HW 2 ECE 9453
Implementation of Advance Encryption System

Mudit Bhargava (mb8630) & Priyanka Johri (pj2167)

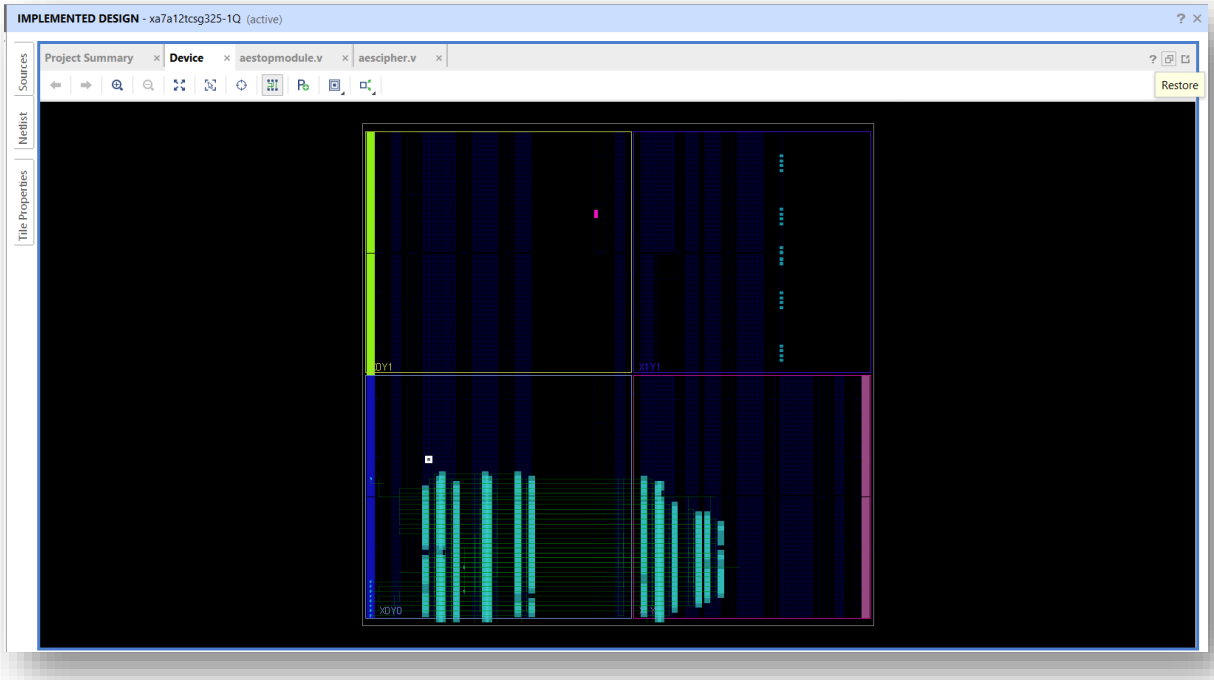
Determine the throughput, latency, and delay for each architecture.

a. Throughput: The number of encrypted or decrypted text produced every second.
Ans. 10 (It is not a pipelines structure; it is an iterative structure)

b. Latency: The number of clock cycles required to perform one encryption or decryption
Ans. 10 (Number of encryption and decryption cycles)

c. Area: Area on FPGA consumed by the architectures.
Ans. 1384

Name	Slice LUTs (8000)	Slice Registers (16000)	F7 Muxes (7300)	F8 Muxes (3650)	Block RAM Tile (20)	Bonded IOB (150)	BUFGCTRL (32)
aestopmodule	1384	1163	353	151	2	9	1
x1 (aes)	1384	1159	353	151	2	0	0
round1 (fir...	0	0	288	135	0	0	0
f0 (Key...	0	0	48	15	0	0	0
f1 (Sbox)	0	0	240	120	0	0	0



Project Summary

Overview | **Dashboard** | + Add Gadget

Utilization (synth_1, Synth Design)

Report	LUT	FF	BRAM	I/O	BUFG
synth_1, Synth Design	17.300%	7.269%	10.000%	6.000%	3.125%

DRC (impl_1, Route Design)

Report	Warning	Critical Warning
impl_1, Route Design	1	2

Power (impl_1, Route Design)

Report	Signals	Logic	BRAM	I/O	PL Static	Total Power
impl_1, Route Design	39.912	26.405	0.826	7.604	0.266	75.012

Utilization (impl_1, Place Design)

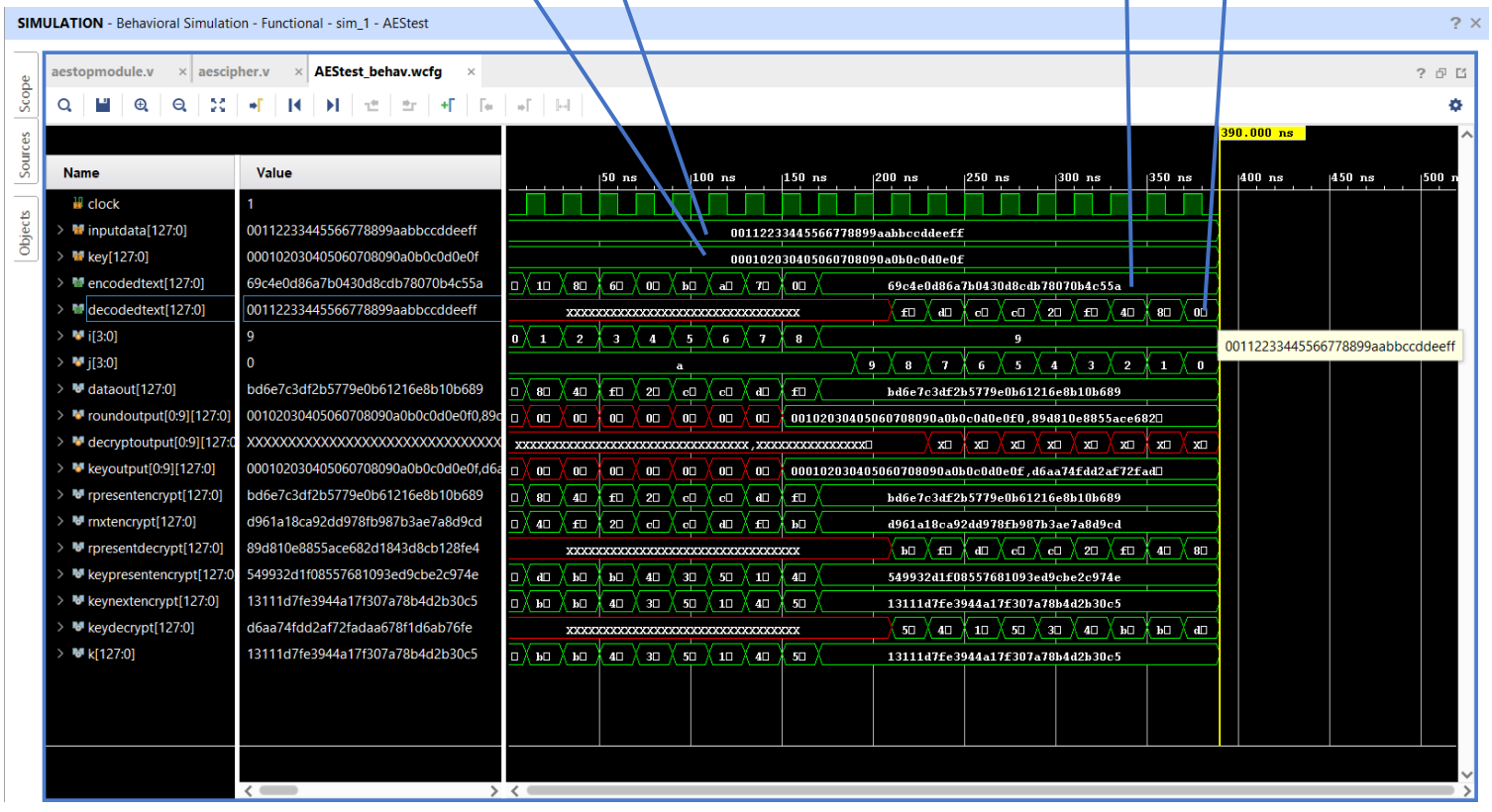
Report	LUT	FF	BRAM	I/O	BUFG
impl_1, Place Design	17.188%	7.269%	10.000%	6.000%	3.125%

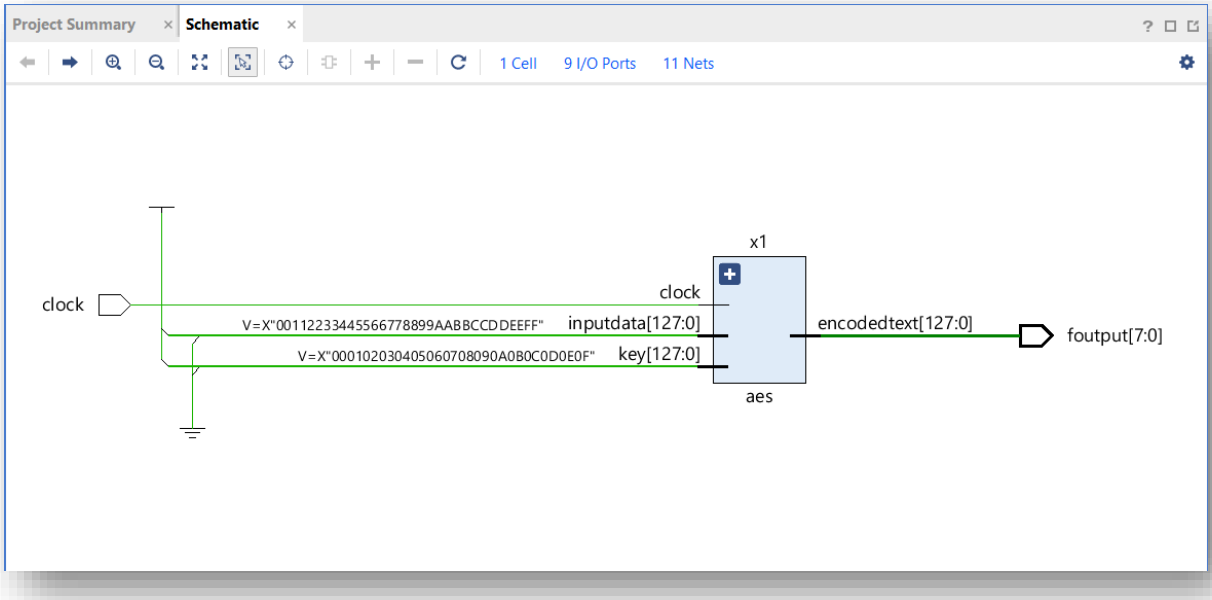
Plain Text: 00112233445566778899aabbccddeeff

Output: 00112233445566778899aabbccddeeff

Key: 000102030405060708090a0b0c0d0e0f

Cipher Text: 69c4e0d86a7b0430d8cdb78070b4c55a





Working:

The AES is an incremental cipher. It incorporates substitution–a permutation network. It consists of a sequence of connected processes, some of which require exchanging specified inputs for outcomes (substitutions) and others requiring shuffling bits around (permutations).

All of AES's calculations are done in bytes rather than bits. As a result, AES considers a plaintext block's 128 bits as 16 bytes. For matrix processing, these 16 bytes are organised into four columns and four rows.

Both encryption and decryption require 10 cycles.

In our code inside module ‘AES’, in the first 9 rounds i goes from 0 to 8 and the last round occurs at i = 9. The result of the last round acts as an input for the decryption module.

In the first 9 rounds there are 4 modules

- Key generation
- Byte Substitution
- Shift Rows
- Mix columns

The last round is the same as the first 9 rounds, just the mix column module is absent.

Byte Substitution

By looking up a predefined table (S-box) provided in design, the 16 input bytes are updated. A four-row, four-column matrix is the outcome.

Shift rows

Each of the matrix's four rows is shifted to the left. Any 'breaking off' entries is re-inserted on the right-hand side of the row.

The first row is kept the same.

The second row has been moved one (byte) to the left.

The third row has been moved two spaces to the left.

The fourth row has been moved three spaces to the left.

The outcome is a new matrix made up of the same 16 bytes that are being shifted in regard to each other.

Mix Columns

A particular mathematical function is now used to alter each column of four bytes. This function takes four bytes from one column as input and returns four entirely new bytes that replace existing column. As a result, a new matrix with 16 additional bytes is created.

Addroundkey

The matrix's 16 bytes are now treated as 128 bits, and they are XORed with the round key's 128 bits. The ciphertext is the output if this is really the final round. Otherwise, the 128 bits are interpreted as 16 bytes, and the cycle begins again.

Decryption Process

Whenever i has reach >=9 i.e., when decryption starts and j goes from 9 to 0.

When j==0 decryption stops.

The reverse process of decrypting an AES ciphertext is analogous to the reverse process of encryption. Each round consists of four processes that are performed in reverse order.

Last key is getting stored ask otherwise it is being stored in the keyout.

At last, the decrypted data is XORed with Key.

TEST BENCH

The top module is called in the test bench and is given a clock.

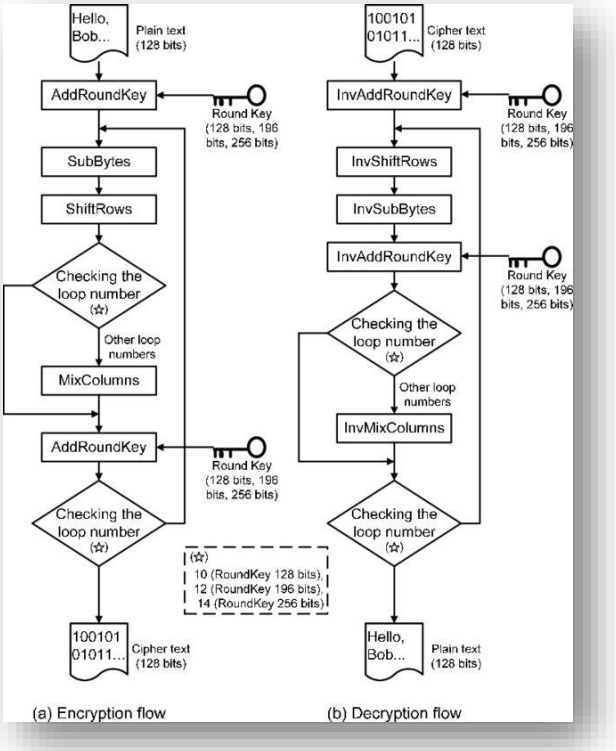


Fig Reference: Kumaki, Takeshi & Fujita, Tomohiro & Nakanishi, Mamoru & Ogura, Takeshi. (2013). "*Morphological pattern spectrum and block cipher processing based image-manipulation detection.*" Nonlinear Theory and Its Applications, IEICE. 4. 400-418. 10.1587/nolta.4.400.