FUNDAMENTAL OF DIGITAL SYSTEM FINAL PROJECT REPORT
DEPARTMENT OF ELECTRICAL ENGINEERING
UNIVERSITAS INDONESIA

SEQUENTIAL COMBINATIONAL LOCK USING HASH FUNCTION

GROUP 15

ABYAN AMMAR ZAKI	**2406421005**
JOSHUA RICHARDO R	**2406361496**
MUH. HASHIF JADE	**2406396786**
TORIQ FATHONI DEZI	**2406487115**

Abstrak

Proyek kami dibuat untuk merancang dan mengimplementasikan sistem enkripsi password yang menerima masukan berupa angka dengan panjang 4 – 6 karakter, lalu menghasilkan nilai hash menggunakan fungsi SHA-3 dan membandingkannya dengan password user yang sudah terenkripsi SHA-3 yang sudah tersimpan pada modul secara hardcode. Sistem enkripsi dengan hashing ini diimplementasikan dalam bentuk bahasa hardware VHDL. Proyek kami hanya menggunakan software VHDL beserta alat pembantunya (modelsim, quartus/vivado) untuk menjalankan sistem. Kami memilih proyek ini karena enkripsi password secara hashing pada level hardware merupakan salah satu metode terbaik dan lebih aman dibandingkan dengan enkripsi biasa secara software untuk mengamankan perangkat.

Depok, December 07, 2025

Kelompok 15

TABLE OF CONTENTS

CHAPTER 1

PENDAHULUAN

1.1 Latar Belakang

Dengan berkembangnya kemajuan teknologi AI, keamanan perangkat menjadi lebih rentan untuk dibobol karena AI dapat melakukan analisis pola dan menghitung banyak kemungkinan yang dapat terjadi dalam waktu yang singkat sehingga keamanan perangkat konvensional menjadi lebih rentan untuk dibobol. Terdapat banyak cara untuk meningkatkan keamanan perangkat, salah satunya adalah dengan mengkonversi password dengan teknik hashing Sha-3 untuk menyimpan password.

Teknik hashing SHA-3 dapat meningkatkan keamanan perangkat dengan cara mengonversi password dari bentuk asli (*plaintext*) menjadi besaran output berbentuk hash (representasi satu arah yang tidak dapat kembali ke bentuk awal dengan panjang yang tetap) sehingga sulit untuk dilacak kembali kecuali mengetahui bentuk password asli. Alasan utama kami menggunakan SHA-3 dengan algoritma keccak adalah

Oleh karena itu, proyek ini fokus pada penerapan teknik hashing SHA-3 pada virtual lock (kunci digital) untuk memverifikasi akses dengan mencocokkan hasil konversi input password dengan hash yang telah ditetapkan sebelumnya. Selain dari mengimplementasikan apa yang sudah kami pelajari di semester ini pada mata kuliah Perancangan Sistem Digital, proyek kami ini juga memiliki tujuan meningkatkan keamanan sistem. Selain dari virtual lock yang kami buat pada proyek ini, teknik yang kami gunakan juga dapat berfungsi pada sistem embedded, akses pintu elektronik, atau aplikasi autentikasi dasar.

1.2 Deskripsi Proyek

Proyek ini dibuat untuk membangun sebuah virtual lock yang melakukan verifikasi password menggunakan teknik hashing SHA3-256. Password asli sebagai pengautentikasi tidak disimpan dalam bentuk plaintext, namun disimpan dengan bentuk hash SHA-3.

Ketika user (pengguna) memasukkan password, sistem akan melakukan hashing terhadap input tersebut dan membandingkannya dengan hash yang tersimpan. Jika hasilnya match, maka kunci virtual akan terbuka, jika tidak, maka akses ditolak. Pendekatan ini

memberikan keamanan yang lebih baik dibandingkan penyimpanan password secara langsung.

1.3 Deskripsi Proyek

1. Dapat merancang enkripsi password dengan hashing SHA-3
2. Merancang fungsi hashing SHA-3
3. Mengimplementasikan modul-modul yang telah dipelajari pada proyek ini
4. Menguji sistem dan menganalisanya

1.4 Peran dan Tanggung Jawab

Peran dan tanggung jawab kelompok

Roles	Responsibilities	Person
TesBench (tb_lock_system)	Membuat Testbench untuk Lock Controller beserta SHA3, dan membuat PPT	Joshua
Lock Controller (lock_controller)	Membuat modul Lock Function dan membuat Laporan	Abyan
SHA3-256 (SHA3_256_Core)	Membuat fungsi SHA 3 dan Offset, FSM, dan membuat Laporan	Hashif
SHA3-256 (SHA3_256_Core)	Membuat Entity beserta Type dan Constanta untuk modul fungsi SHA 3, dan membuat PPT	Toriq

Table 1. Tanggung Jawab

CHAPTER 2

IMPLEMENTASI

2.1 Peralatan

- Visual Code
- Modelsim
- Quartus
- Vivado
- GitHub

2.2 Implementasi

Proyek kami dibagi menjadi 3 bagian utama, dengan dua modul sebagai fungsi lock *(lock_controller)* sebagai top-level entity *(sha3_256_core)* dan fungsi hashing dengan menggunakan algoritma (keccak) dan modul testbench untuk menjalankan program.

Pada modul fungsi hashing, kami menggunakan algoritma keccak standar dengan matriks state internal 5x5 dan 5 state utama (S_IDLE, S_ABSORB, S_PERMUTE, S_SQUEEZE, dan S_DONE) yang diimplementasikan dengan Finite-State Machine (FSM). Tahap awal (S_IDLE) program menunggu perintah, berikutnya pada tahap absorbsi (S_ABSORB), input password digabungkan ke dalam memori internal menggunakan gerbang logika XOR, pada tahap permutasi (S_PERMUTE) password input yang sudah diabsorbsi akan di acak sebanyak 24 kali dan dilakukan permutasi dengan logika XOR dan rotasi bit (RHO_OFFSETS), setelahnya akan dihasilkan sebuah data dengan panjang 256 bit, output dari password yang telah diinput akan terbagi menjadi empat bagian yang dimana satu bagiannya berisikan dengan 64 bit dari baris pertama matriks. Ketika program telah sampai pada state done (S_DONE) password telah terenkripsi.

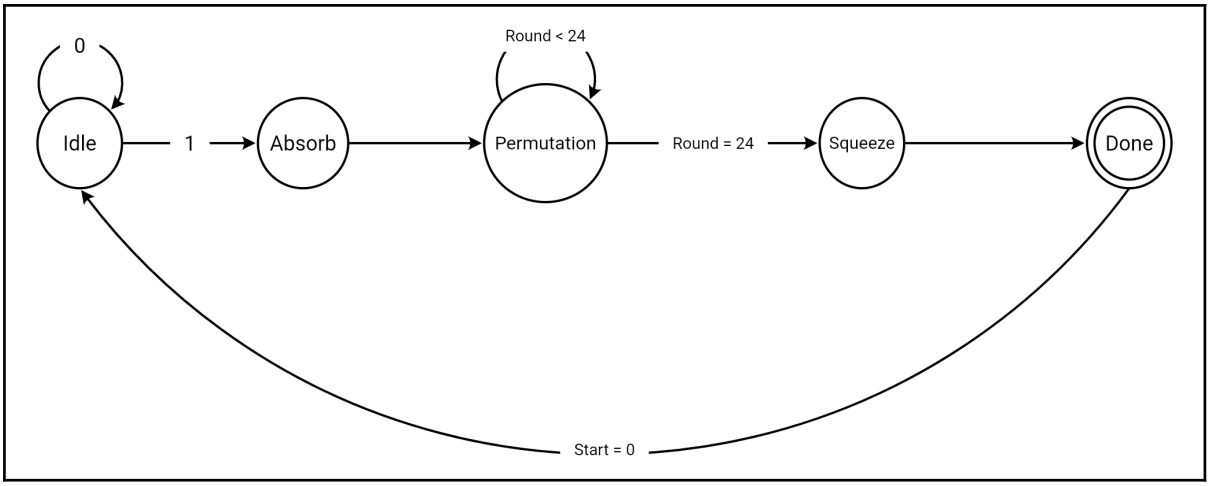


Diagram FSM fungsi hashing

Pada modul fungsi lock, akan dilakukan perbandingan antara hasil dari enkripsi yang telah dilakukan pada modul fungsi hashing dan jika hasilnya cocok, maka digital key akan terbuka (pada *waveform* akan bernilai 1/naik) dan sebaliknya jika hasilnya tidak cocok maka key tidak akan terbuka.

CHAPTER 3

PENGUJIAN DAN ANALISA

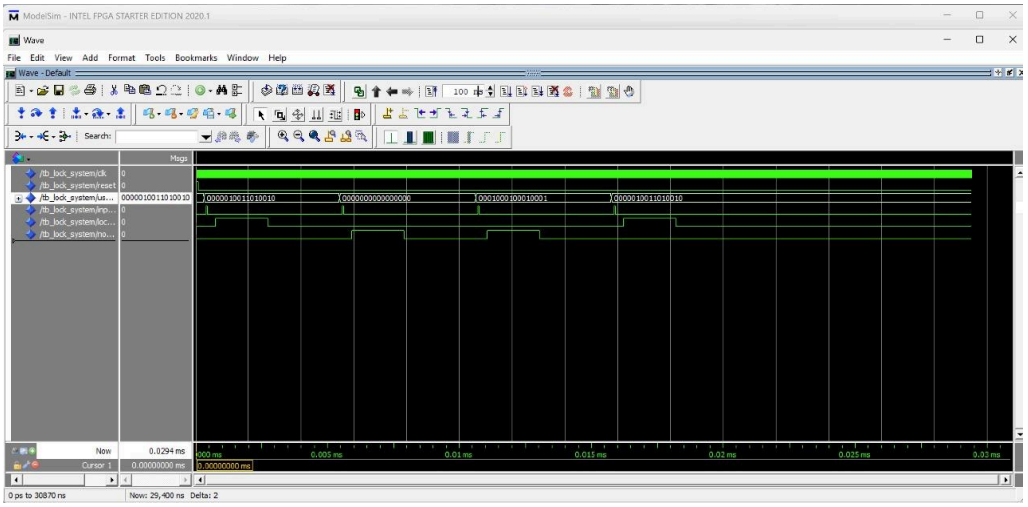
3.1 Pengujian

Testing dilakukan dengan cara memberikan password terenkripsi SHA-3 yang dapat membuka lock secara hardcode pada fungsi lock controller, password yang sudah terenkripsi ini didapat dari X atau SHA-3 converter online. Input disimpan secara hardcode juga pada testbench, untuk mengetest kebenaran sistem, diberikan dua input, satu input yang benar (sesuai dengan password yang dapat membuka lock) dan satu input yang salah

3.2 Hasil Pengujian

Pada hasil pengujian yang dilakukan melalui testbench, sistem berhasil menghasilkan output hash dari input password yang dimasukkan, kemudian membandingkannya dengan hash referensi yang telah ditentukan sebelumnya dalam modul lock controller. Pada percobaan dengan input password yang sesuai (input benar), sinyal output unlock berubah menjadi HIGH (logika ‘1’), yang menunjukkan bahwa virtual lock berhasil terbuka. Selain itu, perubahan state internal pada modul juga terlihat konsisten mengikuti alur proses hashing sehingga kondisi terbukanya lock dapat diamati secara jelas pada waveform hasil simulasi.

Sementara itu, ketika diberikan input password yang salah, hasil hashing tidak sesuai dengan hash referensi sehingga output unlock tetap LOW (logika ‘0’) dan virtual lock tidak terbuka. Hasil ini dapat terlihat pada waveform bahwa perbedaan input password menghasilkan kondisi output yang berbeda sesuai dengan proses verifikasi. Selain itu, ketika terjadi input yang salah berulang-ulang, sistem tetap mempertahankan kondisi locked tanpa menimbulkan perubahan status pada bagian output, sehingga mendukung mekanisme keamanan yang diharapkan.



Testing Result

Pengujian menunjukkan bahwa fungsi hashing, proses pembandingan, dan mekanisme penguncian bekerja sesuai rancangan, di mana virtual lock hanya terbuka jika hash input sesuai dengan hash referensi. Hasil simulasi juga memperlihatkan proses hashing yang deterministik dan stabil, tanpa perubahan nilai setelah eksekusi selesai. Dengan demikian, sistem telah memenuhi tujuan implementasi dan dapat menjadi dasar pengembangan fitur autentikasi digital selanjutnya.

3.3 Analisa

Berdasarkan hasil pengujian, sistem telah mampu melakukan proses hashing SHA3-256 terhadap input password sekaligus melakukan verifikasi sesuai nilai hash yang telah ditentukan. Hal ini menunjukkan bahwa fungsi Keccak yang diimplementasikan pada modul dapat menghasilkan output dengan panjang 256 bit yang konsisten serta dapat digunakan sebagai pembanding yang valid. Selain itu, karena password disimpan dalam bentuk hash dan bukan plaintext, sistem ini menjadi lebih aman dibandingkan metode penyimpanan biasa.

Saat password yang salah diberikan, hash yang dihasilkan tidak sesuai sehingga verifikasi gagal dan virtual lock tetap tertutup, menunjukkan mekanisme pembandingan telah berjalan dengan benar. Secara keseluruhan, implementasi SHA-3 pada virtual lock memberikan performa hashing yang stabil, verifikasi yang akurat, dan dapat dikembangkan lebih lanjut untuk sistem keamanan berbasis autentikasi digital.

CHAPTER 4

KESIMPULAN

Pada proyek ini, kami merancang sebuah virtual lock yang menggunakan teknik hashing SHA3-256 sebagai metode verifikasi password. Dengan algoritma Keccak, password plaintext diubah menjadi hash satu arah yang aman sehingga password asli tidak pernah disimpan ataupun terlihat selama proses kerja sistem. Pendekatan ini memberikan keamanan yang jauh lebih baik dibandingkan penyimpanan password secara langsung.

Proyek ini terdiri dari tiga bagian penting: proses hashing SHA-3, bagian pengendali lock yang melakukan pengecekan kecocokan hash, serta bagian testbench yang digunakan untuk melakukan pengujian keseluruhan sistem. Proses hashing melibatkan tahapan absorbsi, permutasi 24 ronde, dan squeeze hingga menghasilkan keluaran sepanjang 256 bit. Bagian lock kemudian membandingkan hasil hashing dari input dengan nilai hash referensi yang telah ditetapkan. Melalui testbench, dilakukan pengujian menggunakan input yang benar dan input yang salah untuk memastikan sistem berjalan dengan benar.

Hasil pengujian menunjukkan bahwa virtual lock hanya terbuka ketika hash input sesuai dengan hash yang sudah disimpan, dan tetap tertutup ketika input tidak valid. Secara keseluruhan, proyek ini memberikan pengalaman mendalam mengenai penerapan algoritma kriptografi pada sistem digital, perancangan FSM, serta bagaimana beberapa bagian sistem saling terintegrasi. Teknik yang digunakan juga dapat diterapkan pada sistem autentikasi dasar, perangkat embedded, maupun akses elektronik yang membutuhkan tingkat keamanan lebih tinggi.

REFERENSI

[1] National Institute of Standards and Technology, “SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions”, FIPS PUB 202, Aug. 2015. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf

[2] J. Daemen, “Introduction to SHA-3 and Keccak,” Crypto Summer School Lecture, 2015. [Online]. Available: https://summerschool-croatia.cs.ru.nl/2015/SHA3.pdf

[3] “SHA-256 and SHA-3 – GeeksforGeeks,” GeeksforGeeks, Jul. 2025. [Online]. Available: https://www.geeksforgeeks.org/computer-networks/sha-256-and-sha-3/

[4] F. Kurniawan, “Analisis dan Implementasi Algoritma SHA-1 dan SHA-3 pada Sistem Autentikasi,” *Jurnal Pembangunan Teknologi Informasi dan Ilmu Komputer (J-PTIIK)*, 2017. [Online]. Available: https://j-ptiik.ub.ac.id/index.php/j-ptiik/article/view/247

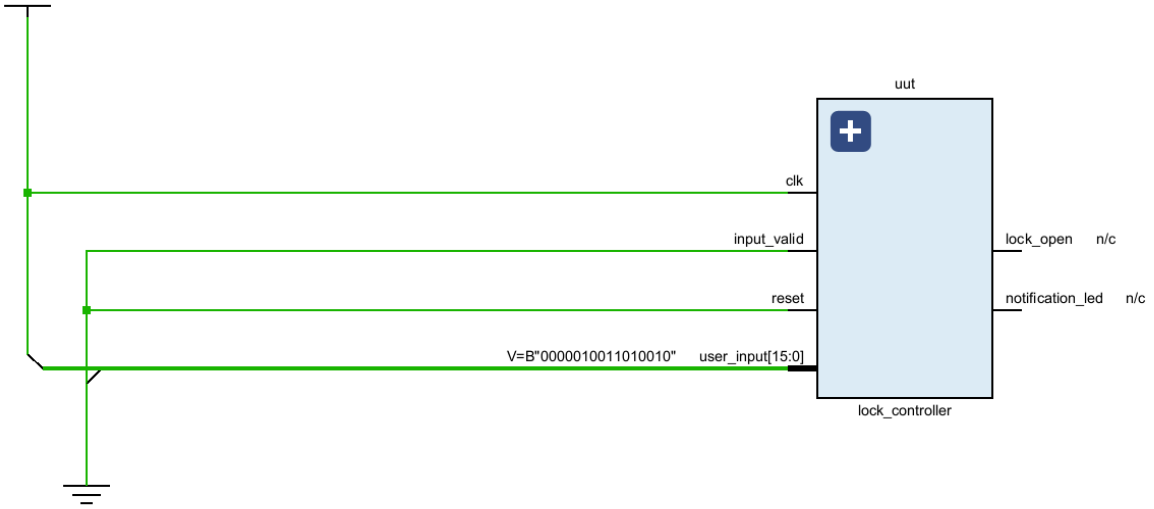
[5] “Hardware implementation of SHA-3 (Keccak) algorithm,” OpenCores, 2014. [Online]. Available: https://opencores.org/projects/sha-3

[6] B. Jungk, “Evaluation of Compact FPGA Implementations for All SHA-3 Finalists,” *Proceedings of the SHA-3 Candidate Conference*, 2012. [Online]. Available: https://csrc.nist.rip/groups/ST/hash/sha-3/Round3/March2012/documents/papers/JUNGK_paper.pdf

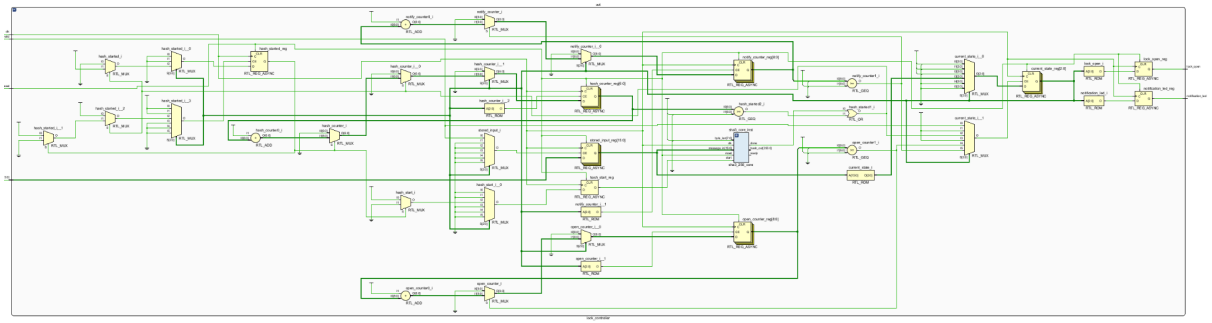
LAMPIRAN

Lampiran A: Project Schematic

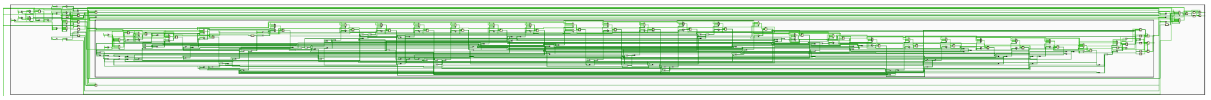
RTL view keseluruhan sistem terlihat seperti berikut:



Kemudian, jika kita buka fungsi lock_controllernya, menjadi seperti ini:



Kemudian jika kita buka fungsi SHA3-nya:



Lampiran B: Documentation