AES ENCRYPTOR DECRYPTOR GENERATOR

FINAL PROJECT PERANCANGAN
SISTEM DIGITAL

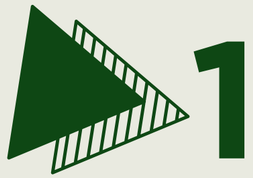
Group 25

AGENDA

- Meet The Team
- Intoduction
- Implementation
- Testing and Analysis
- Conclusion

MEET THE TEAM

- NAHL SYAREZA RAHIDRA
  2206830340
- XAVIER DANISWARA
  2206030230
- M. SESARAFLI ALJAGRA
  2206828071
- REICHAN ADHIGUNO
  2106703273
- SAMUEL TANAKA
  2206059710



Introduction

BACKGROUND

Proyek ini mengembangkan aplikasi berbasis VHDL yang mampu memproses teks melalui proses enkripsi dan dekripsi. Sistem memanfaatkan algoritma AES (Advanced Encryption Standard) karena memiliki tingkat keamanan yang tinggi dan efisiensi yang baik. Mode operasi aplikasi dikendalikan menggunakan opcode, sehingga sistem dapat menentukan apakah data harus dienkripsi atau didekripsi. Selain itu, alur kerja aplikasi diatur menggunakan Finite State Machine (FSM) untuk memastikan proses berjalan secara terstruktur, terkontrol, dan mudah dianalisis.

PROJECT DESCRIPTION

Sistem keamanan digital ini menggunakan AES-128 yang diimplementasikan dengan VHDL. Aplikasi dirancang secara modular dan mampu menjalankan dua fungsi utama, yaitu enkripsi dan dekripsi, dalam satu entitas terpadu. Selain itu, sistem memanfaatkan opcode sebagai pengendali mode, sehingga pemilihan proses enkripsi atau dekripsi dapat dilakukan secara dinamis tanpa perlu mengubah konfigurasi perangkat keras.

Arsitektur dikendalikan oleh Finite State Machine (FSM) untuk mengatur urutan proses kriptografi seperti:

- S-Box
- ShiftRows
- MixColumns
- Key Expansion
- AddRoundKey

Sistem ini dirancang menggunakan pendekatan digital design yang lengkap, mencakup behavioral style, structural modeling, penggunaan loop, serta penerapan functions dan procedures. Dengan pendekatan tersebut, sistem AES yang dihasilkan menjadi aman, efisien, serta memiliki struktur kode yang terorganisir dan mudah dianalisis.

OBJECTIVE

Perancangan Sistem AES-128 dengan VHDL

Merancang dan mengimplementasikan sistem enkripsi dan dekripsi menggunakan algoritma AES-128 dengan bahasa pemrograman VHDL.

Pengendalian Mode Operasi Menggunakan Opcode

Menggunakan opcode dalam pengendalian operasi untuk membedakan mode enkripsi dan dekripsi.

Penerapan FSM dalam Alur Proses

Menerapkan Finite State Machine (FSM) untuk memastikan setiap tahapan AES berjalan secara terstruktur dan berurutan

Perancangan Arsitektur Modular AES

Membangun arsitektur modular yang meliputi S-Box, Inverse S-Box, AddRoundKey, ShiftRows, MixColumns, dan Key Expansion.

Integrasi Konsep-Konsep Perancangan Sistem Digital

Membangun arsitektur modular yang meliputi S-Box, Inverse S-Box, AddRoundKey, ShiftRows, MixColumns, dan Key ExpansionMengintegrasikan modul perancangan sistem digital, seperti behavioral style, structural modeling, loop constructs, functions, procedures, impure functions, dan microprogramming.

.

ROLES AND RESPONSIBILITIES

PERSON	ROLES AND RESPONSIBILITES
Nahl Syareza Rahidra	Menulis laporan Chapter 3 Menulis kode main untuk encryption dan decryption Menyatukan berbagai modul yang ada menjadi satu kesatuan pada main Menulis kode Shift Rows Penggagas ide proyek
Xavier Daniswara	Menulis laporan Chapter 1 dan 2 Merancang Testbench untuk Enkripsi dan Dekripsi
M. Sesarafli Aljagra	Menulis laporan Chapter 1 dan 3 Membuat kode untuk Key Rounds
Reichan Adhiguno	Menulis laporan Chapter 1 dan 2 Membuat kode untuk Inverse Mix Columns
Samuel Tanaka	Menulis laporan Chapter 1 dan 4 Membuat kode bagian Mix Columns

2

Implementation

EQUIPMENT

ModelSim

digunakan untuk simulasi kode VHDL untuk menganalisis dan verifikasi output dari kode

digunakan sebagai code editor untuk menulis dan mengedit kode VHDL

Quartus® Prime Design Software

digunakan untuk proses sintesis, compile FPGA, dan analisis pada design VHDL

GitHub

digunakan sebagai platform untuk menyimpan kode hasil kolaborasi antar anggota kelompok

IMPLEMENTATION (COMPONENT)

SubBytes

- Direpresentasikan oleh s_box.
- Langkah substitusi non-linier yang menggunakan tabel S-Box untuk mengubah setiap byte state.
- Setiap byte diganti dengan nilai S-Box untuk enkripsi, dan dengan nilai inverse S-Box untuk dekripsi.
- Signal s_box_inp untuk original input, s_box_outp untuk hasil input yang sudah diubah, dan encrypt_or_decrypt untuk selector/pemilihan mode enkripsi atau dekripsi.

MixColumns

- Direpresentasikan dengan el_mixer untuk proses enkripsi dan rexim_le untuk proses dekripsi.
- Melakukan operasi matriks pada setiap kolom 4 byte dengan perkalian Galois Field (GF 2^8)
- Melakukan operasi matriks menggunakan matriks invers pada dekripsi.

ShiftRows

- Direpresentasikan oleh le_shift.
- Menggeser baris pada matriks state.
- Pergeseran ke arah kiri untuk enkripsi, dan kanan untuk dekripsi.
- le_shift_inp sebagai input asli, le_shift_outp sebagai output hasil shifting, le_shift_enable untuk memulai proses, le_shift_done sebagai penanda selesai, dan encrypt_or_decrypt untuk menentukan arah pergeseran (kiri untuk enkripsi, kanan untuk dekripsi).
- Baris pertama tidak digeser, baris kedua digeser sekali, baris ketiga digeser dua kali, dan baris keempat digeser tiga kali.

KeyExpansion

- Direpresentasikan oleh key_round.
- KeyExpansion menghasilkan round keys dari key awal.
- Untuk AES-128, terdapat total 11 round keys (round 0 sampai round 10).
- Signal key_round_sel untuk memilih round yang mana, key_round_inp untuk input, key_round_outp untuk output, dan key_round_kagi untuk round yang sudah di-expand.

IMPLEMENTATION FOR EVERY MODULE

- SubBytesBehavioral Style: Menggunakan process pada setiap architecture untuk mendeskripsikan perilaku logika secara terstruktur.
- Testbench: Digunakan untuk simulasi dan verifikasi agar output sesuai dengan hasil yang diharapkan.
- Structural Programming: Menerapkan struktur modular dengan memisahkan setiap tahap AES ke dalam entity tersendiri yang diinstansiasi pada modul utama.
- Looping: Memanfaatkan FOR loop untuk proses yang bersifat repetitif, seperti SubBytes dan Inverse SubBytes.
- Procedure, Function, dan Impure Function: Digunakan untuk mengenkapsulasi operasi berulang, termasuk operasi GF, substitusi byte, dan akses variabel bersama.
- Finite State Machine: Mengatur alur tahapan AES melalui transisi state yang dikendalikan oleh opcode.
- Microprogramming: Menggunakan opcode 4-bit untuk menentukan instruksi yang mengatur mode operasi dan tahapan yang harus dijalankan.

3

Testing And Analysis

TESTING (ENCRYPTION)

Input: 48656C6C6F20576F726C642121212121
Key: 2B7E151628AED2A6ABF7158809CF4F3C

Tahap 1: XOR

- Operasi XOR antara input dan Key Round 0.
- Output: 631B797A478E85C9D99B71A928EE6E1D

Tahap 2: SubBytes

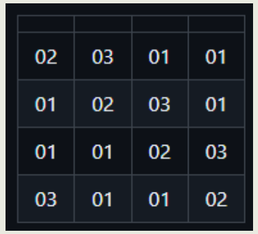
- Mengubah 16 byte menggunakan S-Box.
- Input: output XOR
- Output: FBAFB6DAA01997DD3514A3D334289FA4

Tahap 3: ShiftRows

- Shift kiri siklik sesuai baris (0,1,2,3 kali).
- Output: FB19A3A4A0149FDA3528B6DD34AF97D3

Tahap 4: MixColumns

- Perkalian matriks 4x4 (GF(2^8)).
- Output: C19348FF22E8E4DF79791660C600C0D9

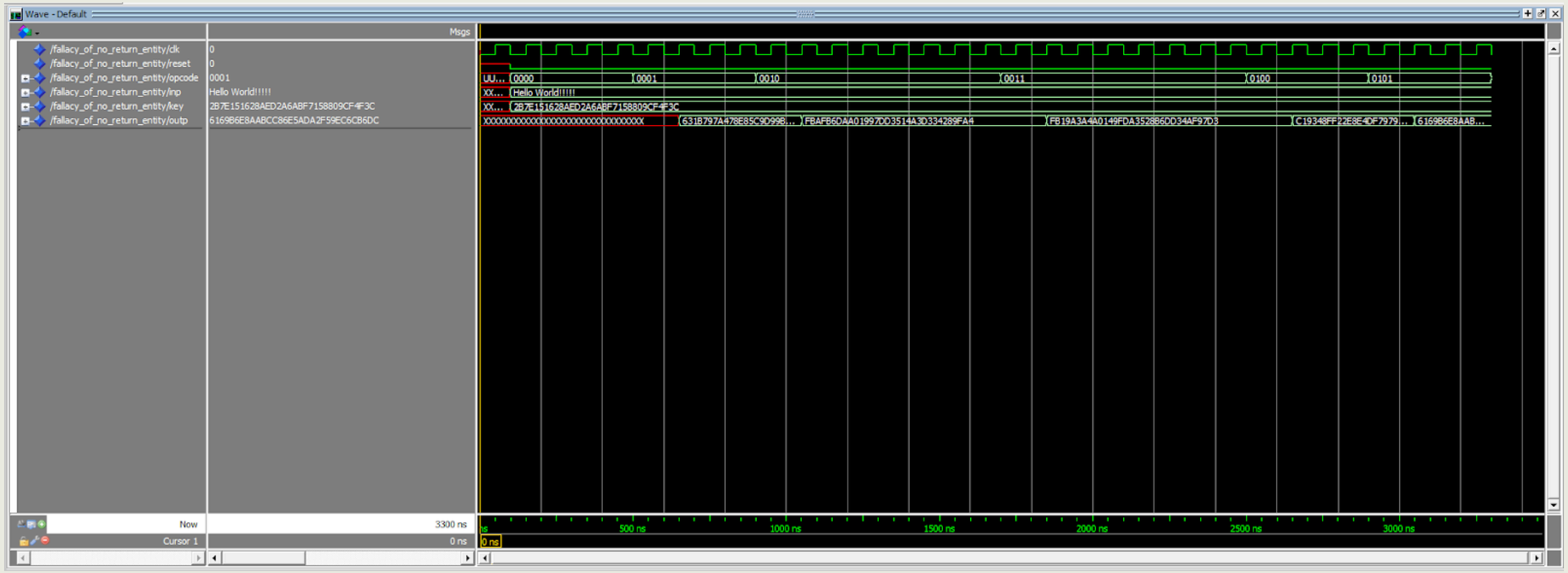


02	03	01	01
01	02	03	01
01	01	02	03
03	01	01	02

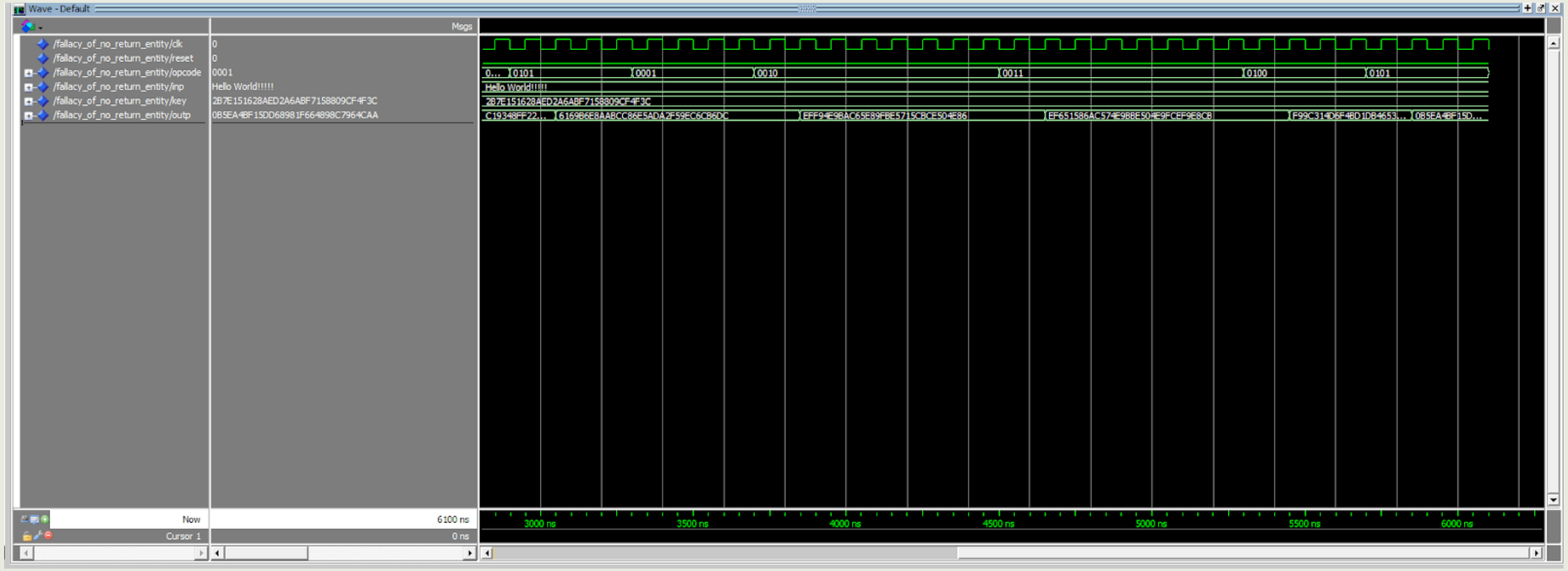
Tahap 5: Add Round Key

- XOR dengan Key Round 1 (hasil key expansion).
- Output Ronde 1: 6169B6E8AABCC86E5ADA2F59EC6CB6DC

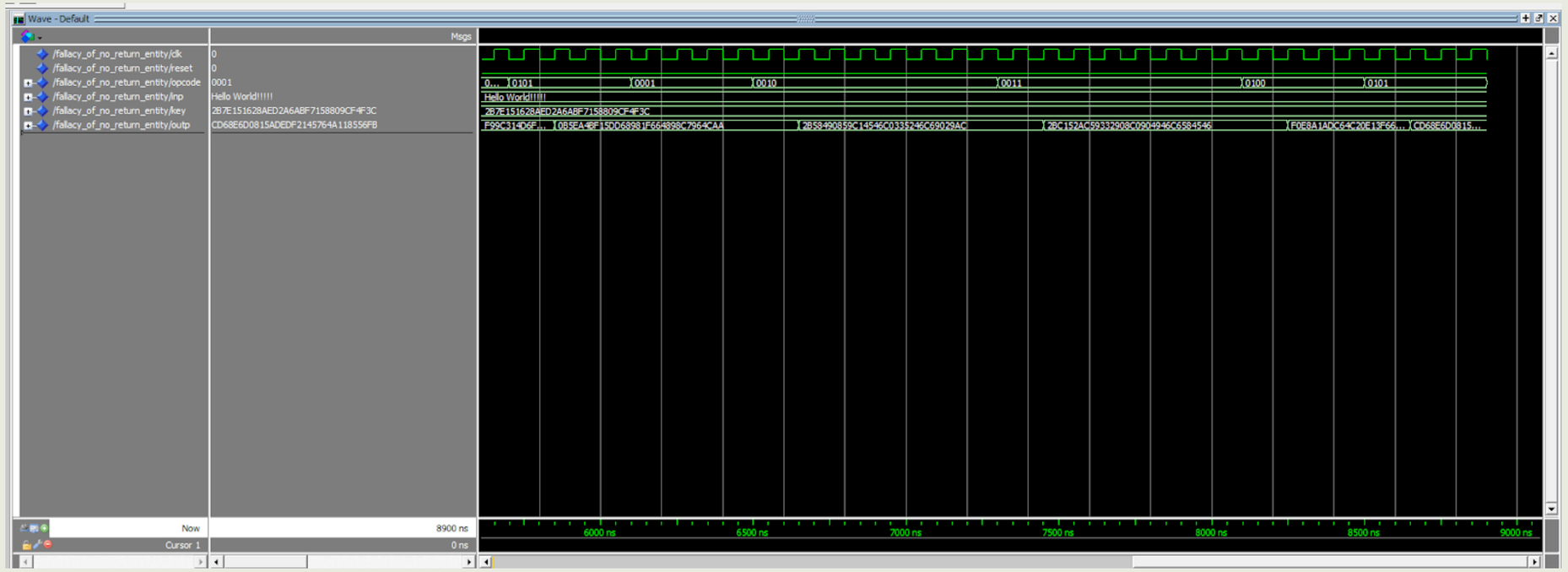
RESULT (ENCRYPTION)



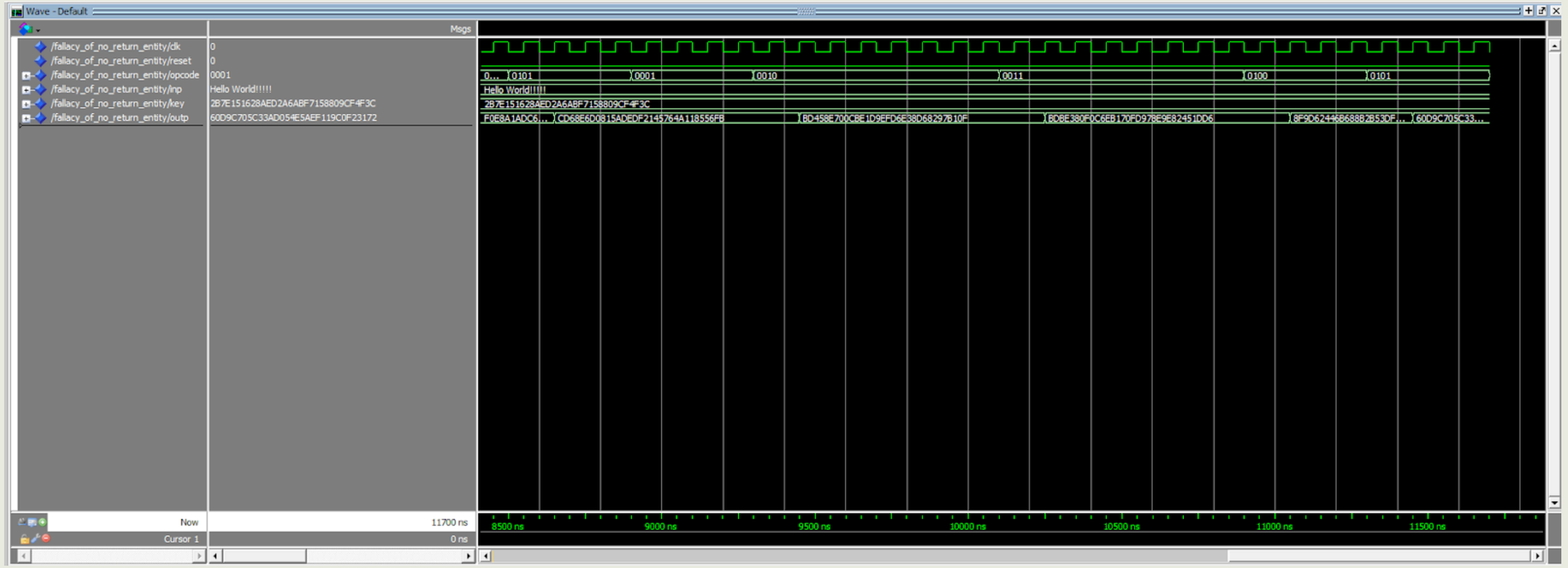
RESULT (ENCRYPTION)



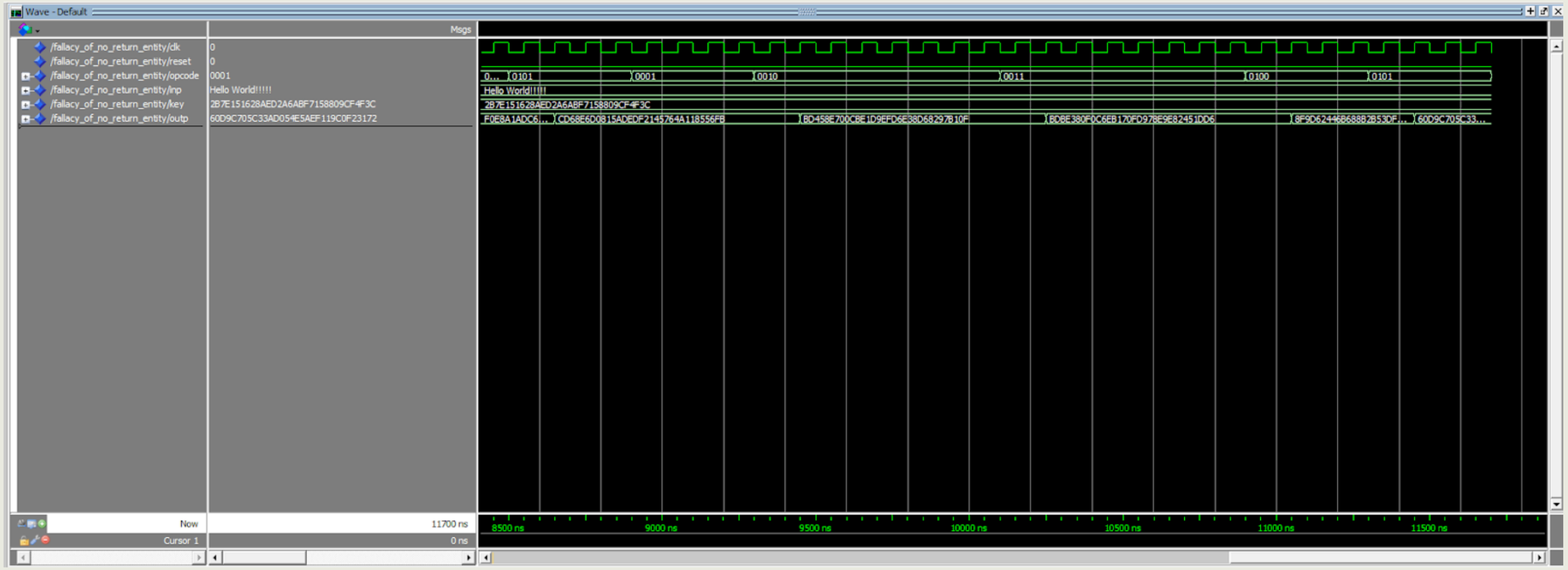
RESULT (ENCRYPTION)



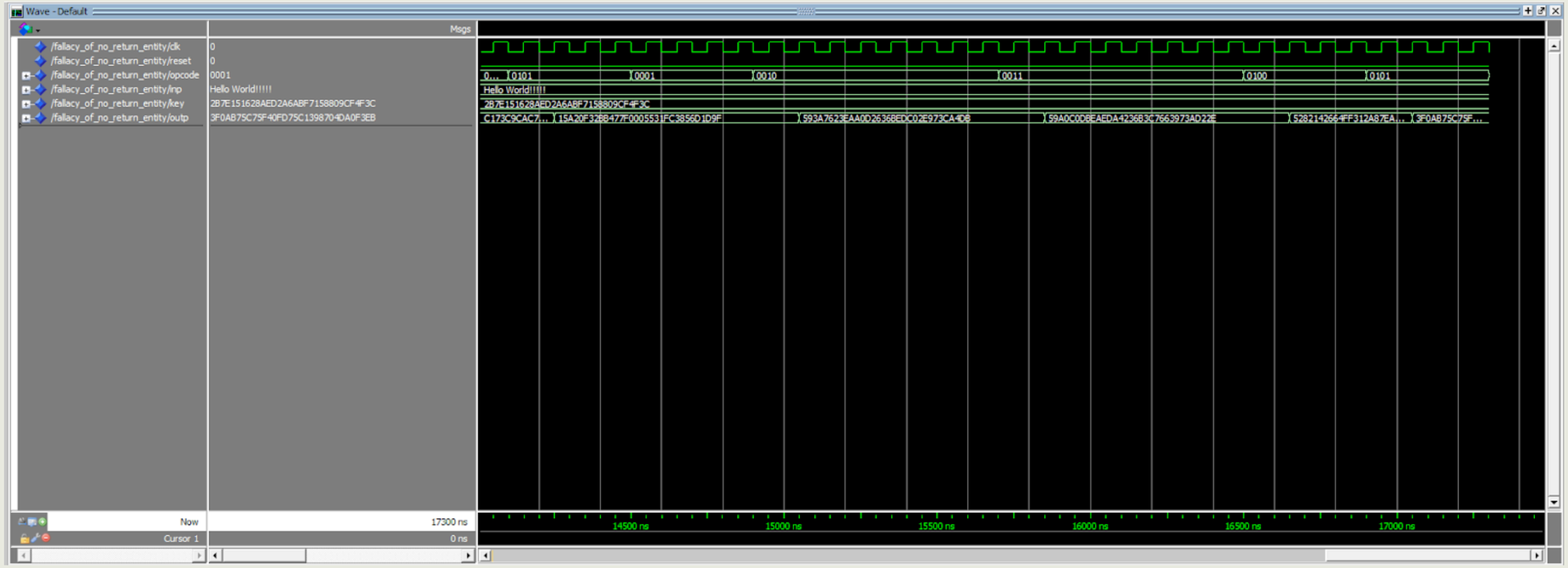
RESULT (ENCRYPTION)



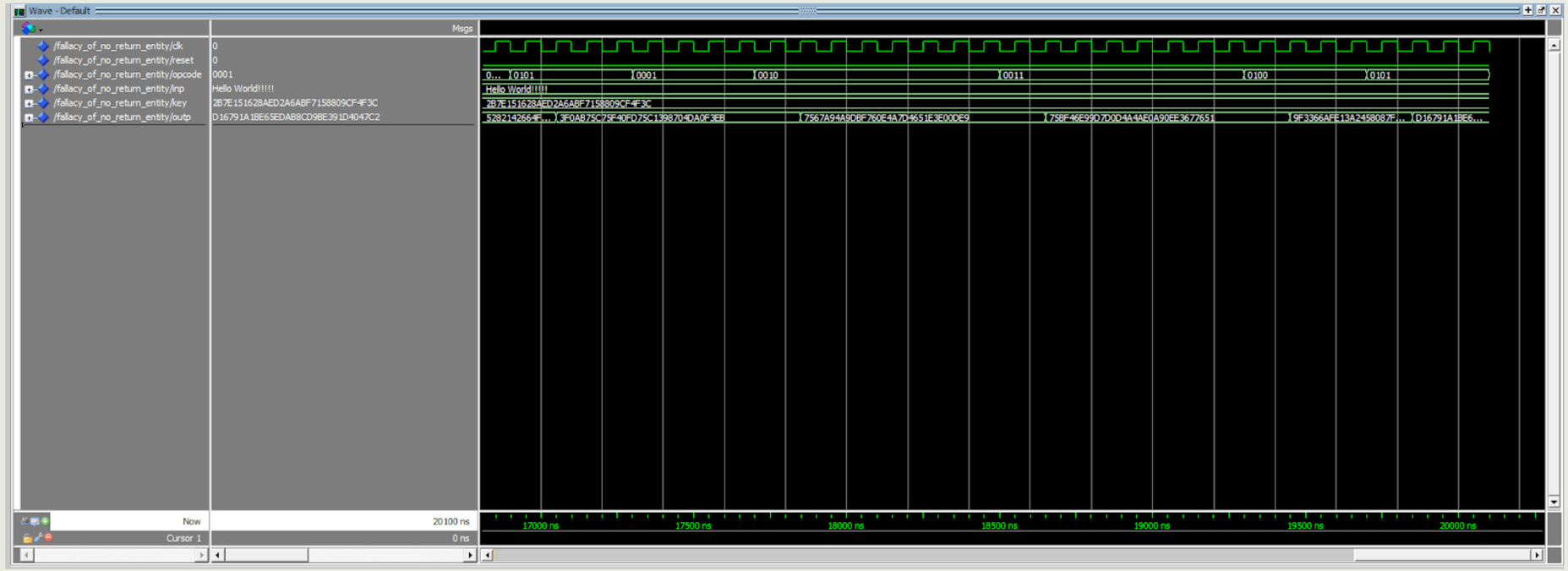
RESULT (ENCRYPTION)



RESULT (ENCRYPTION)



RESULT (ENCRYPTION)



RESULT (ENCRYPTION)

RESULT (ENCRYPTION)

RESULT (ENCRYPTION)

RESULT (ENCRYPTION)

```
Input: 48656C6C6F20576F726C642121212121
Key: 2B7E151628AED2A6ABF7158809CF4F3C

Output: 052BC3FB4A020CEDCBAB86FC0C06D404
Key: 2B7E151628AED2A6ABF7158809CF4F3C

Output: 052BC3FB4A020CEDCBAB86FC0C06D404
Key: 2B7E151628AED2A6ABF7158809CF4F3C

Output: 052BC3FB4A020CEDCBAB86FC0C06D404
Key: 2B7E151628AED2A6ABF7158809CF4F3C
```

Hasil dari output itu sendiri merupakan hasil akhir dari masing-masing ronde.

TESTING (DECRYPTION)

Input: 052BC3FB4A020CEDCBAB86FC0C06D404
Key: 2B7E151628AED2A6ABF7158809CF4F3C (sama seperti enkripsi)

Tahap 1: XOR

- Output: D53F3A5383EC29642A948A34BA65D8A2

Tahap 2: InvShiftRows

- Shift kanan sesuai baris.
- Output: D5658A64833FD8342AEC3AA2BA942953

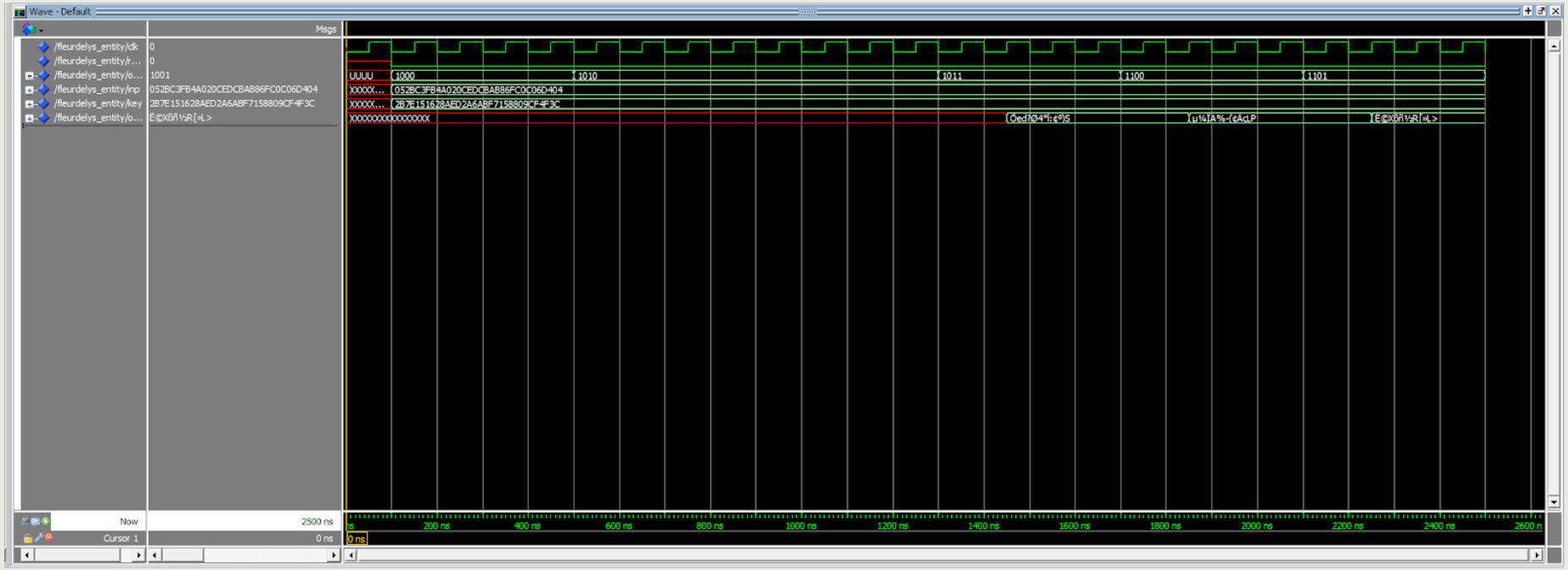
Tahap 3: InvSubBytes

- Menggunakan Inverse S-Box.
- Output (hasil akhir Ronde 9): B5BCCF8C41252D289583A21AC0E74C50

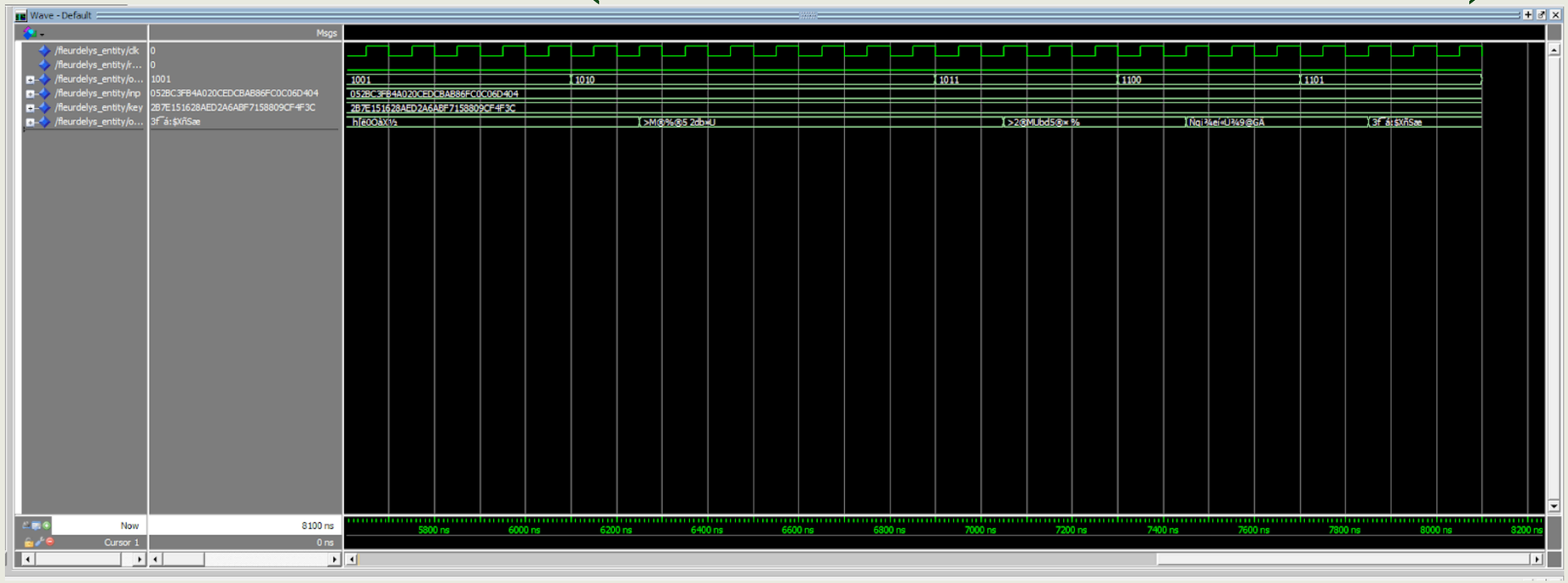
Tahap 4: Add Round Key (Round 9)

- XOR dengan Key Round 9.
- Output: 19CBA97F58DFF109BD528B5B97BB4C3E

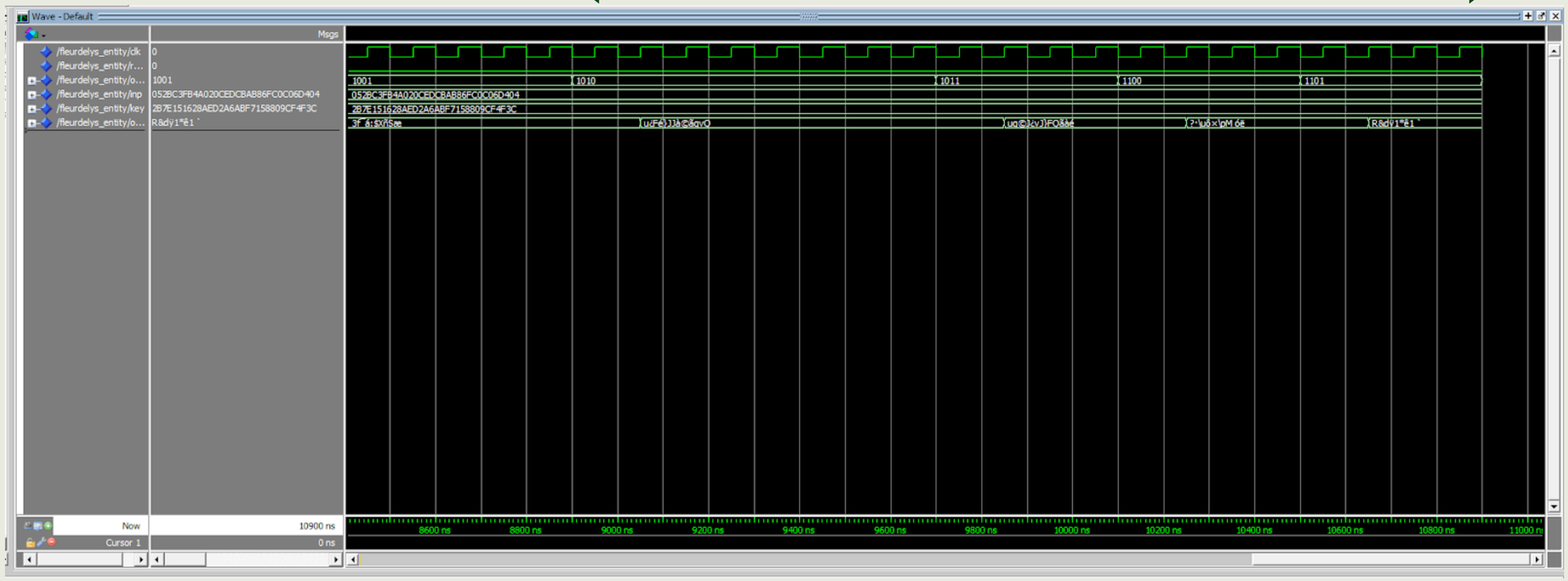
RESULT (DECRYPTION)



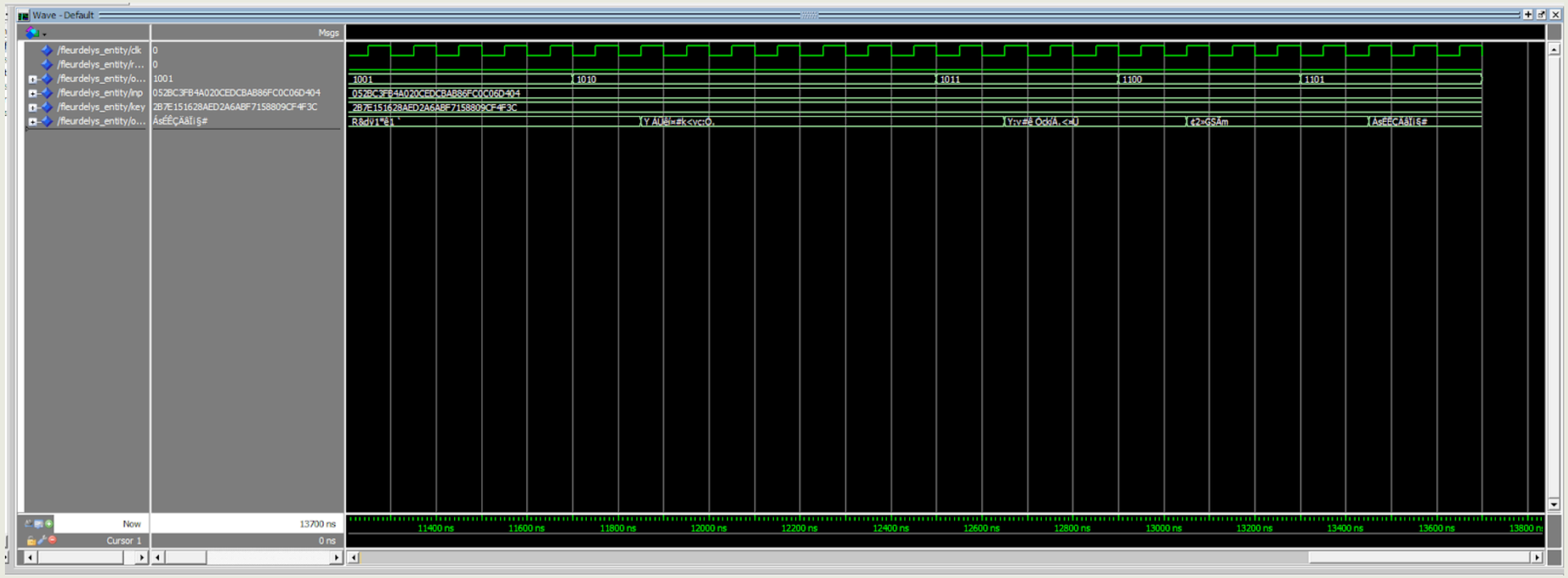
RESULT (DECRYPTION)



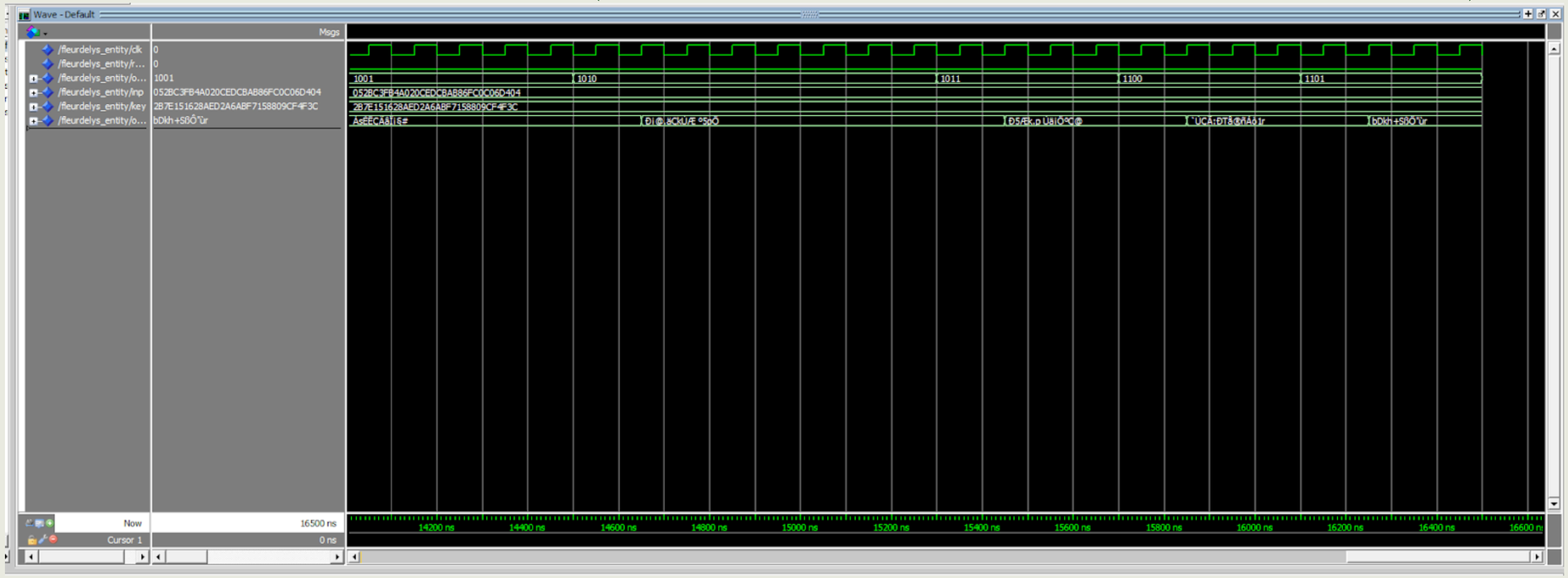
RESULT (DECRYPTION)



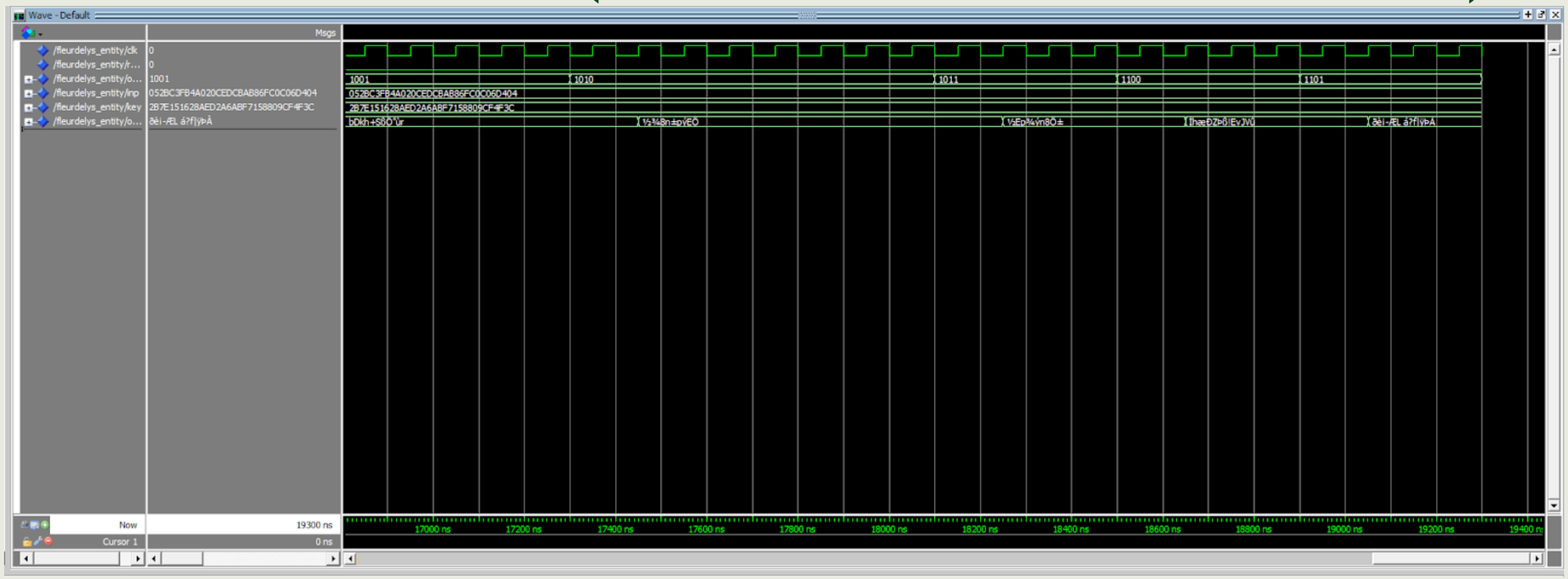
RESULT (DECRYPTION)



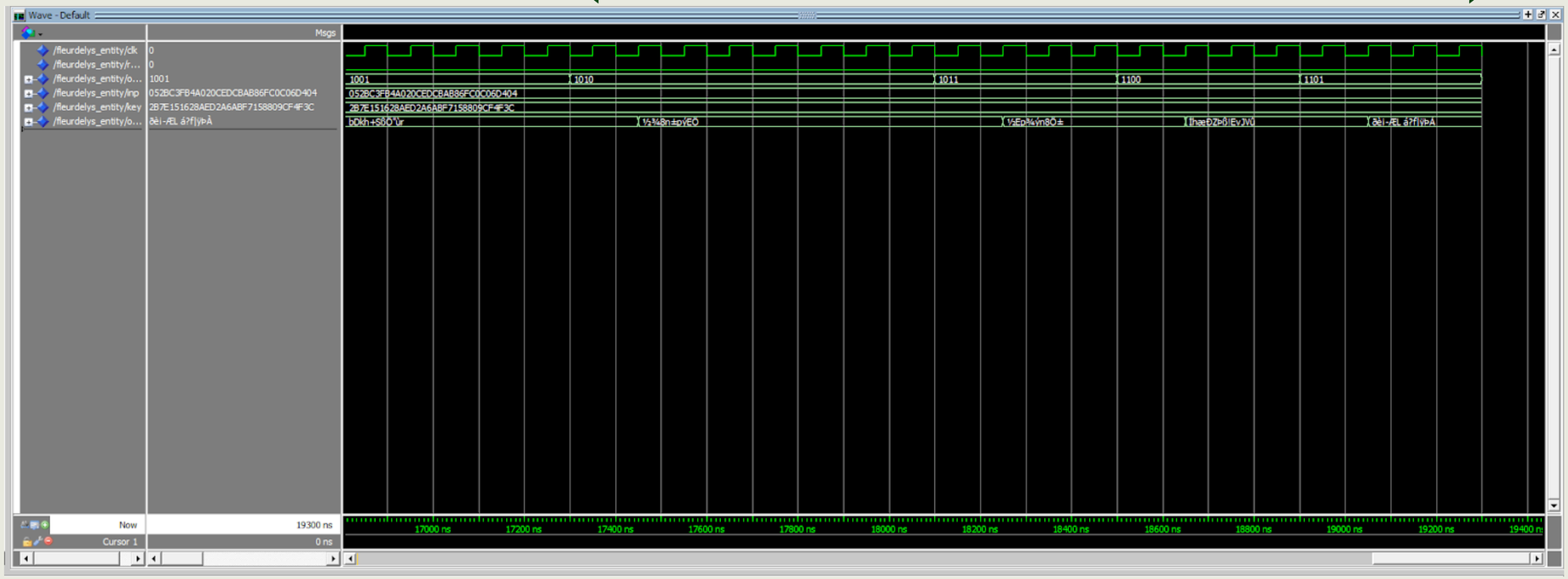
RESULT (DECRYPTION)



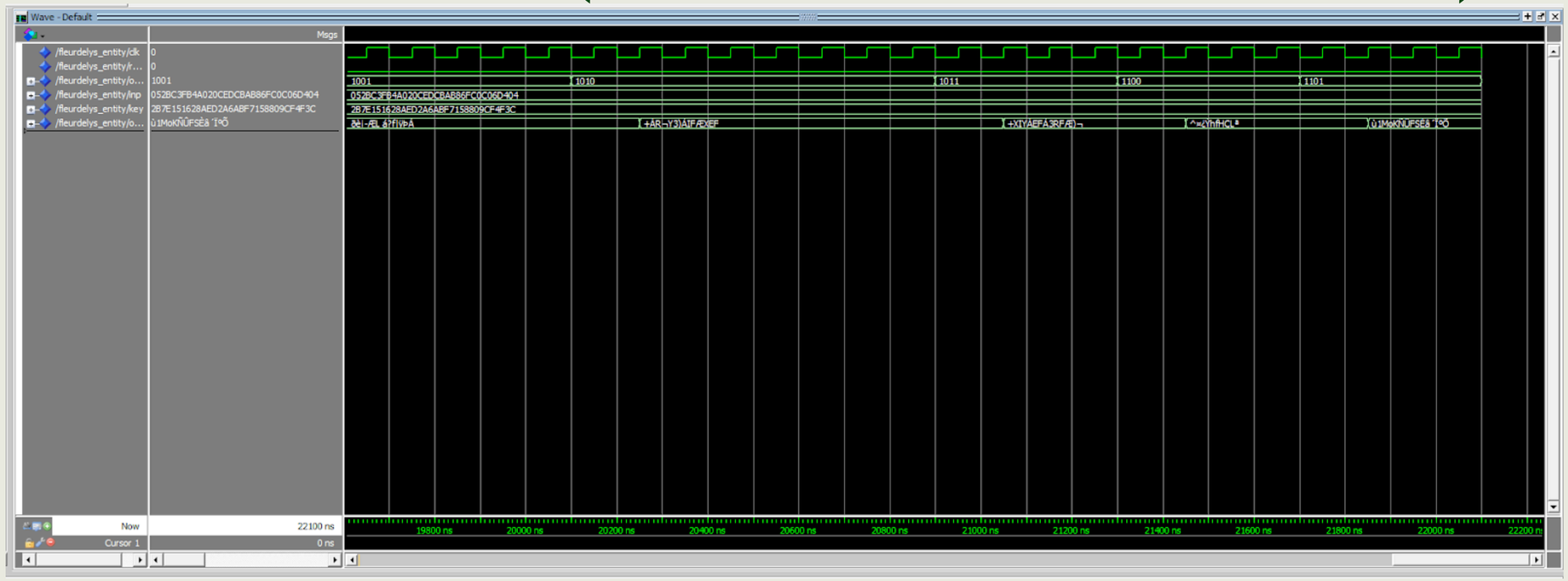
RESULT (DECRYPTION)



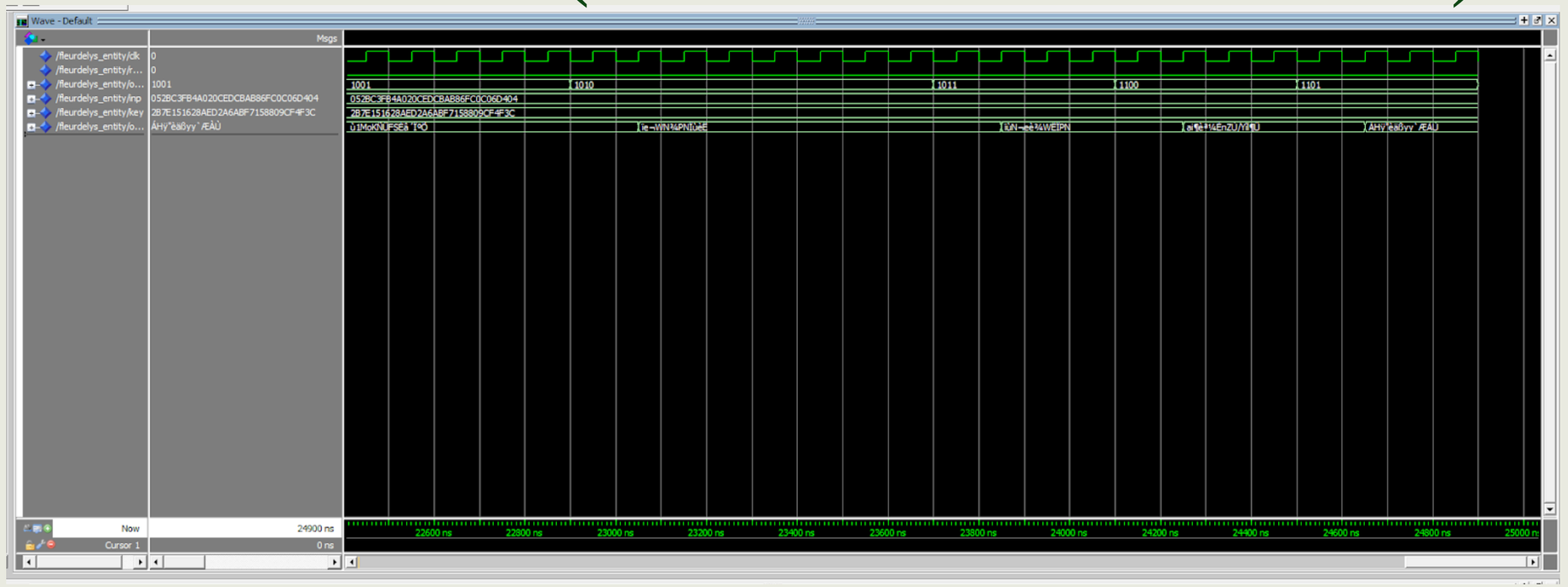
RESULT (DECRYPTION)



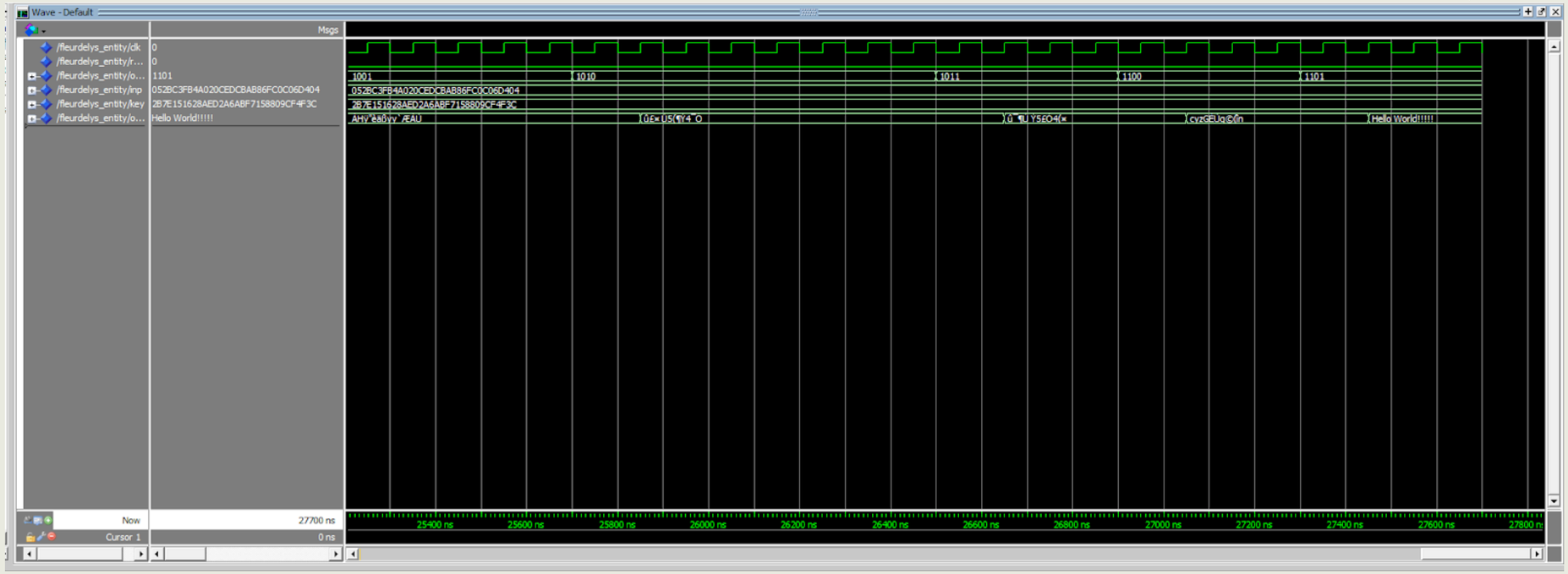
RESULT (DECRYPTION)



RESULT (DECRYPTION)



RESULT (DECRYPTION)



RESULT (DECRYPTION)

```
Input: 052BC3FB4A020CEDCBAB86FC0C06D404
Key: 2B7E151628AED2A6ABF7158809CF4F3C

Output: 48656C6C6F20576F726C642121212121
Key: 2B7E151628AED2A6ABF7158809CF4F3C
```

Untuk bagian dekripsi, output bukanlah hasil output dari ronde sebelumnya, namun merupakan data yang di XOR dengan key pada ronde tersebut yang akan menghasilkan output akhir per ronde nya.

ANALYSIS

ANALISIS SISTEM

- Analisis difokuskan pada cara kerja internal kode AES dalam VHDL.
- Sistem menambahkan tahap khusus (INIT & FINAL) untuk memastikan alur enkripsi/dekripsi berjalan benar.
- Penggunaan Nested FSM membuat eksekusi tiap modul lebih terstruktur.

TAHAPAN INIT & FINAL

- INIT
  - Digunakan di awal enkripsi/dekripsi.
  - Ronde pertama memiliki perlakuan berbeda (XOR dulu pada enkripsi, key transformasi pada dekripsi).
- FINAL
  - Menentukan output akhir karena port output hanya bisa mengambil hasil dari tahap sebelumnya.

ANALYSIS

MODUL & MODE OPERASI

- Modul AES dibuat dengan dua mode dalam satu file: encrypt & decrypt.
- Mempermudah pengendalian alur.
- MixColumns dipisah menjadi dua file (Encrypt & Decrypt) karena perbedaan logika Inverse MixColumns.

NESTED FSM & SINKRONISASI

- FSM utama memanggil modul-modul yang memiliki FSM internal.
- Menggunakan sinyal Done untuk memberi tahu bahwa proses modul selesai.
- Menjamin alur tidak tumpang tindih antar-FSM.

ENABLE & SOLUSI

- Child FSM membutuhkan port Enable agar tidak berjalan di waktu yang salah.
- Masalah awal: child FSM tidak reset sempurna jika Enable dimatikan terlalu cepat.
- Solusi: Mematikan Enable pada state IDLE, memastikan modul siap digunakan pada ronde berikutnya.

4

Conclusion

CONCLUSION

1

Sistem AES-128 Encryptor–Decryptor berhasil diimplementasikan dengan VHDL dan dikendalikan oleh opcode (0000: enkripsi, 1000: dekripsi).

2

Proses berjalan terstruktur melalui Finite State Machine (FSM) yang mengatur seluruh tahapan enkripsi dan dekripsi.

3

Proyek mengintegrasikan seluruh modul Perancangan Sistem Digital: Behavioral style, Structural modeling, Looping, Procedure, Function, Impure function, FSM, Microprogramming, serta Testbench untuk verifikasi.

4

Implementasi ini membuktikan penerapan efektif konsep desain digital dalam membangun sistem kriptografi yang aman, modular, dan fungsional.

Thank you.

Group 25