Lab3 : Sequential Circuit – AES operation

Advisor : Lih-Yih Chiou

Lecturer : David

Date : 2024/03/07

EE

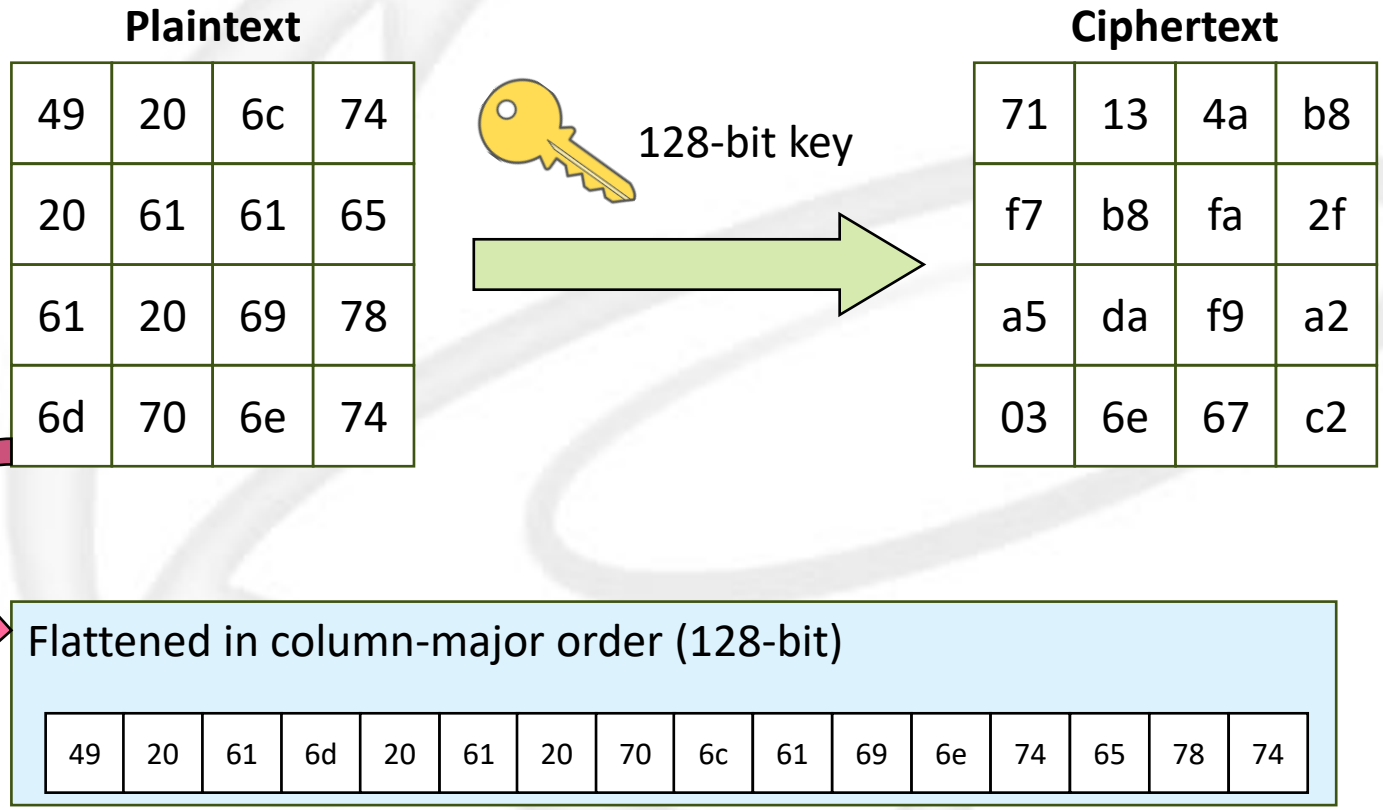
LPHPLAB VLSI Design LAB

Outline

- Introduction
- Design Specifications
- System Description
- Criteria

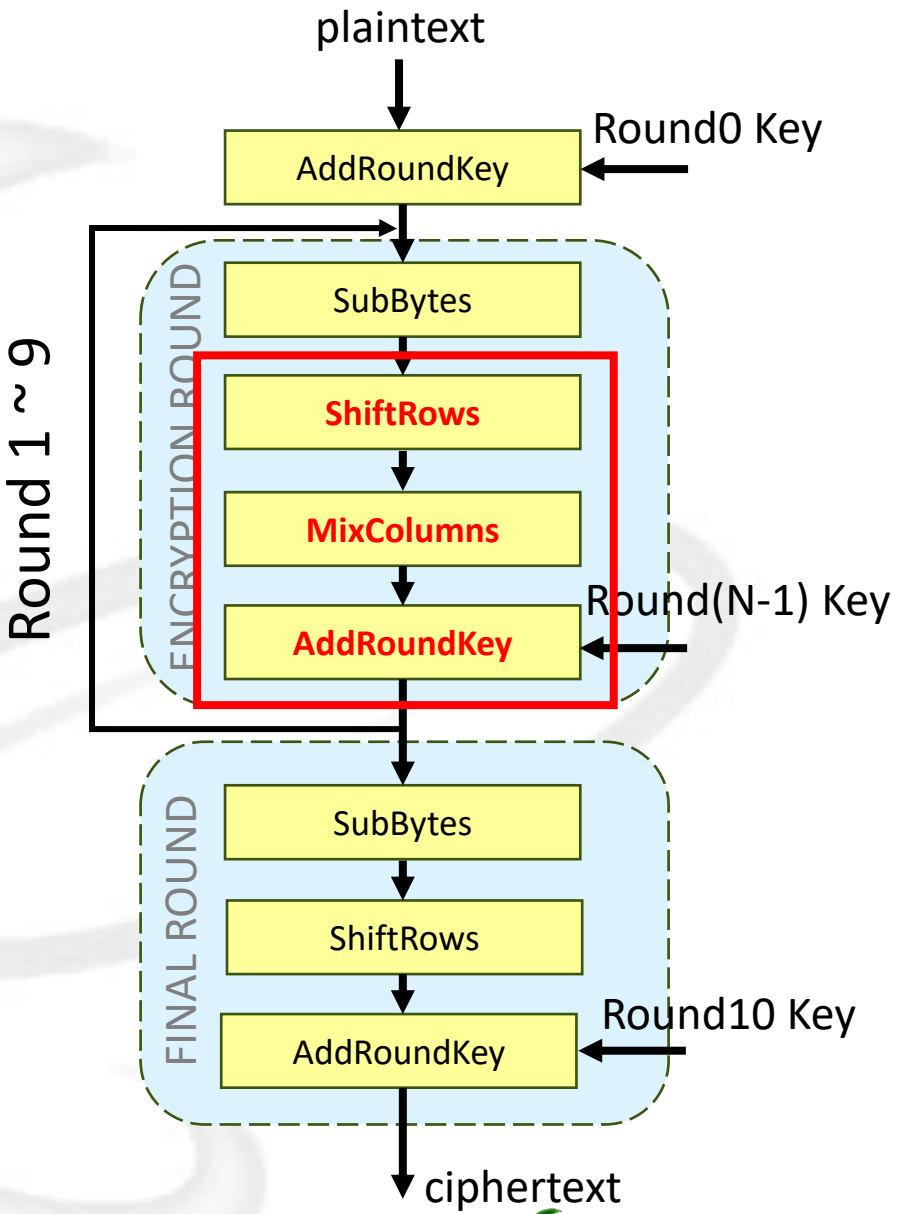
Introduction (1/2)

- AES, a variant of Rijndael, features a consistent block size of 128 bits and supports key sizes of 128, 192, 256 bits. Most AES computations are done in a specific finite field.



Introduction (2/2)

- AES encryption flow:
  - A total of 10 rounds are performed, with no MixColumns transformation required in the final round.
  - In this lab, we will focus on completing three operations in AES encryption.
  - These three operations are sequentially connected, each subsequent computation taking the previous computation result as its input.

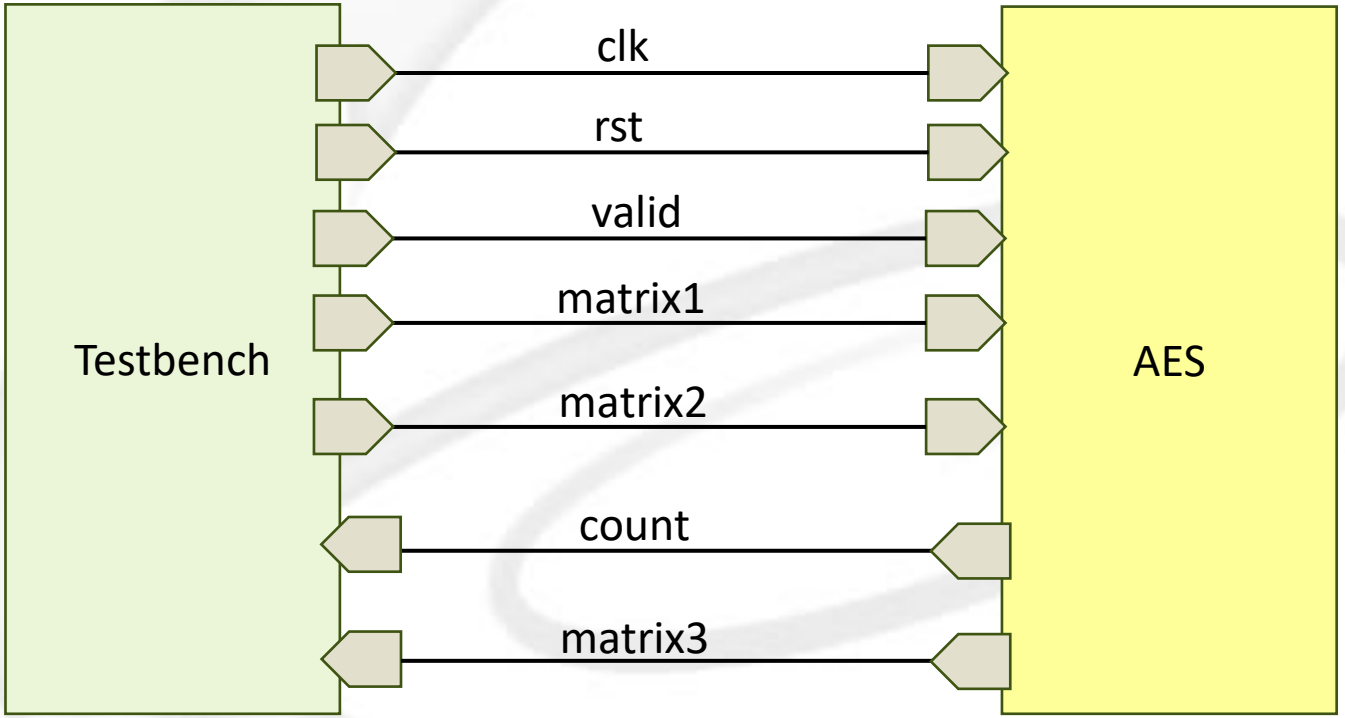


Outline

- Introduction
- Design Specifications
- System Description
- Criteria

Design Specifications (1/2)

Block Diagram



Design Specifications (2/2)

I/O information

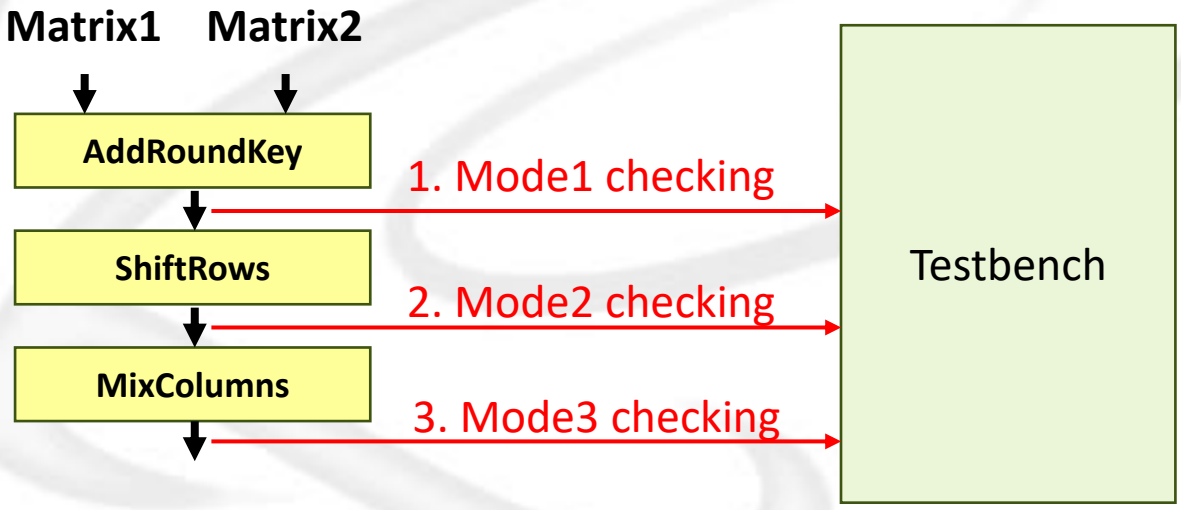
Signal	I/O	width	Description
clk	I	1	Clock signal (positive edge trigger)
rst	I	1	Synchronous reset signal (active high)
valid	I	1	Specify that the testbench is transmitting matrix1 and matrix2.
matrix1	I	128	A 4*4 byte matrix, flattened in column-major order. (plaintext)
matrix2	I	128	A 4*4 byte matrix, flattened in column-major order. (key)
count	O	2	Display to testbench for the operation you are executing. ➢ Mode 0: In default mode, the outcome is meaningless. ➢ Mode 1: AddRoundKey ➢ Mode 2: ShiftRows ➢ Mode 3: MixColumns
matrix3	O	128	The result of each operation is a 4*4 byte matrix, flattened in column-major order.

Outline

- Introduction
- Design Specifications
- System Description
- Criteria

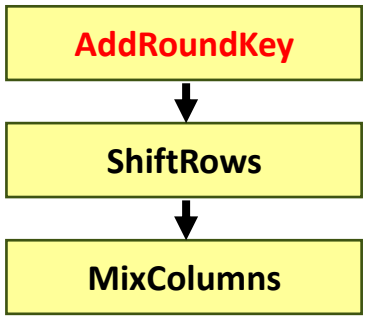
System Description (1/6)

- We consider plaintext as Matrix1 and key as Matrix2.
- Chaining three operations together, you need to directly change the mode pattern to inform testbench for which operation it is done currently.



System Description (2/6)

- AddRoundKey: bitwise XOR operation



31	2d	2b	2c
29	23	21	22
2f	2c	2b	2e
35	30	28	2d

Matrix 1

54	49	65	74
68	73	63	4b
69	41	72	65
73	53	65	79

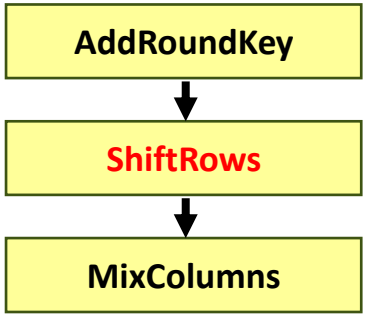
Matrix 2

65	64	4e	58
41	50	42	69
46	6d	59	4b
46	63	4d	54

Matrix 3

System Description (3/6)

- ShiftRows: shift array circularly



65	64	4e	58
41	50	42	69
46	6d	59	4b
46	63	4d	54

Current Matrix 3

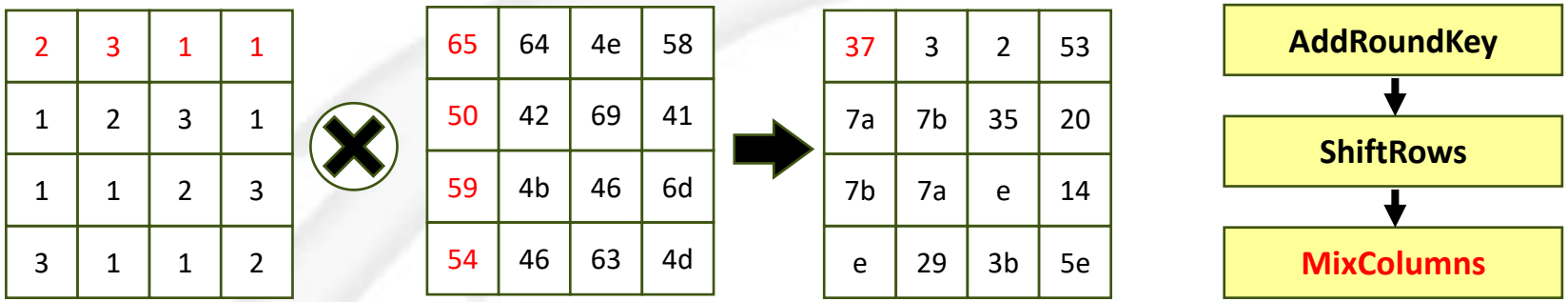
65	64	4e	58
50	42	69	41
59	4b	46	6d
54	46	63	4d

Next Matrix 3

← fixed
← Left rotate 1
← Left rotate 2
← Left rotate 3

System Description (4/6)

MixColumns: matrix multiplication and XOR addition



Constant Matrix | Current Matrix 3 | Next Matrix 3

1. Polynomial Multiplication:

- $\{2\}_{16} * \{65\}_{16} = \{10\}_2 * \{01100101\}_2$
  $\rightarrow (x) * (x^6 + x^5 + x^2 + 1) = (x^7 + x^6 + x^3 + x)$
  $\rightarrow \{11001010\}_2$
- $\{1\}_{16} * \{54\}_{16} = \{1\}_2 * \{1010100\}_2$
  $\rightarrow \{1010100\}_2$
- $\{3\}_{16} * \{50\}_{16} = \{11\}_2 * \{1010000\}_2$
  $\rightarrow (x + 1) * (x^6 + x^4) = (x^7 + x^6 + x^5 + x^4)$
  $\rightarrow \{11110000\}_2$
- $\{1\}_{16} * \{59\}_{16} = \{1\}_2 * \{1011001\}_2$
  $\rightarrow \{1011001\}_2$

2. Addition:

$\{2\}_{16} * \{65\}_{16} + \{3\}_{16} * \{50\}_{16} + \{1\}_{16} * \{54\}_{16} + \{1\}_{16} * \{59\}_{16}$
$= \{11001010\}_2 \oplus \{11110000\}_2 \oplus \{1010100\}_2 \oplus \{1011001\}_2 = \{00110111\}_2 = \{37\}_{16}$

Recommend of Hardware implementation:
Understanding AES Mix-Columns Transformation Calculation

Appendix: Finite Field (1/2)

- The finite field with p^n element is denote as GF(p^n), where p is a prime number, and n represents polynomials of a specific degree.

  GF(3) used in integer:

  - $4 \equiv 1 \pmod 3$, $4+2 \equiv 0 \pmod 3$, $1-2 \equiv 2 \pmod 3$

  GF(5^3) used in polynomials (mod to coefficient):

  - $(3x^2 + 4x + 2) + (4x^2 + x + 1) = 7x^2 + 5x + 3 \equiv 2x^2 + 3 \pmod 5$

- p = 2, the finite field GF(2^n) is a special case that addition is XOR logic and multiplication is AND logic.
  - Addition: $(x+1) + (x^2+x) \rightarrow \{011\}_2 \oplus \{110\}_2 = \{101\}_2 \rightarrow x^2+1$
  - Multiplication: $(x+1) * (x^2+x+1) = x^3 + x^2 + x + x^2 + x + 1 \equiv x^3 + 1 \pmod 2$

Appendix: Finite Field (2/2)

- MixColumns: GF(2^8) used in polynomial
  - In binary system, each pixel is represented by a 1-byte value.
  - Limit polynomial coefficients to be 0 or 1. (coefficient mod by 2)
  - Limit the degree of the polynomial by ''modding'' it by a polynomial of degree 8. (In AES the mod polynomial is $x^8 + x^4 + x^3 + x^1 + 1$)
- Example:
  - $(100000000)_2$ => x^8 (The degree exceeds 8)

    $x^8 = (x^8 + x^4 + x^3 + x + 1)*1 + (x^4 + x^3 + x + 1)$

    AES modulus

    $\therefore\ x^8 \equiv x^4 + x^3 + x + 1 \pmod{x^8+x^4+x^3+x^1+1}$

  - $\{2\}_{16} * \{d4\}_{16} = \{10\}_2 * \{11010100\}_2$

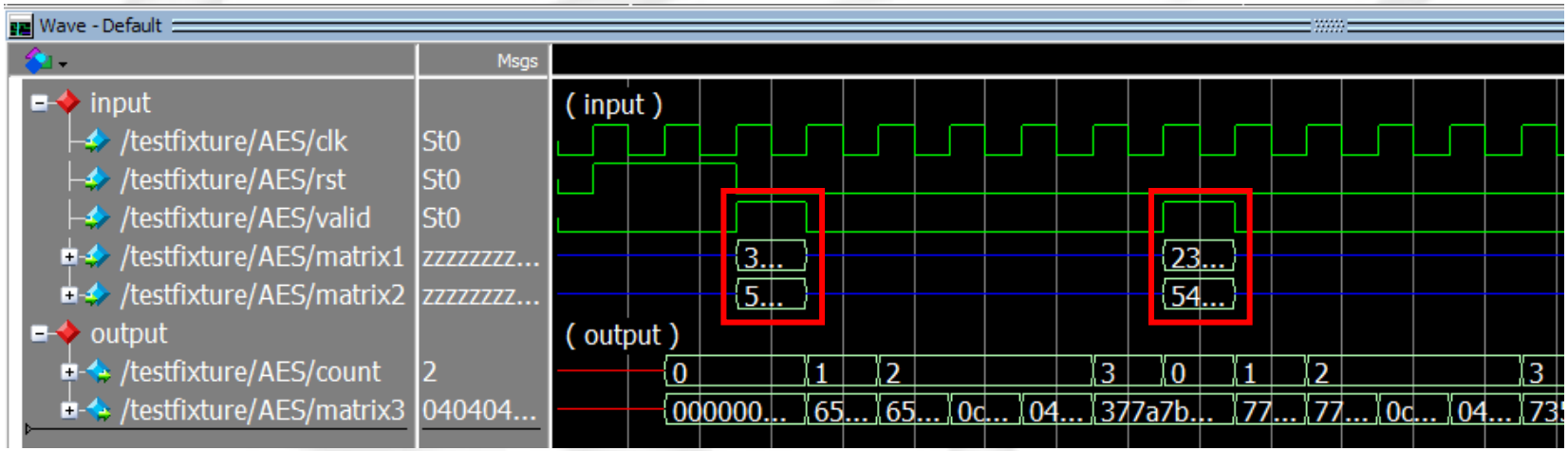
    $\rightarrow x*(x^7 + x^6 + x^4 + x^2) = x^8 + x^7 + x^5 + x^3$

    $= (x^4 + x^3 + x + 1) + (x^7 + x^5 + x^3) \equiv (x^7 + x^5 + x^4 + x + 1) \pmod 2$

    $\rightarrow \{10110011\}_2 = \{b3\}_{16}$

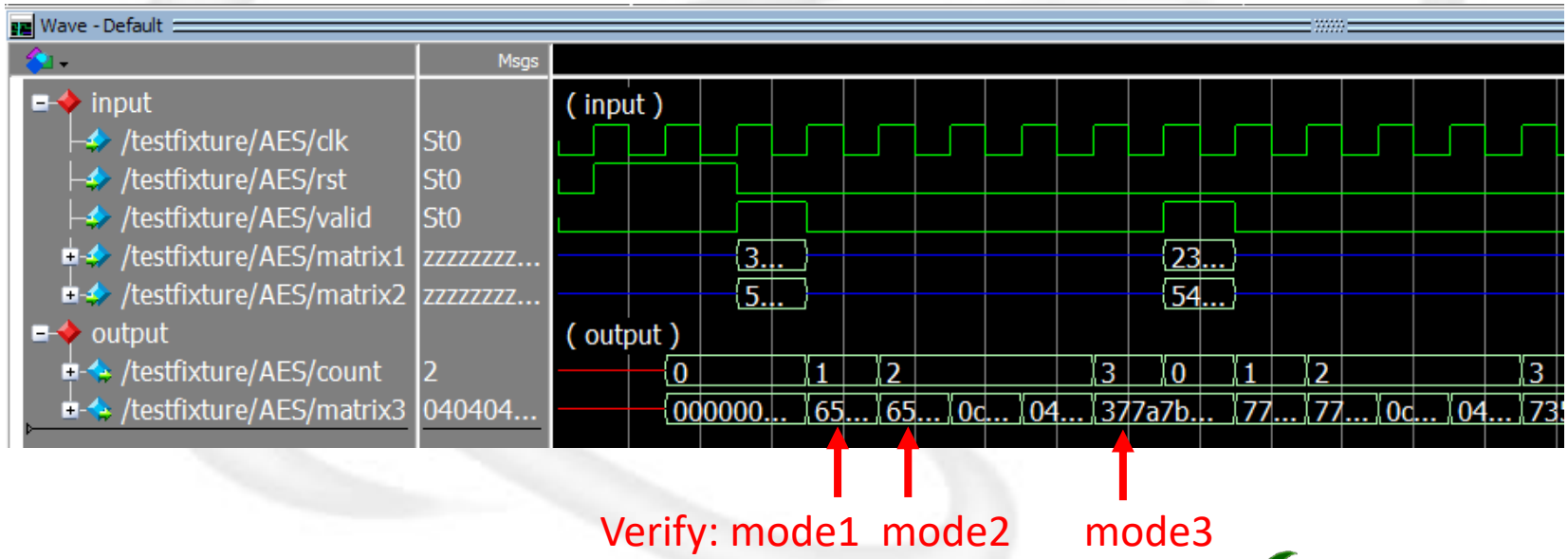
System Description (5/6)

- The first pattern will be sent promptly upon the assertion of the ‘rst’ signal to a low state. And each pattern will be transmitted within a single cycle.
- Once the testbench detects that the output signal ‘count’ equals 3, the testbench will proceed to send the next pattern.



System Description (6/6)

- Once the output signal ‘count’ transitions to the specified operational mode, the testbench will promptly verify the value of ‘matrix3’.
- It is not a requirement for each operation to be completed within a single clock cycle.
- The count signal should repeat in the order 0, 1, 2, 3



Outline

- Introduction
- Design Specifications
- System Description
- Criteria

Criteria (1/3)

- Grading policy(100%)
  - Lab3
    - Simulation pass (90%)
      - AddRoundKey pass (10%)
      - ShiftRows pass (20%)
      - MixColumns pass (60%)
    - Report (10%)

Criteria (2/3)

Simulation result

Pass

```
# -------- Simulation report --------
# AddRoundKey Operation ERROR amount:           0
# ShiftRows   Operation ERROR amount:           0
# MixColumns  Operation ERROR amount:           0
#
#
#
#
#
#          ****************************
#          **                        **       |__||
#          **  Congratulations !!    **      / O.O  |
#          **                        **    /_____   |
#          **  Simulation PASS!!     **   /^ ^ ^ \  |
#          **                        **  |^ ^ ^ ^ |w|
#          ****************************   \m___m__|_|
#
#
#
#
# ---------- Your score: 90/90 ----------
# ** Note: $stop    : C:/Users/david/Desktop/HDL_Lab3/tb.sv(150)
#    Time: 184355 ns  Iteration: 1  Instance: /testfixture
```

Your score

Failed

```
# Pattern 3072/3072
# AddRoundKey Operation: Correct, ShiftRows Operation: Correct, MixColumns Operation:  Error
#
#
#
# -------- Simulation report --------
# AddRoundKey Operation ERROR amount:           0
# ShiftRows   Operation ERROR amount:           0
# MixColumns  Operation ERROR amount:        3072
#
# -> MixColumns Operation: the first error was detected in Pattern    0
# Your matrix:
# 7a  7b  35  20
# 7a  7b  35  20
# 7b  7a  0e  14
# 0e  29  3b  5e
#
# Golden matrix:
# 37  03  02  53
# 7a  7b  35  20
# 7b  7a  0e  14
# 0e  29  3b  5e
#
#
#
#
#
#          ****************************
#          **                        **       |__||
#          **  OOPS!!                **      / X,X  |
#          **                        **    /_____   |
#          **  Simulation Failed!!   **   /^ ^ ^ \  |
#          **                        **  |^ ^ ^ ^ |w|
#          ****************************   \m___m__|_|
#
#
#
# ---------- Your score: 30/90 ----------
# ** Note: $stop    : C:/Users/david/Desktop/HDL_Lab3/tb.sv(150)
#    Time: 184355 ns  Iteration: 1  Instance: /testfixture
```

Display the first detected error for each operation and print the matrix for debugging.

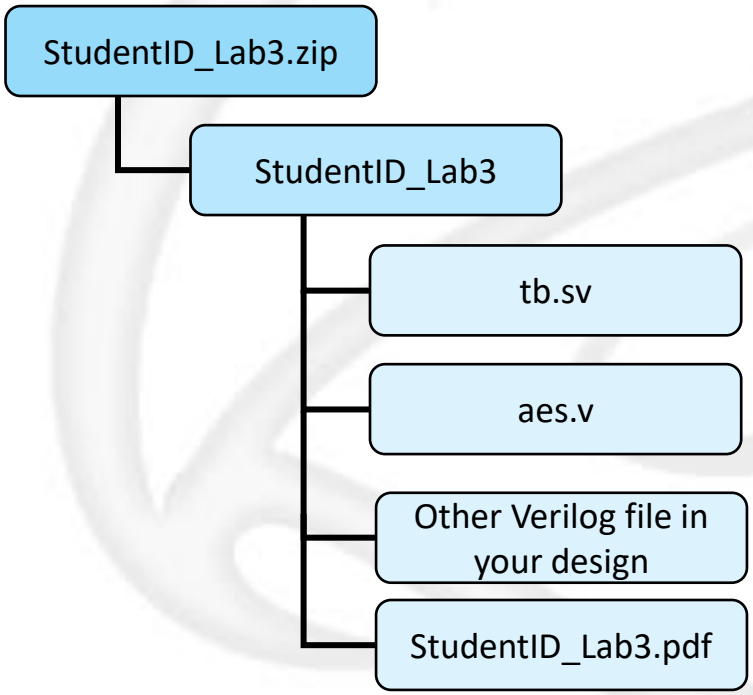
Your score

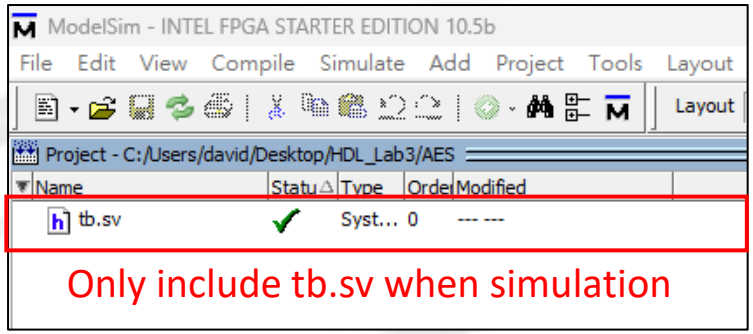
Criteria (3/3)

- Friendly reminder
  - Please complete the assignment by your own, discussion with peers is recommended, but do not cheat.
  - Warning! Any dishonesty found will result in zero grade.
  - Warning! Any late submission will also receive zero.
  - Warning! Please make sure that your code can be compiled in Modelsim, any dead body that we cannot compile will also receive zero.
  - Warning! Please submit your work according to the specified file format, making sure not to include any unnecessary files. Any unnecessary file found, will lead to 10% deduction from the overall score.
- Deadline: 2024/03/14 8:59 a.m.

Lab3 Requirement & file format

- File format
  - ➔ We will only include ‘tb.sv’ in ModelSim project to verify your design. Be cautious about the other file path you included in top module ‘aes.v’.





LPHPLAB
VLSI Design LAB

Thanks for your attention !!