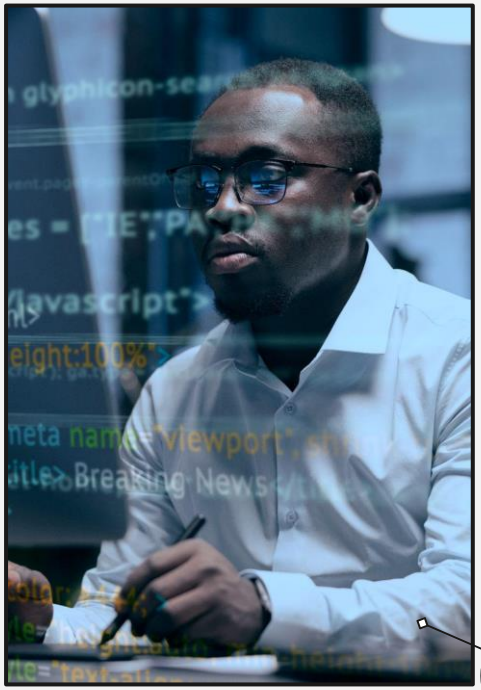
Enhancing Data Security: Analysis of Symmetric and Asymmetric Cryptography Algorithms

Presented By Brad Bolluyt, Steven Do, Charles Tran, and Elizabeth Woo

Table of contents

Executive summary

This benchmarking study evaluates the cryptographic performance of the Nexys A7-100T FPGA (Field-Programmable Gate Array) when implementing three widely used encryption algorithms: AES (Advanced Encryption Standard), RSA (Rivest-Shamir-Adleman), and DES (Data Encryption Standard). The Nexys A7-100T FPGA offers a parallel processing architecture, making it well-suited for cryptographic applications.



01

Symmetric vs. Asymmetric Cryptography

The Differences

Number of keys:

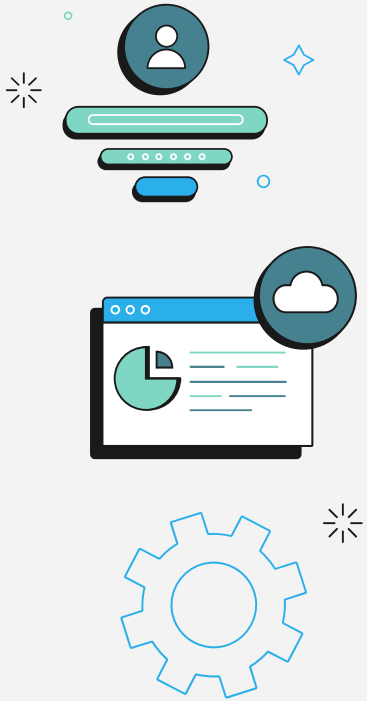
- Symmetric - Uses a single shared secret key for both encryption/decryption
- Asymmetric - Involves a pair of mathematically related keys: a public key for encryption and a private key for decryption

Complexity:

- Symmetric - Generally faster and more computationally efficient
- Asymmetric - Involves more complex mathematical operations, making it slower

Uses:

- Symmetric - Commonly used for encrypting large amounts of data (e.g., files, messages) due to its efficiency.
- Asymmetric - Often used for secure key exchange, digital signatures, and establishing secure communication channels.



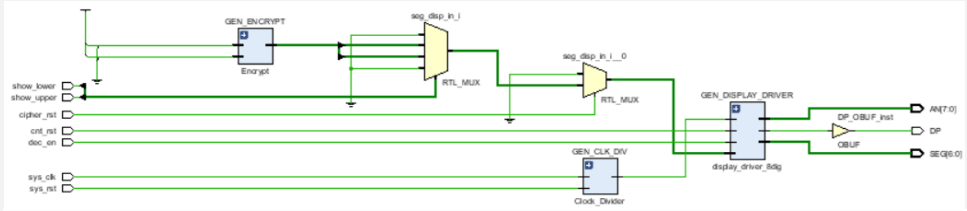
02

DES (Data Encryption Standard)

What is DES?

The Data Encryption Standard (DES) is a symmetric key encryption algorithm designed for securing electronic data. Developed by IBM in the 1970s and later standardized by the U.S. National Institute of Standards and Technology (NIST), DES operates on fixed-size blocks of data, typically 64 bits. Its fundamental principle involves the use of a single secret key for both encryption and decryption. DES performs a series of substitution and permutation operations, organized into multiple rounds, to transform plaintext into ciphertext and vice versa. Despite its historical significance, DES is considered a legacy algorithm due to its susceptibility to brute-force attacks with modern computing power. It has been largely replaced by more secure encryption standards like AES.

DES Synthesis on Board

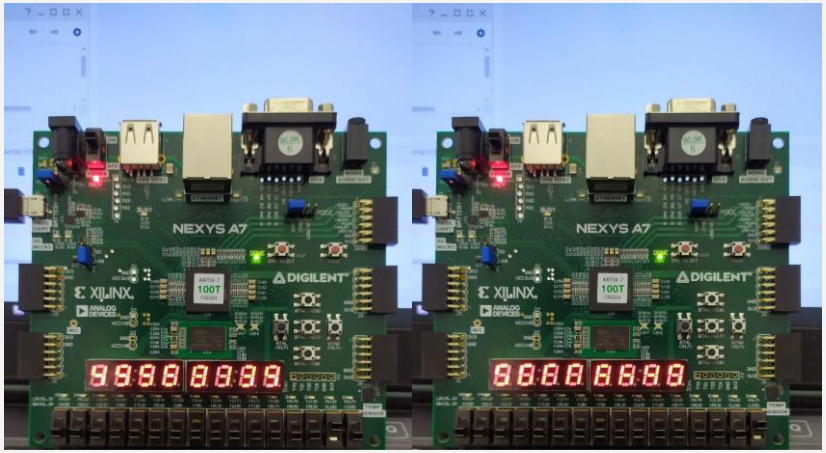


Resource	Utilization	Available	Utilization %
LUT	14	63400	0.02
FF	5	126800	0.00
IO	23	210	10.95

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement	Source Clock
Path 5	∞	3	2	7	show_lower	SEG[6]	6.831	5.225	1.606	∞	input port clock
Path 6	∞	3	2	7	show_lower	SEG[1]	6.815	5.209	1.606	∞	input port clock
Path 7	∞	3	2	7	show_lower	SEG[5]	6.809	5.204	1.606	∞	input port clock
Path 8	∞	3	2	7	show_lower	SEG[3]	6.804	5.198	1.606	∞	input port clock
Path 9	∞	3	2	8	dec_en	AN[2]	6.804	5.198	1.606	∞	input port clock
Path 10	∞	3	2	7	show_lower	SEG[0]	6.791	5.185	1.606	∞	input port clock

- This DES implementation uses a low amount of resources on the board
- The highest delay through this program is 6.831ns. This means the maximum frequency is around 1/6.831ns ≈ 146.39 MHz

DES Implementation on Board



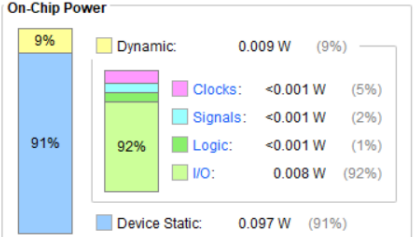
Upper 32 bits of Ciphertext

Lower 32 bits of Ciphertext

- For the implementation the amount of resources remained the same as the synthesis
- The amount of power used was shown to be 0.106W,

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.106 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.5°C**
Thermal Margin:	59.5°C (12.9 W)
Effective ϑJA:	4.6°C/W
Power supplied to off-chip devices:	0 W



03

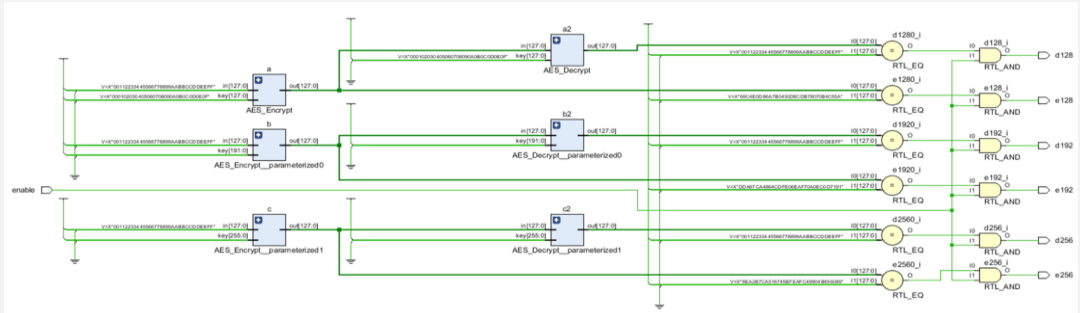
AES (Advanced Encryption Standard)

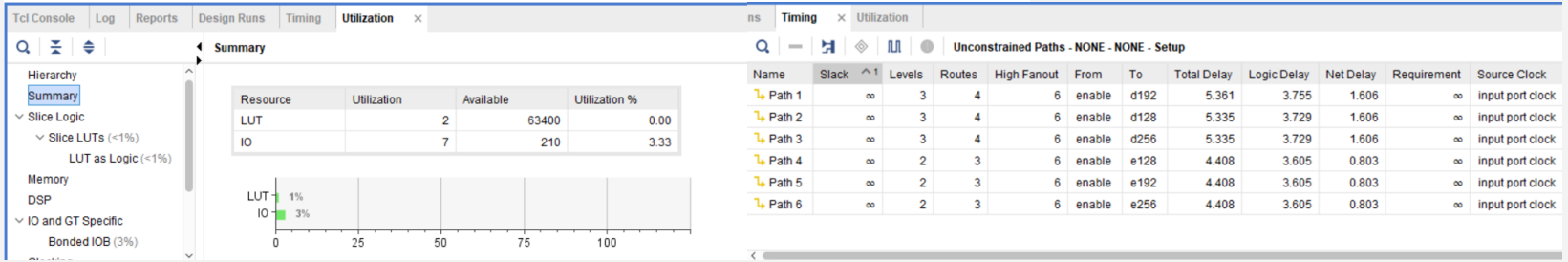
What is AES?

AES, or Advanced Encryption Standard, is a widely used symmetric encryption algorithm designed to secure sensitive information. It was established as a standard by the National Institute of Standards and Technology (NIST) in 2001. AES operates on fixed-size blocks of data and supports key sizes of 128, 192, or 256 bits.

The algorithm employs a series of substitution, permutation, and mixing operations, organized into multiple rounds, to transform plaintext data into ciphertext. AES is known for its strength and efficiency, making it a cornerstone in securing data for a variety of applications, including securing communications over the internet, encrypting files, and protecting sensitive information in various computing environments. The algorithm's security is attributed to its resistance against known cryptographic attacks and its widespread adoption in various industries.

AES Synthesis on Board

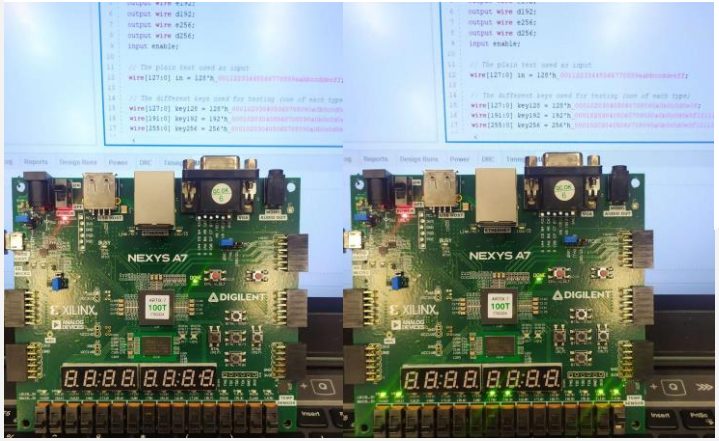




- This AES implementation uses a very low amount of LUTs and IOs. It is very lightweight in comparison to the previous DES implementation.
- The highest delay through this program is 5.361ns. This means the maximum frequency is around $1/5.361\text{ns} \approx 186.53$ MHz

AES Implementation on Board

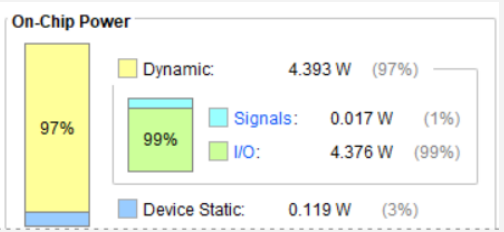
Resource	Utilization	Available	Utilization %
IO	7	210	3.33



AES Enable Off AES Enable On

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**4.512 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**45.6°C**



- For the implementation the amount of LUTs went down to 0
- The amount of power used was shown to be 4.512W, mainly from the 7 I/O ports that were used

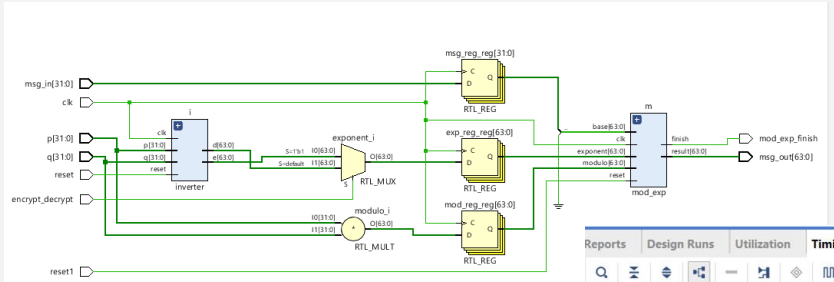
04

RSA (Rivest–Shamir–Adleman)

What is RSA?

RSA, named after its inventors Rivest, Shamir, and Adleman, is an asymmetric encryption algorithm widely employed for secure communication and digital signatures. In RSA, a pair of keys is generated—a public key for encryption and a private key for decryption. The security of RSA hinges on the challenge of factoring the product of two large prime numbers, forming the basis for its robust encryption. Users can encrypt data using the recipient's public key, and only the recipient, possessing the corresponding private key, can decrypt the message. Moreover, RSA's digital signature capabilities ensure data authenticity and integrity, with the sender using their private key to create a signature that anyone can verify using the sender's public key.

RSA Synthesis on Board



Reports | Design Runs | DRC | Methodology | Power | Timing | Utilization

Summary

Resource	Utilization	Available	Utilization %
LUT	25141	32600	77.12
FF	754	65200	1.16
DSP	44	120	36.67
IO	165	210	78.57

Reports | Design Runs | Utilization | Timing

Timing Checks - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Logic %	Net %
(none) (10)											
Path 11	∞	1168	1168	131	i/b_reg[0]/C	i/y_next0__3/B[12]	339.878	249.932	89.945	73.5	26.5
Path 12	∞	1167	1167	131	i/b_reg[0]/C	i/y_next0__3/B[8]	339.764	249.818	89.945	73.5	26.5
Path 13	∞	1166	1166	131	i/b_reg[0]/C	i/y_next0__3/B[4]	339.650	249.704	89.945	73.5	26.5
Path 14	∞	1177	1177	64	m/result_reg_reg[16]/C	m/result_mul_base__0/B[10]	339.638	250.674	88.963	73.8	26.2
Path 15	∞	1168	1168	131	i/b_reg[0]/C	i/y_next0__3/B[11]	339.633	249.853	89.779	73.6	26.4
Path 16	∞	1177	1177	64	m/result_reg_reg[16]/C	m/result_mul_base__0/B[12]	339.619	250.655	88.963	73.8	26.2
Path 17	∞	1168	1168	131	i/b_reg[0]/C	i/y_next0__3/B[10]	339.590	249.950	89.639	73.6	26.4
Path 18	∞	1177	1177	64	m/result_reg_reg[16]/C	m/result_mul_base__0/B[11]	339.546	250.582	88.963	73.8	26.2
Path 19	∞	1165	1165	131	i/b_reg[0]/C	i/y_next0__3/B[0]	339.536	249.590	89.945	73.5	26.5
Path 20	∞	1177	1177	64	m/result_reg_reg[16]/C	m/result_mul_base__0/B[9]	339.525	250.561	88.963	73.8	26.2

- This RSA implementation is far more resource intensive than either of symmetrical encryption algorithms in terms of LUTs, FFs, and DSPs
- The highest delay through this program is 339.878ns. This means the maximum frequency is around 1/339.878ns ≈ 2.94 MHz

05
Results & Conclusions

So overall...

- The RSA verilog code was intended for user input. Attempting to create a wrapper for the RSA algorithm proved to be difficult so the synthesized results were used for benchmarking instead
- Based off these results, we can see that AES is an improvement to DES in terms of resource usage. Additionally, the AES algorithm has a smaller maximum path in comparison to the DES algorithm
- Overall, both DES and AES algorithms had similar maximum frequencies (approximately 146 MHz and 187 MHz respectively), while the RSA algorithm had a much slower maximum frequency in comparison to them (approximately 3 MHz)
- These results provide a good insight on the structural design of symmetric and asymmetric cryptographic algorithms and provide numerical evidence of both the complexity and speeds of both types of systems where symmetric algorithms generally use less LUTs, FFs, and etc. and have faster speeds, and asymmetric algorithms use more LUTs, FFs, and etc, and have slower speeds

Resources

DES Verilog Code:
https://github.com/jpszczolowski/des-verilog

AES Verilog Code:
https://github.com/michaelehab/AES-Verilog

RSA Verilog Code:
https://github.com/Rajandeep/RSA-CRYPTOSYSTEM-using-verilog