PA2 Explanation

Andy, Yu-Guang Chen
Associate Professor, Department of EE
National Central University
andyygchen@ee.ncu.edu.tw
Slides Credit: TA Wen-Ti, Tsai

Outline

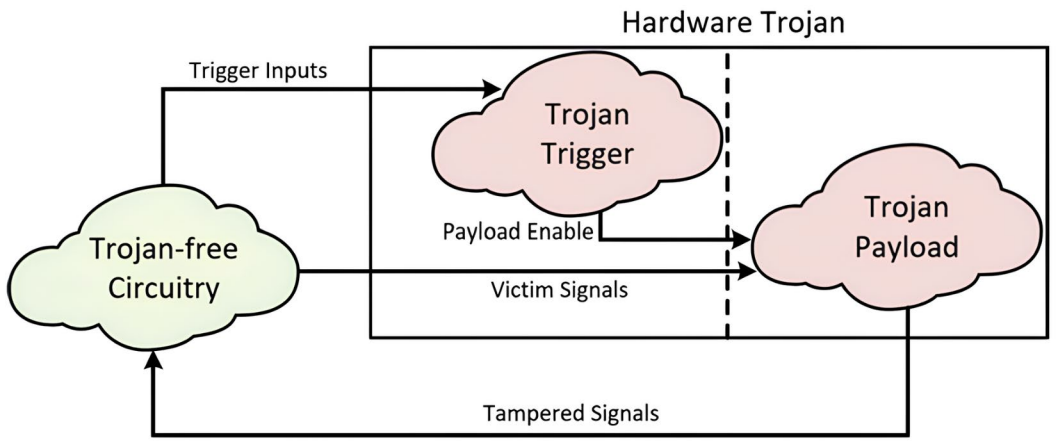
- Background
- Problem Description
- Part 1: AES
- Part 2: Hardware Trojan
- Submission Requirement
- Evaluation

Outline

- Background
- Problem Description
- Part 1: AES
- Part 2: Hardware trojan
- Submission Requirement
- Evaluation

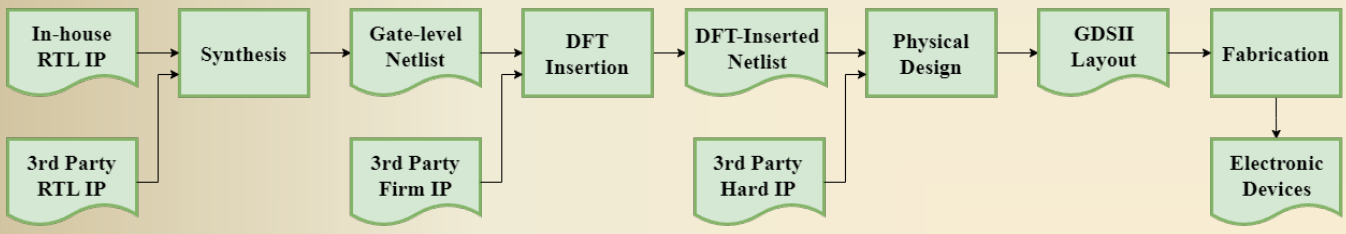
Background

- What is hardware Trojan?



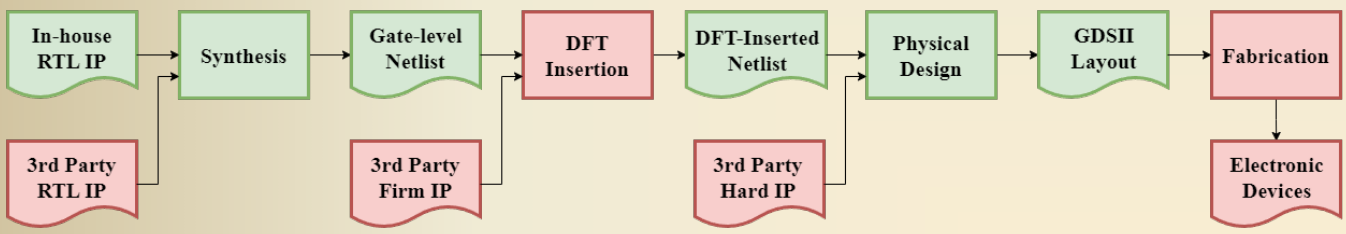
Background

- Modern chip design flow
  - Collaboration among SoC integrators, 3PIP vendors, and offshore foundries



Background

- Hardware Trojan attacks
  - Malicious additions or modifications to ICs



Outline

- Background
- Problem Description
- Part 1: AES
- Part 2: Hardware trojan
- Submission Requirement
- Evaluation

Problem Description

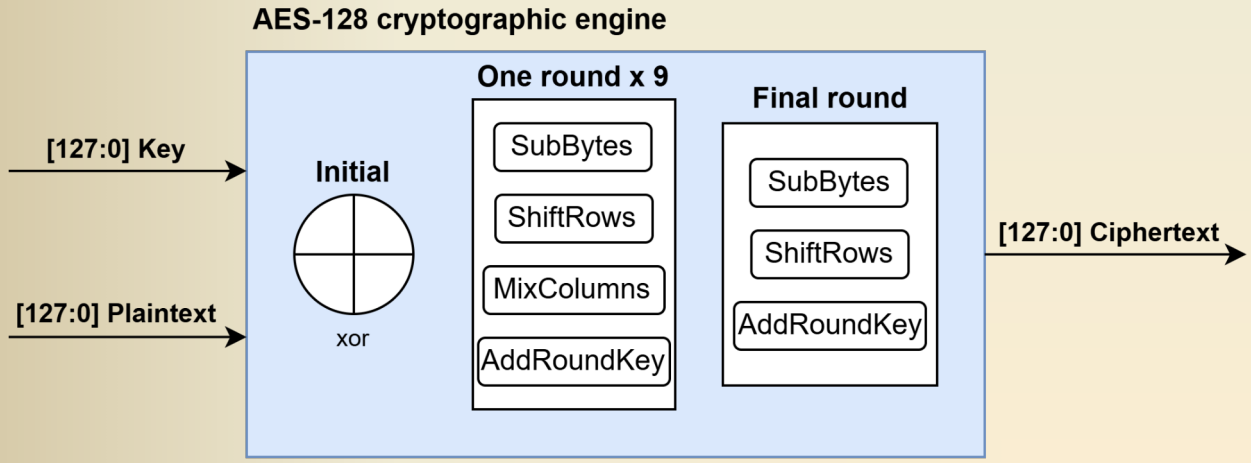
- Part 1 : Complete the AES-128 system
- Part 2 : Implement the hardware Trojan you designed in this system
  - Implement the sample hardware Trojan
  - Implement the hardware Trojan described in the paper *"A Novel Tampering Attack on AES Cores with Hardware Trojans"*
  - (Bonus) Design and implement your own novel hardware Trojan

Outline

- Background
- Problem Description
- Part 1: AES-128
- Part 2: Hardware Trojan
- Submission Requirement
- Evaluation

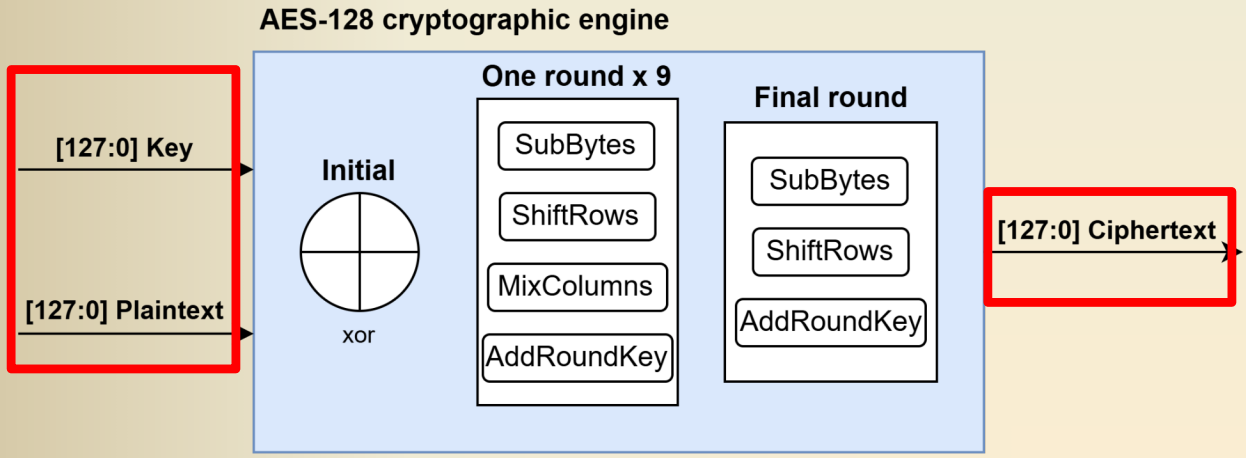
Part 1: AES-128

- Implement an AES-128 cryptographic engine



Part 1: AES-128

◆Data elements explanation



Part 1: AES-128

◆Functional explanation

key[127:0]

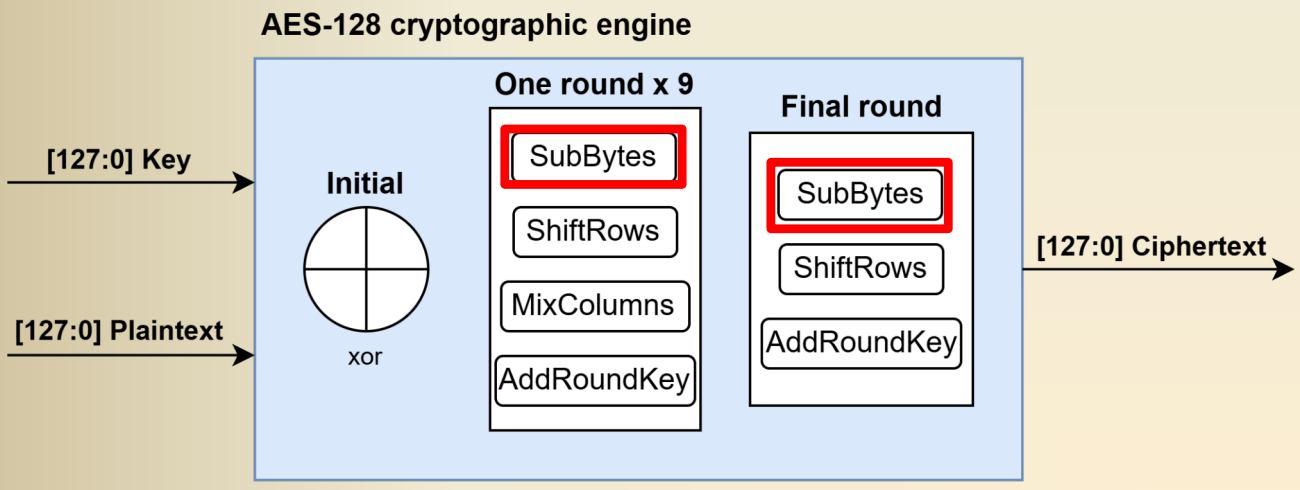
[127:120]	[119:112]	[111:104]	[103:96]
[95:88]	[87:80]	[79:72]	[71:64]
[63:56]	[55:48]	[47:40]	[39:32]
[31:24]	[23:16]	[15:8]	[7:0]

state[127:0]

[127:120]	[119:112]	[111:104]	[103:96]
[95:88]	[87:80]	[79:72]	[71:64]
[63:56]	[55:48]	[47:40]	[39:32]
[31:24]	[23:16]	[15:8]	[7:0]

Part 1: AES-128

◆Functional explanation - SubBytes



Part 1: AES-128

◆Functional explanation – SubBytes

➢module S in table.v

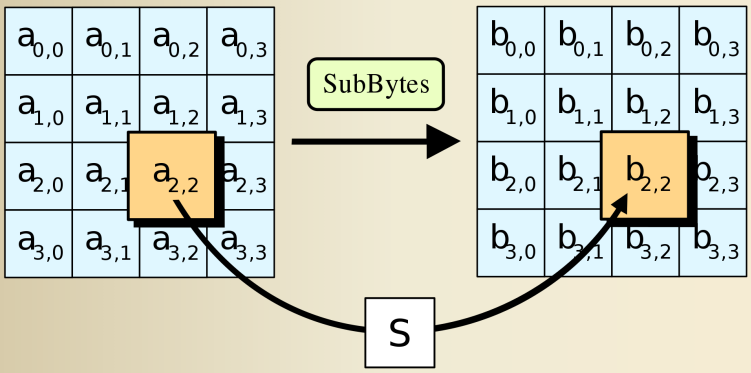


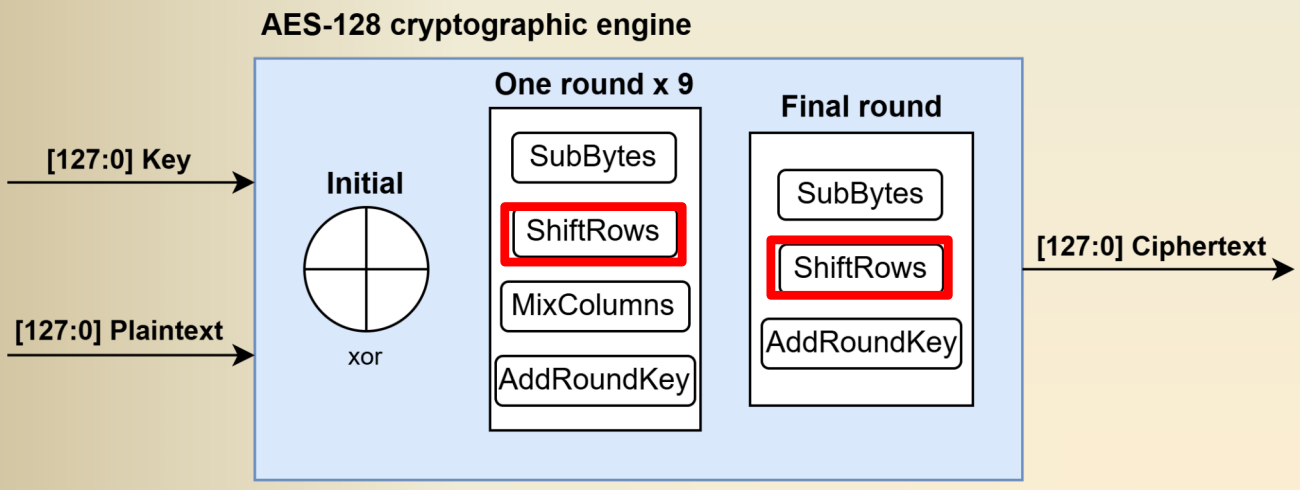
table.v

C: > Users > ASUSZE~1 > AppData > Local > Temp >

```
42
43  /* S box */
44  module S (clk, in, out);
45      input clk;
46      input [7:0] in;
47      output reg [7:0] out;
48
49      always @ (posedge clk)
50      case (in)
51      8'h00: out <= 8'h63;
52      8'h01: out <= 8'h7c;
53      8'h02: out <= 8'h77;
54      8'h03: out <= 8'h7b;
55      8'h04: out <= 8'hf2;
56      8'h05: out <= 8'h6b;
57      8'h06: out <= 8'h6f;
58      8'h07: out <= 8'hc5;
59      8'h08: out <= 8'h30;
60      8'h09: out <= 8'h01;
61      8'h0a: out <= 8'h67;
62      8'h0b: out <= 8'h2b;
63      8'h0c: out <= 8'hfe;
```

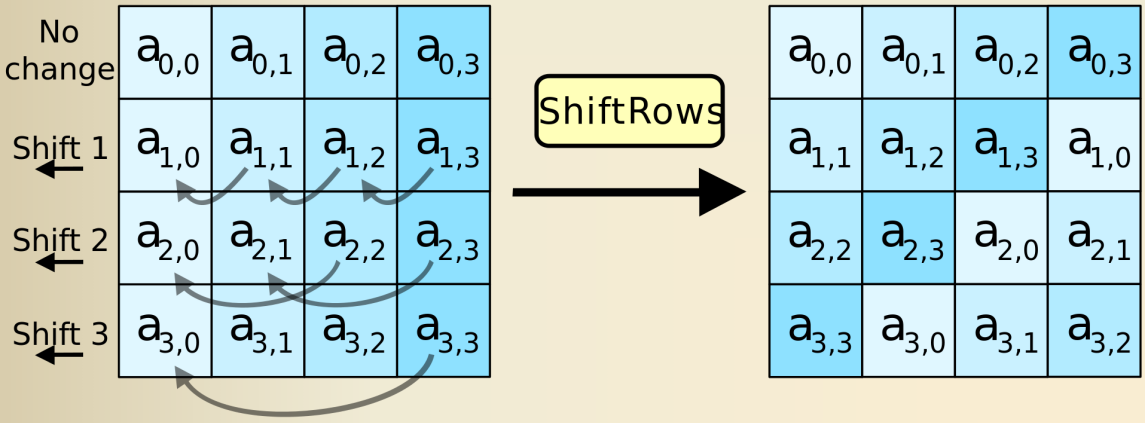
Part 1: AES-128

◆Functional explanation - ShiftRows



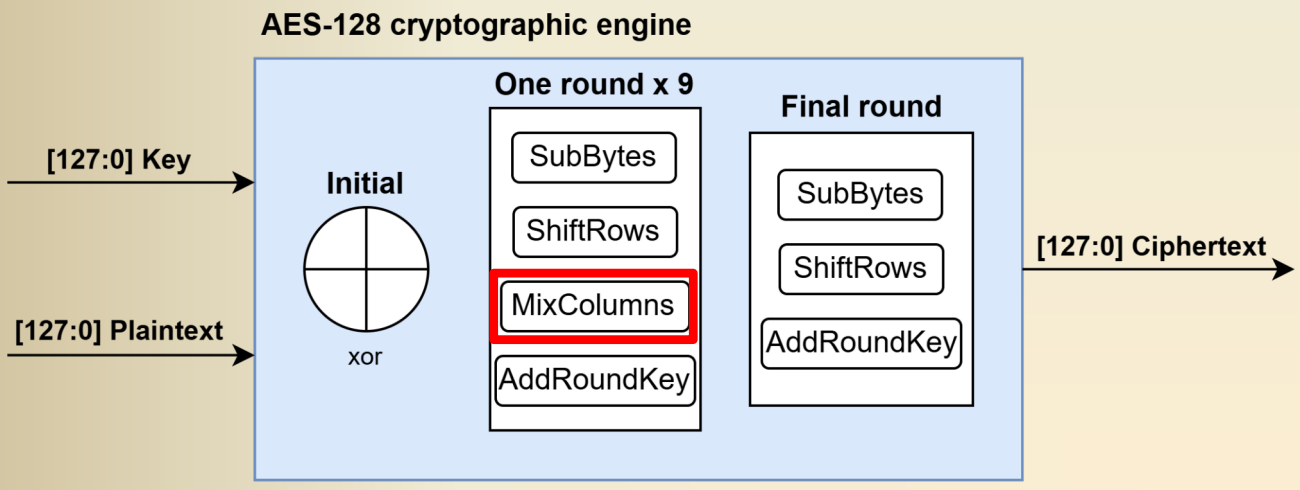
Part 1: AES-128

◆Functional explanation - ShiftRows



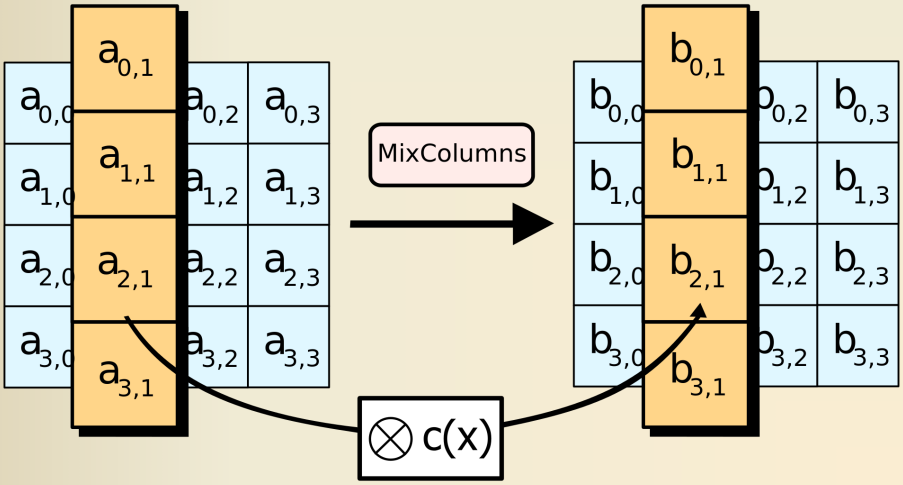
Part 1: AES-128

◆Functional explanation– MixColumns



Part 1: AES-128

◆Functional explanation– MixColumns



Part 1: AES-128

- Functional explanation– MixColumns
  - MixColumns ➔ module xS in table.v

table.v

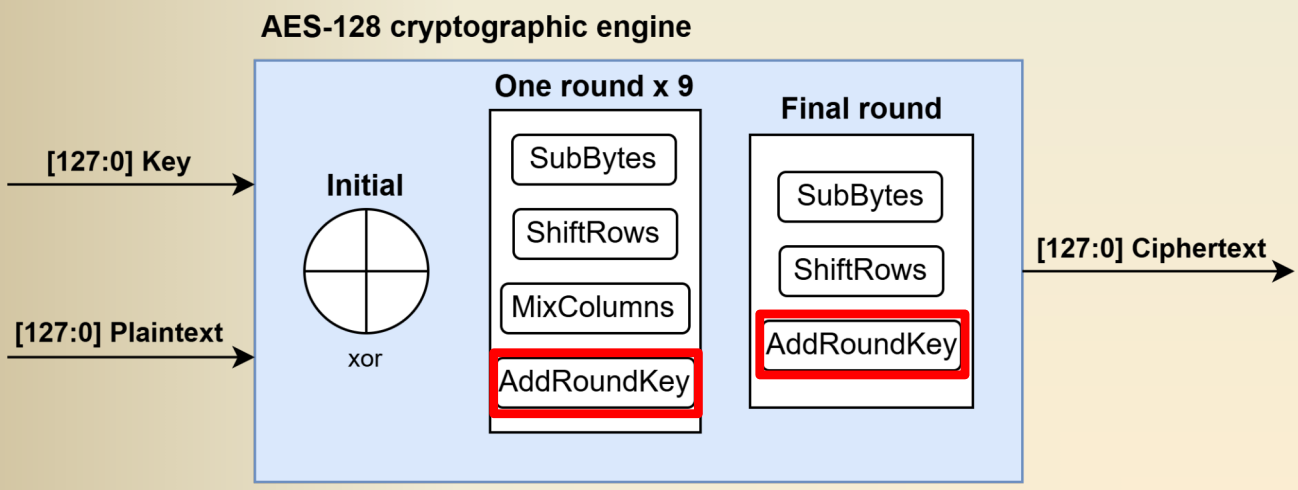
C: > Users > ASUSZE~1 > AppData > Local > Temp > Mxt21

```
/* S box * x */
module xS (clk, in, out);
    input clk;
    input [7:0] in;
    output reg [7:0] out;

    always @ (posedge clk)
    case (in)
    8'h00: out <= 8'hc6;
    8'h01: out <= 8'hf8;
    8'h02: out <= 8'hee;
    8'h03: out <= 8'hf6;
    8'h04: out <= 8'hff;
    8'h05: out <= 8'hd6;
    8'h06: out <= 8'hde;
```

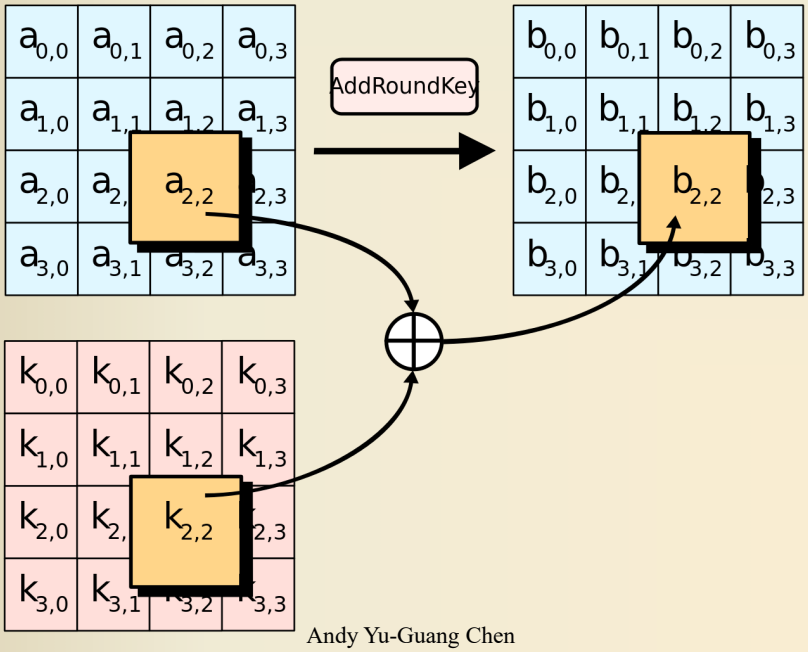

Part 1: AES-128

- Functional explanation–AddRoundKey



Part 1: AES-128

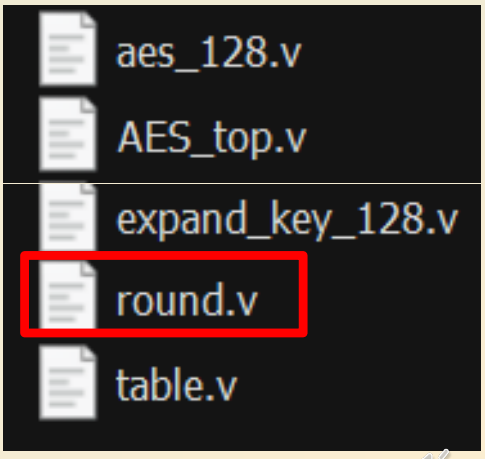
◆Functional explanation– AddRoundKey



Part 1: AES-128

◆ Your objective is to complete two modules in the file named as "round.v"

- module one_round
- module final_round



Part 1: AES-128

- round.v

round.v

```
: > Users > ASUSZE~1 > AppData > Local > Temp > Mxt215 > RemoteFiles > 593072_2_7 >
1   /* one AES round for every two clock cycles */
2   module one_round (clk, state_in, key, state_out);
3       input             clk;
4       input      [127:0] state_in, key;
5       output reg [127:0] state_out;
6
7   // SubBytes
8
9   // ShiftRows
10
11  // MixColumns
12
13  // AddRoundKey
14
15
16  endmodule
```

```
18  /* AES final round for every two clock cycles */
19  module final_round (clk, state_in, key_in, state_out);
20      input             clk;
21      input      [127:0] state_in;
22      input      [127:0] key_in;
23      output reg [127:0] state_out;
24
25  // SubBytes
26
27  // ShiftRows
28
29  // AddRoundKey
30
31
32  endmodule
```


Outline

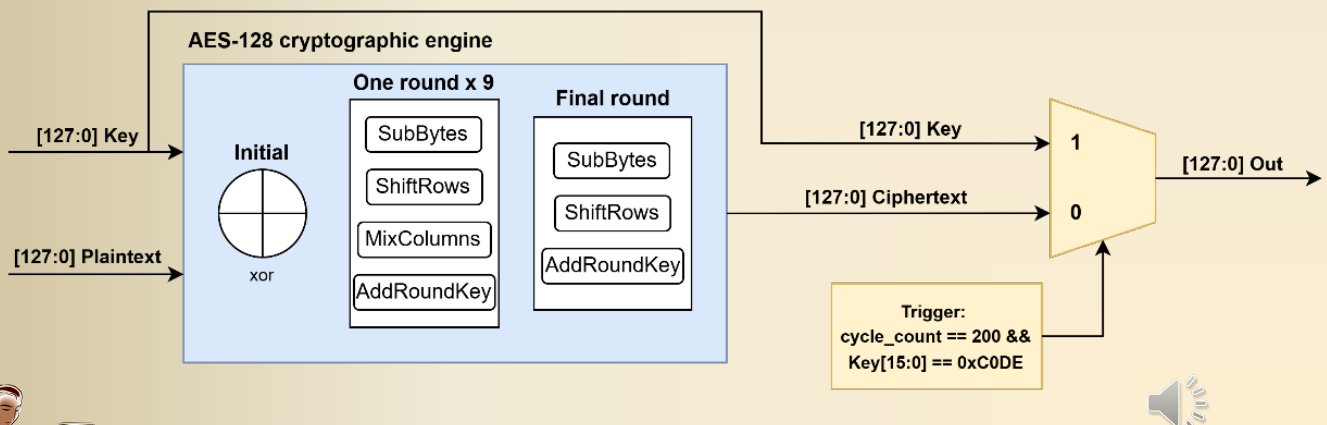
- Background
- Problem Description
- Part 1: AES-128
- Part 2: Hardware Trojan
- Submission Requirement
- Evaluation

Part 2: Hardware trojan

- Implement a hardware Trojan for an AES system. (Including triggers and payloads)
  - Implement the sample hardware Trojan
  - Implement the hardware Trojan described in the paper *"A_Novel_Tampering_Attack_on_AES_Cores_with_Hardware_Trojans"*
  - (Bonus) Design and implement your own novel hardware Trojan

Part 2: Hardware trojan

- Implement a hardware Trojan for an AES system
  - The sample hardware Trojan



Outline

- Background
- Problem Description
- Part 1: AES-128
- Part 2: Hardware trojan
- Submission Requirement
- Evaluation

Submission Requirement

- StudID_PA2_AES.zip
  - Related to the purely AES system in part one
- StudID_PA2_HT.zip
  - Related to the hardware Trojan you implemented in part two
- StudID_Name_PA2_report.pdf

Outline

- Background
- Problem Description
- Part 1: AES-128
- Part 2: Hardware trojan
- Submission Requirement
- Evaluation

Evaluation

- Complete the AES-128 cryptographic engine: 30%
- Sample hardware Trojan implement: 30%
- Reference hardware Trojan implement: 10%
- The report: 10%
- Demo: 20%
- Design a novel hardware Trojan and implement: 10% (bonus)

The report

- In your report, you have to include at least:
  - How to compile and execute your program
  - The completion of the assignment
  - The hardware Trojan you design (Including Trojan trigger and payload)
  - Your simulation waveform and explain it;
- We don’t restrict the report format and length

Demo session

- Attendance(8%)
- Three Questions(4% * 3 = 12%)

Andy, Yu-Guang Chen
Assistant Professor, Department of EE, NCU
Email: andyygchen@ee.ncu.edu.tw
FB: Yu-Guang Chen
IG: ncu.eda.andy
Google account: andyygchen.ncu