SCIENCE
PASSION
TECHNOLOGY

The Joy of Hardware Implementation using ChatGPT

Julian Sechshauser

July 4, 2025

Motivation

- Increasing importance of lightweight cryptography.
- Challenges in hardware implementations.
- Rapid evolution of Large Language Models (LLMs).

Motivation

- Increasing importance of lightweight cryptography.
- Challenges in hardware implementations.
- Rapid evolution of Large Language Models (LLMs).

Motivation

- Increasing importance of lightweight cryptography.
- Challenges in hardware implementations.
- Rapid evolution of Large Language Models (LLMs).

Main Question

How effective is ChatGPT in assisting hardware implementations?

ASCON

- Standardized by NIST in 2023.
- Provides authenticated encryption and hashing.
- Advantages: small area footprint, high security, energy efficiency.

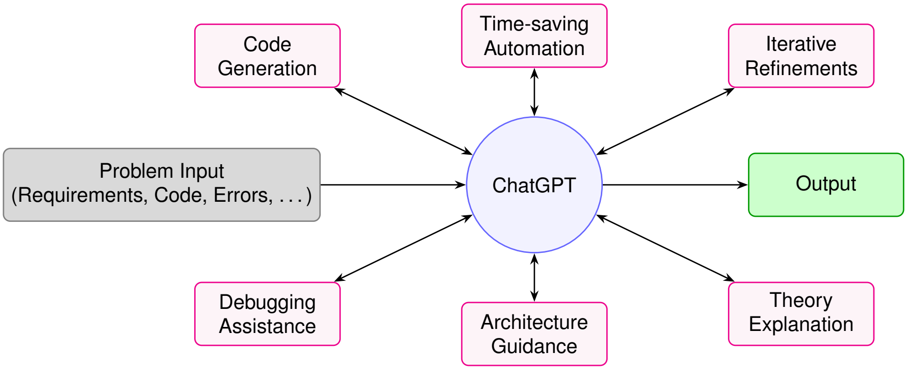
ASCON

- Standardized by NIST in 2023.
- Provides authenticated encryption and hashing.
- Advantages: small area footprint, high security, energy efficiency.

ASCON

- Standardized by NIST in 2023.
- Provides authenticated encryption and hashing.
- Advantages: small area footprint, high security, energy efficiency.

ChatGPT's Workflow



Starting Point

- No prior experience in cryptographic hardware design.
- Began with Python implementation of ASCON.
- Gradually translated logic into Verilog modules step-by-step.

Starting Point

- No prior experience in cryptographic hardware design.
- Began with Python implementation of ASCON.
- Gradually translated logic into Verilog modules step-by-step.

Starting Point

- No prior experience in cryptographic hardware design.
- Began with Python implementation of ASCON.
- Gradually translated logic into Verilog modules step-by-step.

First Steps with ChatGPT

- Asking ChatGPT if it was familiar with the Ascon.
- Prompted it to generate an exact Python implementation from the paper.

First Steps with ChatGPT

- Asking ChatGPT if it was familiar with the Ascon.
- Prompted it to generate an exact Python implementation from the paper.

ASCON Implementation Process

- Development of Verilog modules for hardware realization.
- Simulation using Icarus Verilog.
- Verifying modules with generated testbenches.

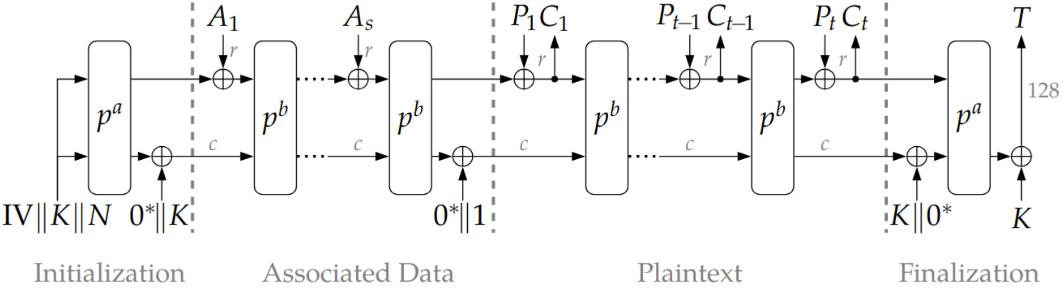
ASCON Implementation Process

- Development of Verilog modules for hardware realization.
- Simulation using Icarus Verilog.
- Verifying modules with generated testbenches.

ASCON Implementation Process

- Development of Verilog modules for hardware realization.
- Simulation using Icarus Verilog.
- Verifying modules with generated testbenches.

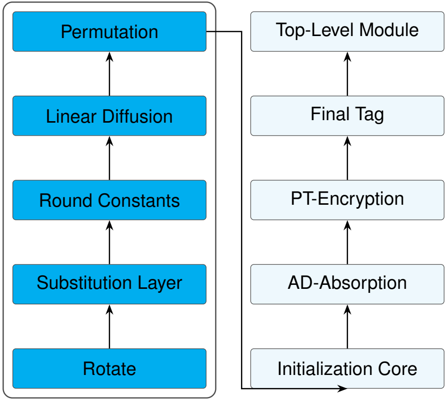
Encryption Pipeline



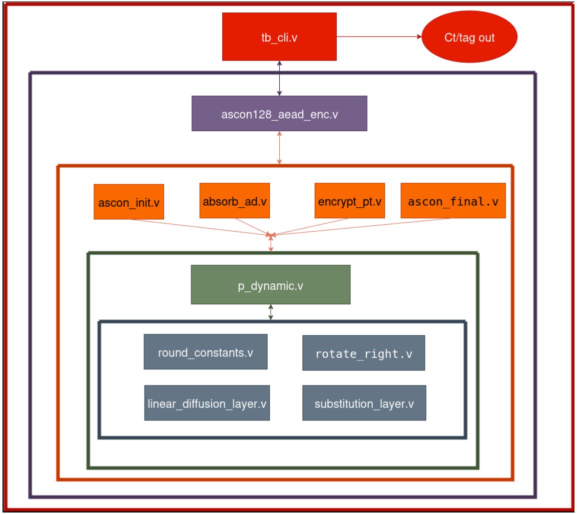
(a) Encryption $\mathcal{E}_{k,r,a,b}$

Source: https://csrc.nist.gov/CSRC/media/Projects/lightweight-cryptography/

Module Implementation



High-level View



Debugging Strategies

- Modular, step-by-step testing.
- Instrumented with reference-style debug output.
- Compared outputs with official C reference implementation.

Debugging Strategies

- Modular, step-by-step testing.
- Instrumented with reference-style debug output.
- Compared outputs with official C reference implementation.

Debugging Strategies

- Modular, step-by-step testing.
- Instrumented with reference-style debug output.
- Compared outputs with official C reference implementation.

Completed Implementations

- Fully working implementation of **ASCON AEAD** (encryption + tag).
- Separate, standalone **ASCON Hash function** implementation.
- Code volume produced:
  - 1 518 lines—ASCON128 AEAD encryptor (Verilog) (32 byte AD & PT 141 clk‘s)
  - 1 900 lines—testbenches (Verilog)
  - 560 lines—ASCON hash core (Verilog)
  - 200 lines—reference hashing script (Python)
  - 3878 lines—in total

Completed Implementations

- Fully working implementation of **ASCON AEAD** (encryption + tag).
- Separate, standalone **ASCON Hash function** implementation.
- Code volume produced:
  - 1 518 lines—ASCON128 AEAD encryptor (Verilog) (32 byte AD & PT 141 clk‘s)
  - 1 900 lines—testbenches (Verilog)
  - 560 lines—ASCON hash core (Verilog)
  - 200 lines—reference hashing script (Python)
  - 3878 lines—in total

Completed Implementations

- Fully working implementation of **ASCON AEAD** (encryption + tag).
- Separate, standalone **ASCON Hash function** implementation.
- Code volume produced:
  - 1 518 lines—ASCON128 AEAD encryptor (Verilog) (32 byte AD & PT 141 clk‘s)
  - 1 900 lines—testbenches (Verilog)
  - 560 lines—ASCON hash core (Verilog)
  - 200 lines—reference hashing script (Python)
  - 3878 lines—in total

Completed Implementations

- Fully working implementation of **ASCON AEAD** (encryption + tag).
- Separate, standalone **ASCON Hash function** implementation.
- Code volume produced:
  - 1 518 lines—ASCON128 AEAD encryptor (Verilog) (32 byte AD & PT 141 clk‘s)
  - 1 900 lines—testbenches (Verilog)
  - 560 lines—ASCON hash core (Verilog)
  - 200 lines—reference hashing script (Python)
  - 3878 lines—in total

Completed Implementations

- Fully working implementation of **ASCON AEAD** (encryption + tag).
- Separate, standalone **ASCON Hash function** implementation.
- Code volume produced:
  - 1 518 lines—ASCON128 AEAD encryptor (Verilog) (32 byte AD & PT 141 clk‘s)
  - 1 900 lines—testbenches (Verilog)
  - 560 lines—ASCON hash core (Verilog)
  - 200 lines—reference hashing script (Python)
  - 3878 lines—in total

Completed Implementations

- Fully working implementation of **ASCON AEAD** (encryption + tag).
- Separate, standalone **ASCON Hash function** implementation.
- Code volume produced:
  - 1 518 lines—ASCON128 AEAD encryptor (Verilog)
    (32 byte AD & PT 141 clk's)
  - 1 900 lines—testbenches (Verilog)
  - 560 lines—ASCON hash core (Verilog)
  - 200 lines—reference hashing script (Python)
  - 3878 lines—in total

Completed Implementations

- Fully working implementation of **ASCON AEAD** (encryption + tag).
- Separate, standalone **ASCON Hash function** implementation.
- Code volume produced:
  - 1 518 lines—ASCON128 AEAD encryptor (Verilog) (32 byte AD & PT 141 clk‘s)
  - 1 900 lines—testbenches (Verilog)
  - 560 lines—ASCON hash core (Verilog)
  - 200 lines—reference hashing script (Python)
  - 3878 lines—in total

Key Takeaways

- ChatGPT accelerated development and helped navigate initial stages quickly.
- Debugging and verification is not trivial and needs the right approach.
- It matters which model is being used.

Key Takeaways

- ChatGPT accelerated development and helped navigate initial stages quickly.
- Debugging and verification is not trivial and needs the right approach.
- It matters which model is being used.

Key Takeaways

- ChatGPT accelerated development and helped navigate initial stages quickly.
- Debugging and verification is not trivial and needs the right approach.
- It matters which model is being used.

Which Models I Used & Why

Model	Used When	Reason / Strength
GPT-4o	Beginning / setup	Fast – great for initial structuring
GPT-o3	Most debugging	Reliable; strong at reasoning
GPT-o3-pro	Final passes	Slower, but most accurate

Final Reflection

- I started this journey from zero hardware knowledge.
- Now I have working ASCON AEAD and hash implementations in Verilog.
- ChatGPT made hardware accessible but it still required a lot of work.
- Was it a joy?
- Would I use ChatGPT again? Yes—but correct prompt usage is very important.

Final Reflection

- I started this journey from zero hardware knowledge.
- Now I have working ASCON AEAD and hash implementations in Verilog.
- ChatGPT made hardware accessible but it still required a lot of work.
- Was it a joy?
- Would I use ChatGPT again? Yes—but correct prompt usage is very important.

Final Reflection

- I started this journey from zero hardware knowledge.
- Now I have working ASCON AEAD and hash implementations in Verilog.
- ChatGPT made hardware accessible but it still required a lot of work.
- Was it a joy?
- Would I use ChatGPT again? Yes—but correct prompt usage is very important.

Final Reflection

- I started this journey from zero hardware knowledge.
- Now I have working ASCON AEAD and hash implementations in Verilog.
- ChatGPT made hardware accessible but it still required a lot of work.
- Was it a joy?
- Would I use ChatGPT again? Yes—but correct prompt usage is very important.

Final Reflection

- I started this journey from zero hardware knowledge.
- Now I have working ASCON AEAD and hash implementations in Verilog.
- ChatGPT made hardware accessible but it still required a lot of work.
- Was it a joy?
- Would I use ChatGPT again? Yes—but correct prompt usage is very important.

Final Reflection

- I started this journey from zero hardware knowledge.
- Now I have working ASCON AEAD and hash implementations in Verilog.
- ChatGPT made hardware accessible but it still required a lot of work.
- Was it a joy?
- Would I use ChatGPT again? Yes—but correct prompt usage is very important.