 3);

    -- Row 1: geser kanan 1
    result(1, 0) := state_in(1, 3);
    result(1, 1) := state_in(1, 0);
    result(1, 2) := state_in(1, 1);
    result(1, 3) := state_in(1, 2);

    -- Row 2: geser kanan 2
    result(2, 0) := state_in(2, 2);
    result(2, 1) := state_in(2, 3);
    result(2, 2) := state_in(2, 0);
    result(2, 3) := state_in(2, 1);

    -- Row 3: geser kanan 3
    result(3, 0) := state_in(3, 1);
    result(3, 1) := state_in(3, 2);
    result(3, 2) := state_in(3, 3);
    result(3, 3) := state_in(3, 0);

    return result;
end function;
```

MixColumns

Enkripsi

```
function mix_columns(state_in: state_array) return state_array is
    variable result : state_array;
begin
    for col in 0 to 3 loop
        result(0, col) := gf_mult(X"02", state_in(0, col)) xor
                          gf_mult(X"03", state_in(1, col)) xor
                          state_in(2, col) xor
                          state_in(3, col);

        result(1, col) := state_in(0, col) xor
                          gf_mult(X"02", state_in(1, col)) xor
                          gf_mult(X"03", state_in(2, col)) xor
                          state_in(3, col);

        result(2, col) := state_in(0, col) xor
                          state_in(1, col) xor
                          gf_mult(X"02", state_in(2, col)) xor
                          gf_mult(X"03", state_in(3, col));

        result(3, col) := gf_mult(X"03", state_in(0, col)) xor
                          state_in(1, col) xor
                          state_in(2, col) xor
                          gf_mult(X"02", state_in(3, col));
    end loop;
    return result;
end function;
```

Deskripsi

```
function inv_mix_columns(state_in: state_array) return state_array is
    variable result : state_array;
begin
    for col in 0 to 3 loop
        result(0, col) := gf_mult(X"0e", state_in(0, col)) xor
                          gf_mult(X"0b", state_in(1, col)) xor
                          gf_mult(X"0d", state_in(2, col)) xor
                          gf_mult(X"09", state_in(3, col));

        result(1, col) := gf_mult(X"09", state_in(0, col)) xor
                          gf_mult(X"0e", state_in(1, col)) xor
                          gf_mult(X"0b", state_in(2, col)) xor
                          gf_mult(X"0d", state_in(3, col));

        result(2, col) := gf_mult(X"0d", state_in(0, col)) xor
                          gf_mult(X"09", state_in(1, col)) xor
                          gf_mult(X"0e", state_in(2, col)) xor
                          gf_mult(X"0b", state_in(3, col));

        result(3, col) := gf_mult(X"0b", state_in(0, col)) xor
                          gf_mult(X"0d", state_in(1, col)) xor
                          gf_mult(X"09", state_in(2, col)) xor
                          gf_mult(X"0e", state_in(3, col));
    end loop;
    return result;
end function;
```

GfMult

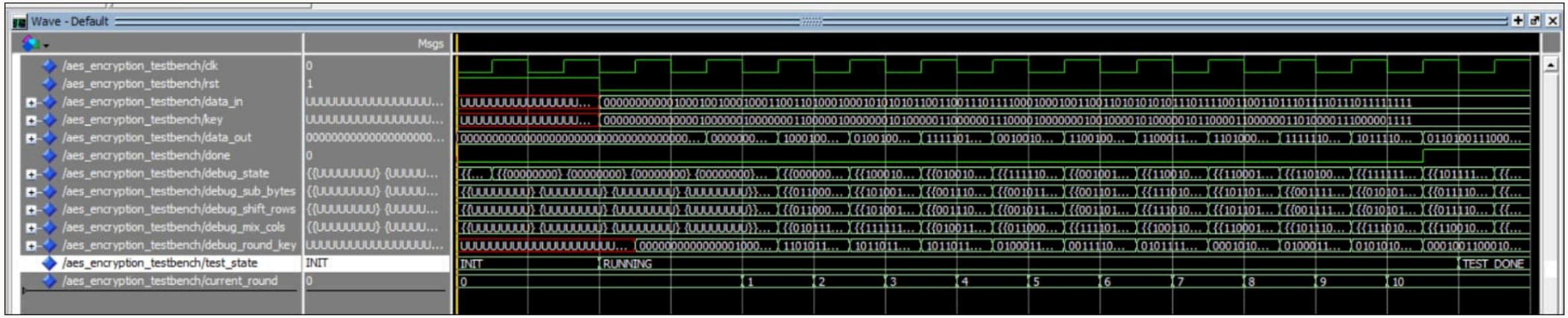
```
function gf_mult(a, b: STD_LOGIC_VECTOR(7 downto 0)) return STD_LOGIC_VECTOR is
    variable p : STD_LOGIC_VECTOR(7 downto 0);
    variable hi_bit : STD_LOGIC;
    variable temp : STD_LOGIC_VECTOR(7 downto 0);
 begin
    p := (others => '0');
    temp := a;

    for i in 0 to 7 loop
        if (b(i) = '1') then
            p := p xor temp;
        end if;
        hi_bit := temp(7);
        temp := temp(6 downto 0) & '0';
        if (hi_bit = '1') then
            temp := temp xor X"1B";  -- Reduction polynomial
        end if;
    end loop;
    return p;
 end function;
```

Testing

Proses pengujian dilakukan untuk memastikan bahwa proses enkripsi dan dekripsi berjalan sesuai prosedur serta memenuhi persyaratan fungsi dan kinerja yang diinginkan. Pada proyek ini, pengujian dilakukan menggunakan tiga metode utama, yaitu wave test (ModelSim), synthesize test (Quartus), dan pengujian berbasis text file.

Hasil Uji Coba

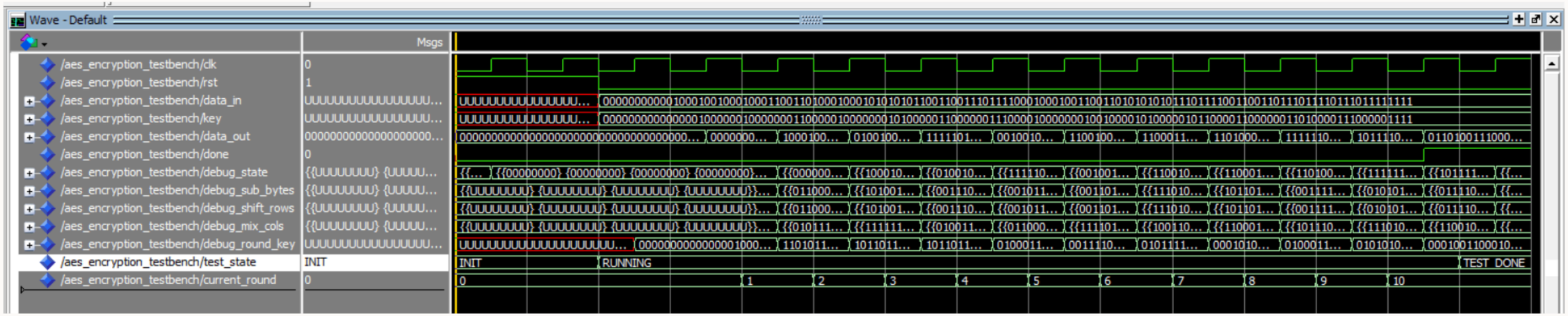
- Enkripsi



```
=== AES Encryption Test ===
Plaintext: 00112233445566778899aabbccddeeff
Key      : 000102030405060708090a0b0c0d0e0f
Expected : 69c4e0d86a7b0430d8cdb78070b4c55a

----------------------------------------
Round 0
Current state:
  State     : 00000000000000000000000000000000
  SubBytes  : 00000000000000000000000000000000
  ShiftRows : 00000000000000000000000000000000
  MixCols   : 00000000000000000000000000000000
  RoundKey  : 000102030405060708090a0b0c0d0e0f

----------------------------------------
Round 1
Current state:
  State     : 004080c0105090d02060a0e03070b0f0
  SubBytes  : 6309cdbaca536070b7d0e0e10451e78c
  ShiftRows : 6309cdba536070cae0e1b7d08c0451e7
  MixCols   : 5f57f71d72f5beb964bc3bf91592291a
  RoundKey  : d6aa74fdd2af72fadaa678f1d6ab76fe

----------------------------------------
Round 2
Current state:
  State     : 89852dcbd85a181210ce438fe868d8e4
  SubBytes  : a797d81f61beadc9ca8b1a739b456169
  ShiftRows : a797d81fbeadc9611a73ca8b699b4561
  MixCols   : ff31647787d8513a966a51d08451fa09
  RoundKey  : b692cf0b643dbdf1be9bc5006830b3fe

----------------------------------------
Round 3
Current state:
  State     : 4955da1f15e5ca0a59d794638fa0faf7
  SubBytes  : 3bfc57c059d97467cb0e22fb73e02d68
  ShiftRows : 3bfc57c0d974675922fbcb0e6873e02d
  MixCols   : 4cf72c539c713f4d1ef086f266768e56
  RoundKey  : b6ff744ed2c2c9bf6c590cbf0469bf41

----------------------------------------
Round 4
Current state:
  State     : fa25405763b366246a398a4d28c93117
  SubBytes  : 2d3f095bfb6d333602127ee334ddc7f0
  ShiftRows : 2d3f095b6d3336fb7ee30212f034ddc7
  MixCols   : 63fc97758553be47b78d47d69ff98e91
  RoundKey  : 47f7f7bc95353e03f96c32bcfd058dfd

----------------------------------------
Round 5
Current state:
  State     : 24696e887266d24240b3755b23fa326c
  SubBytes  : 36f99fc44033b52c096d9d39262d2350
  ShiftRows : 36f99fc433b52c409d39096d50262d23
  MixCols   : f432751dbce5f1d0d454d63b54d0c53c
  RoundKey  : 3caaa3e8a99f9deb50f3af57adf622aa

----------------------------------------
Round 6
Current state:
  State     : c89b25b0167a022677c97919bc3b9296
  SubBytes  : e8143fe747da77f7f5ddb6d465e24f90
  ShiftRows : e8143fe7da77f747b6d4f5dd9065e24f
  MixCols   : 98006b8e16f82c5aee7f04d074559c36
  RoundKey  : 5e390f7df7a69296a7553dc10aa31f6b

----------------------------------------
Round 7
Current state:
  State     : c6f7cc842f5e79f9e1ed39cf09c35d5d
  SubBytes  : b4684b5f1558b699f855128a012e4c4c
  ShiftRows : b4684b5f58b69915128af8554c012e4c
  MixCols   : c59af0987e9b5fc61cd24b341586e039
  RoundKey  : 14f9701ae35fe28c440adf4d4ea9c026

----------------------------------------
Round 8
Current state:
  State     : d179b4d687c4556f6c3094f40f0aad1f
  SubBytes  : 3eb68df6171cfca8500422bf766795c0
  ShiftRows : 3eb68df61cfca81722bf5004c0766795
  MixCols   : baa1d55fa0f951413db52c4de76eba23
  RoundKey  : 47438735a41c65b9e016baf4aebf7ad2

----------------------------------------
Round 9
Current state:
  State     : fd0535f1e3e547febad09637d2d74ef1
  SubBytes  : 546b96a111d9a0bbf470909ab50e2fa1
  ShiftRows : 546b96a1d9a0bb11909af470a1b50e2f
  MixCols   : e9021b35f730f23c4e20cc21ecf6f2c7
  RoundKey  : 549932d1f08557681093ed9cbe2c974e

----------------------------------------
Round 10
Current state:
  State     : bdf20b8b6eb561107c7721b63d9e6e89
  SubBytes  : 7a892b3d9fd5efca10f5fd4e270b9fa7
  ShiftRows : 7a892b3dd5efca9ffd4e10f5a7270b9f
  MixCols   : ca4a08aa70b99f83bc93dce9f36fb108
  RoundKey  : 13111d7fe3944a17f307a78b4d2b30c5

========== Encryption Complete ==========
Final Results:
Output   : 69c4e0d86a7b0430d8cdb78070b4c55a
Expected : 69c4e0d86a7b0430d8cdb78070b4c55a
Status: PASS
```

Hasil Uji Coba

- Dekripsi



=== AES Decryption Test ===
Ciphertext: 69c4e0d86a7b0430d8cdb78070b4c55a
Key : 000102030405060708090a0b0c0d0e0f
Expected : 00112233445566778899aabbccddeeff

Round 10
Current state:
State : 00000000000000000000000000000000
SubBytes : 52525252525252525252525252525252
ShiftRows : 52525252525252525252525252525252
MixCols : 52525252525252525252525252525252
RoundKey : 00000000000000000000000000000000

Round 9
Current state:
State : c08bf90cf671eaed85aa6ceb7be0b9c6
SubBytes : 1fce6981d62cbb536762b83c03a0dbc7
ShiftRows : 1fce698153d62cbbb83c6762a0dbc703
MixCols : ef5d57e1bd6bb771e04a88b4e6838d7f
RoundKey : 13111d7fe3944a17f307a78b4d2b30c5

Round 8
Current state:
State : 611597bc3a539a0fee81fd418df064db
SubBytes : d82f8578a25037fb999121f8b4178c9f
ShiftRows : d82f8578fba2503721f89991178c9fb4
MixCols : ea2f557874ded94d3911f445b219ab1a
RoundKey : 549932d1f08557681093ed9cbe2c974e

Round 7
Current state:
State : c3e9dde58661cc1ed9d44ac587013156
SubBytes : 33ebc92adcd827e9e5195c07ea092eb9
ShiftRows : 33ebc92ae9dcd8275c07e519092eb9ea
MixCols : a0f648b3ae7a2e0dca26fe674bb4d527
RoundKey : 47438735a41c65b9e016baf4aebf7ad2

Round 6
Current state:
State : 1c1906a6dc37844bd4038559b09957f1
SubBytes : c48ea5c593b24fcc19d56715fcf9da2b
ShiftRows : c48ea5c5cc93b24f671519d5f9da2bfc
MixCols : 24f65b461fa3bb56b0d2ad0b1d5568b8
RoundKey : 14f9701ae35fe28c440adf4d4ea9c026

Round 5
Current state:
State : 5da5fa1d4a9c5cbaf7e9f34ecc83c7ec
SubBytes : 8d2914de5c1ca7c026eb7eb627413183
ShiftRows : 8d2914dec05c1ca77eb626eb41318327
MixCols : 39aa7e925a1f90632eca0dd63f8d4e92
RoundKey : 5e390f7df7a69296a7553dc10aa31f6b

Round 4
Current state:
State : b6808c68b5754868c1b668cd468a3768
SubBytes : 793af0f7d23fd4f7dd79f78098cfb2f7
ShiftRows : 793af0f7f7d23fd4f780dd79cfb2f798
MixCols : 017f4d3766bd6136f86ab126297278e5
RoundKey : 3caaa3e8a99f9deb50f3af57adf622aa

-Round 3
Current state:
State : 6cda953f9838d3e510f9a858b40c0667
SubBytes : b87aad25e276a92a7c696f5ec681a50a
ShiftRows : b87aad252ae276a96f5e7c6981a50ac6
MixCols : 4797db2d8bd03bb568754b27d851069c
RoundKey : 47f7f7bc95353e03f96c32bcfd058dfd

Round 2
Current state:
State : 2a5fa6fdaeab0cd4f60ec140ceb20661
SubBytes : 9584c521be0e8119d6d7dd72ec3ea5d8
ShiftRows : 9584c52119be0e81dd72d6d73ea5d8ec
MixCols : 97c1b3a02c9567ad01f6317bd54f20ed
RoundKey : b6ff744ed2c2c9bf6c590cbf0469bf41

Round 1
Current state:
State : 3a64060d78cb2bfbe813c0962ebac310
SubBytes : a28ca5f3c1590b63c8821f35c3c0337c
ShiftRows : a28ca5f363c1590b1f35c882c0337cc3
MixCols : 17e1d7c1cff2ca0996ce2ad950967fa8
RoundKey : b692cf0b643dbdf1be9bc5006830b3fe

Round 0
Current state:
State : 17e9c7d9cdf8d81392c23ec5569869b7
SubBytes : a28ca5f3c1590b63c8821f35c3c0337c
ShiftRows : a28ca5f363c1590b1f35c882c0337cc3
MixCols : 17e1d7c1cff2ca0996ce2ad950967fa8
RoundKey : d6aa74fdd2af72fadaa678f1d6ab76fe

========== Decryption Complete ==========
Final Results:
Output : f4b4b4015cb6a8195b9c808a22de55b0
Expected : 00112233445566778899aabbccddeeff
Status: FAIL

Kesimpulan

AES_ENKRIPSI

Proses enkripsi AES-128 berhasil diterapkan menggunakan FPGA dengan hasil yang sesuai dengan ekspektasi. Teknik paralelisme dan pipelining meningkatkan efisiensi, menghasilkan ciphertext yang akurat sesuai pengujian.

AES_DESKRIPSI

Proses dekripsi mengalami kegagalan akibat kesalahan urutan round key, sehingga plaintext yang dihasilkan tidak sesuai.

Referensi

[1] M. Dworkin, “Recommendation for Block Cipher Modes of Operation: Methods and Techniques,” NIST Special Publication 800-38A, 2001. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf

[2] National Institute of Standards and Technology (NIST), “Advanced Encryption Standard (AES),” FIPS Publication 197, U.S. Department of Commerce, Nov. 2001. [Online]. Available: https://csrc.nist.gov/csrc/media/publications/fips/197/final/documents/fips-197.pdf

[3] S. Frankel, R. Glenn, S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec," IETF RFC 3602, Sept. 2003. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc3602

[4] National Institute of Standards and Technology (NIST), "Cryptographic Algorithm Validation Program," [Online]. Available: https://csrc.nist.gov/projects/cryptographic-algorithm-validation-program/block-ciphers

[5] Assistant Digital Laboratory UI, "*Module 4: Testbench*," Universitas Indonesia, [Online]. Available: https://learn.digilabdte.com/books/digital-system-design/chapter/module-4-testbench

[6] Assistant Digital Laboratory UI, "*Module 6: Looping*," Universitas Indonesia, [Online]. Available: https://learn.digilabdte.com/books/digital-system-design/chapter/module-6-looping

[7] Assistant Digital Laboratory UI, "*Module 7: Procedure, Function, and Impure Function*," Universitas Indonesia, [Online]. Available: https://learn.digilabdte.com/books/digital-system-design/chapter/module-7-procedure-function-and-impure-function

[8] Assistant Digital Laboratory UI, "*Module 8 : Finite State Machine*," Universitas Indonesia, [Online]. Available: https://learn.digilabdte.com/books/digital-system-design/chapter/module-8-finite-state-machine

[9] Assistant Digital Laboratory UI, "*Module 7: Module 8 : Finite State Machine*," Universitas Indonesia, [Online]. Available: https://learn.digilabdte.com/books/digital-system-design/chapter/module-9-microprogramming

Terima Kasih