**RAVAN A 512-BIT CRYPTO ENGINE**

The project is fully based on cryptography that can be developed in an hardware as new architecture that can encrypt and decrypt a data using a 512 bit key.

This project can store the data as an encrypted data in memory and decrypt the data from the memory. The main goal of this project is to secure a data by using a hardware not by a software. The technologies used in this project are a sha 256 hashing algorithm that can secure data more efficiently.

INSTALATION :

Download the entire repository (ZIP method)

Best for: Getting all project files at once

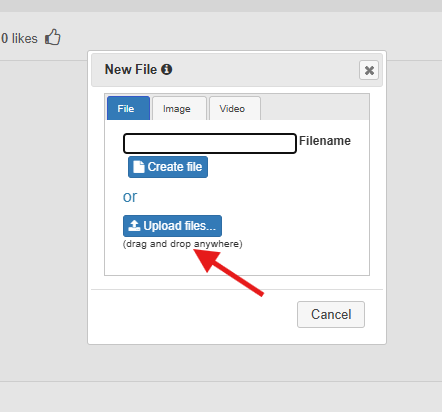
* Go to the GitHub repository page (e.g., https://github.com/user/repo-name)
* Click the green <> Code button
* Click Download ZIP
* Extract the downloaded ZIP file on your system

RECQUIRED TOOLS:

* Synopsys vcs 2023(or)
* Cadance Xcelium 23.09(or)
* EDA playground

USAGE:

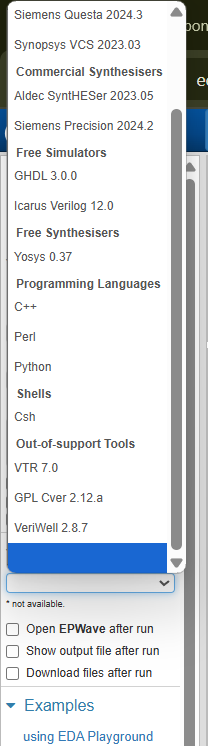
If you use eda playground you need to just import the files with import files button,



After that you just check that files are in right place,

After that just select the tool you need in the side bar,like Synopsys vcs 2023(or)

Cadance Xcelium 23.09,



Then save the file and just click run option to simulate the code,

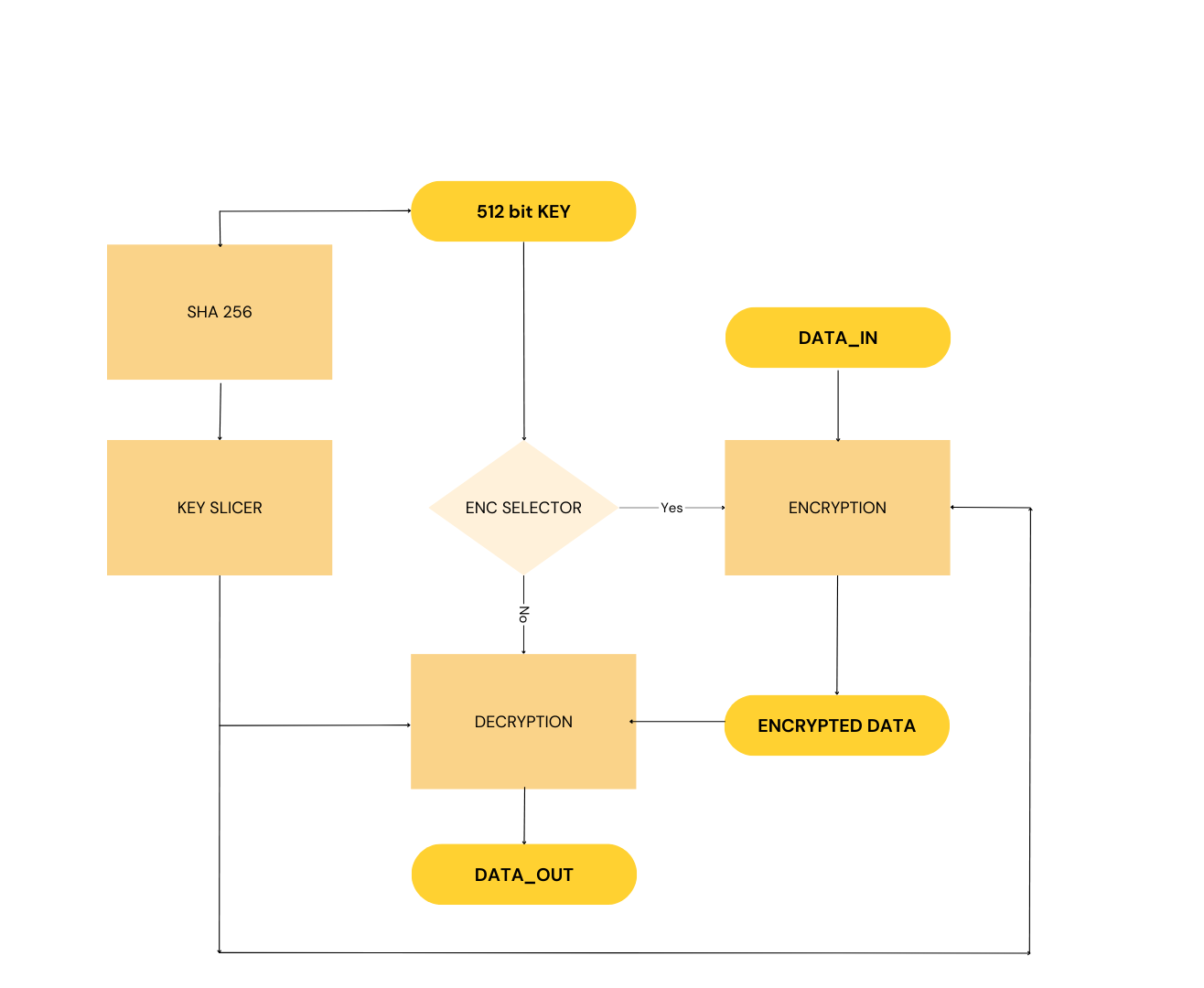
If you are using candance or synopsys you just need to create a new project and import all the files and then simulate it.

FEATURES:

* SHA 256
* Custom encryption / decryption
* AXI bus(needs to be implemented)

ARCHITECHTURE:

The RAVAN Encryption Engine is a custom 512-bit cryptographic system designed for secure communication, inspired by reversible logic and mythological strength. It features a modular architecture for encryption, decryption, and key management, with strong emphasis on unpredictability, dynamic slicing, and layered transformation.



1. RAVAN\_TOP

Main control unit that routes operations based on a control signal.

Manages encryption or decryption mode.

Connects to key slicer and SHA-256 module.

Integrates rounds, handles enable/reset signals.

2. RAVAN\_Encryption

Accepts 512-bit plain data and 512-bit main key.

Internally performs 21 reversible logic rounds:

XOR, AND, OR-based transformations

Bit permutations and shift-based obfuscation

Each round uses one 64-bit key slice.

3. RAVAN\_Decryption

Mirrors encryption but in reverse round order.

Uses the same key slices, applied in reverse sequence.

Ensures full data recovery without loss.

4. Key Slice Engine

Dynamically generates 8 x 64-bit keys from a single 512-bit master key.

Uses hashed indexing, selective bit slicing, and non-linear logic.

Prevents predictability and enhances security.

5. SHA-256 Module

Hashes either data or key, depending on security level required.

Adds a hashed authentication layer or supports hashed key derivation.

Data Flow

Input: 512-bit plain data + 512-bit main key

Key Engine: Slices the key into 8 parts

Encryption Module: Executes 21 rounds with 64-bit key slices

Output: 512-bit ciphertext

Decryption: Reverses the same logic using reversed key order

Reversible Logic Rationale

The RAVAN engine uses reversible logic blocks (like XOR/AND combinations) to:

Avoid information loss

Support backtracking in decryption

Optimize for low-power hardware implementations

RTL Modules Included

ravan\_encryption.v

ravan\_decryption.v

ravan\_key\_slicer.v

sha256.v

ravan\_top.v

ravan\_testbench.v

Future Enhancements

* Add AES-style substitution-permutation network
* Introduce counter-based randomized slicing
* Implement fault detection and tamper-proofing

Contact Information:

Linkedin:[https://www.linkedin.com/in/devaprasath-thangavelu-424011294](https://www.linkedin.com/public-profile/settings" \t "https://www.linkedin.com/in/devaprasath-thangavelu-424011294/edit/forms/contact-info/new/_blank)

Email:devaprasathece05@gmail.com