EE6094
CAD for VLSI Design

PA4: Hardware Trojan

Andy, Yu-Guang Chen
Assistant Professor, Department of EE
National Central University
andyygchen@ee.ncu.edu.tw
Slides Credit: TA李宗穎

Outline

- Verilog Simulation
- Examples
- Cases

Outline

- Verilog Simulation
- Examples
- Cases

Verilog Simulation

- How to use the Verilog Simulation on workstation?
  - Step1: Key in the following source commands.
    source /usr/cad/synopsys/CIC/customexplorer.cshrc
    source /usr/cad/cadence/CIC/incisiv.cshrc
    source /usr/cad/synopsys/CIC/verdi.cshrc
    source /usr/cad/synopsys/CIC/synthesis.cshrc
    source /usr/cad/synopsys/CIC/vcs.cshrc

```
[109521121@eda359_forclass ~]$ source /usr/cad/synopsys/CIC/customexplorer.cshrc
[109521121@eda359_forclass ~]$ source /usr/cad/cadence/CIC/incisiv.cshrc
[109521121@eda359_forclass ~]$ source /usr/cad/synopsys/CIC/verdi.cshrc
[109521121@eda359_forclass ~]$ source /usr/cad/synopsys/CIC/synthesis.cshrc
Platform = amd64
[109521121@eda359_forclass ~]$ source /usr/cad/synopsys/CIC/vcs.cshrc
```

Verilog Simulation

- How to use the Verilog Simulation on workstation?
  - Step2: You need to add some program syntax for testbench to generate waveform.
    - $fsdbDumpfile(“your_circuit.fsdb");
    - $fsdbDumpvars(0, your_circuit _tb);

```
1  //Half Adder Full Case Testbench
2  `timescale 1ns/1ns
3
4  module Half_Adder_full_tb;
5
6  reg   x, y;
7  wire  sum, carry;
8
9  Half_Adder ha1( .x(x) , .y(y) , .sum(sum) , .carry(carry) );
10
11 initial
12 begin
13
14   $fsdbDumpfile("Half_Adder.fsdb");
15   $fsdbDumpvars(0,Half_Adder_full_tb);
16   #0  x=1'b0; y=1'b0;
17   #5  x=1'b0; y=1'b1;
18   #5  x=1'b1; y=1'b0;
19   #5  x=1'b1; y=1'b1;
20   #5  $finish;
21
22 end
23
24 endmodule
```

Verilog Simulation

- How to use the Verilog Simulation on workstation?
  - Step3: Key in the following command for your circuit and testbench.
    - ncverilog +access+r your_circuit.v your_circuit_tb.v

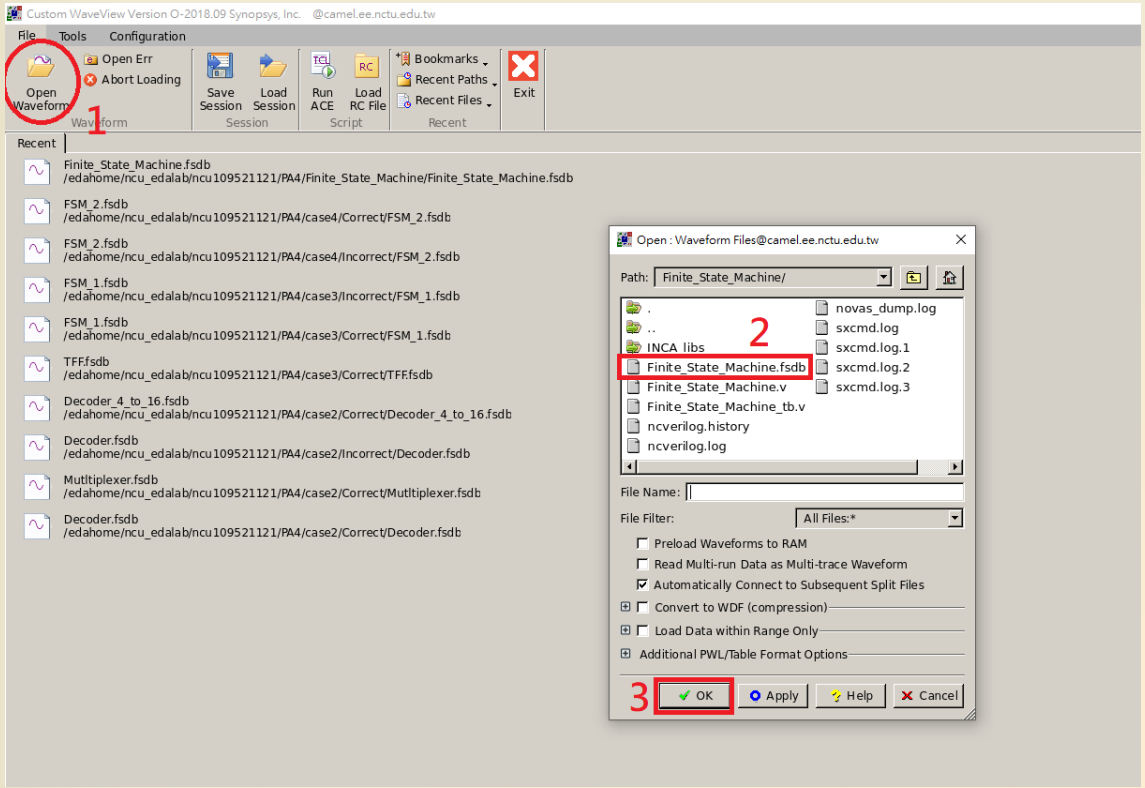
```
[109521121@eda359_forclass Half_Adder]$ ncverilog +access+r  Half_Adder.v Half_Adder_tb.v
```

  - Step4: Open wave viewer to check the waveform
    - wv&

```
[109521121@eda359_forclass Half_Adder]$ wv&
[1] 36277
```

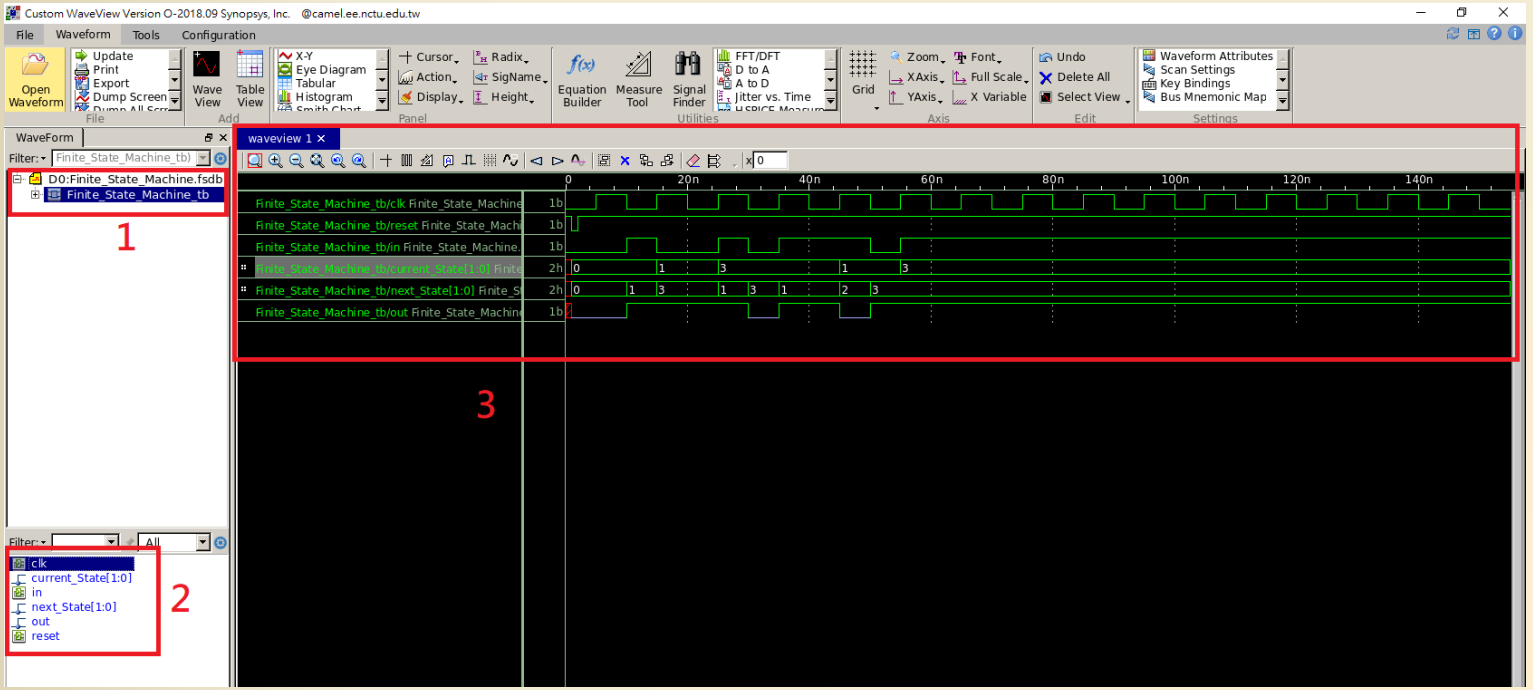
Verilog Simulation

- How to use the Verilog Simulation on workstation?
  - Step4: Open wave viewer to check the waveform
    - wv&



Verilog Simulation

- How to use the Verilog Simulation on workstation?
  - Step4: Open wave viewer to check the waveform
    - wv&



Verilog Simulation

- How to use the Verilog Simulation on workstation?
  - Step5(Option): You can write some values from Verilog for c++/c to use for automated detection.
    - https://www.cnblogs.com/oomusou/archive/2008/02/11/1066839.html

Outline

- Verilog Simulation
- Examples
  - Half Adder
  - Full Adder
  - Finite State Machine
- Cases

Half Adder

- Half Adder Correct Behavior

Carry_in	Augend	Addend	Sum	Carry_out
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Full Adder

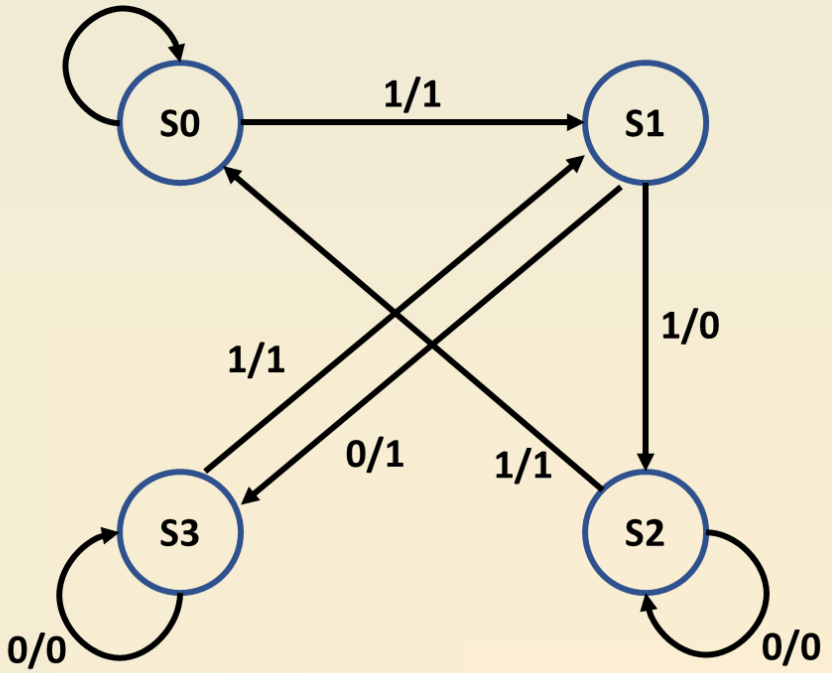
- Full Adder Correct Behavior

X	Y	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Finite State Machine

- Finite State Machine Correct Behavior

Current State	Next State		out	
	in=0	in=1	in=0	in=1
S0	S0	S1	0	1
S1	S3	S2	1	0
S2	S2	S0	0	1
S3	S3	S1	0	1



Outline

- Verilog Simulation
- Example
- Cases
  - Case 1: Comparator 4-bit
  - Case 2: Decoder
  - Case 3: FSM 1
  - Case 4: FSM 2

Comparator 4-bit

- Comparator
  - A<B ➡ Result: 100
  - A>B ➡ Result: 010
  - A=B ➡ Result: 001

A	B	Result
4'd3	4'd9	3'b100
4'd12	4'd5	3'b010
4'd7	4'd7	3'b001

Decoder

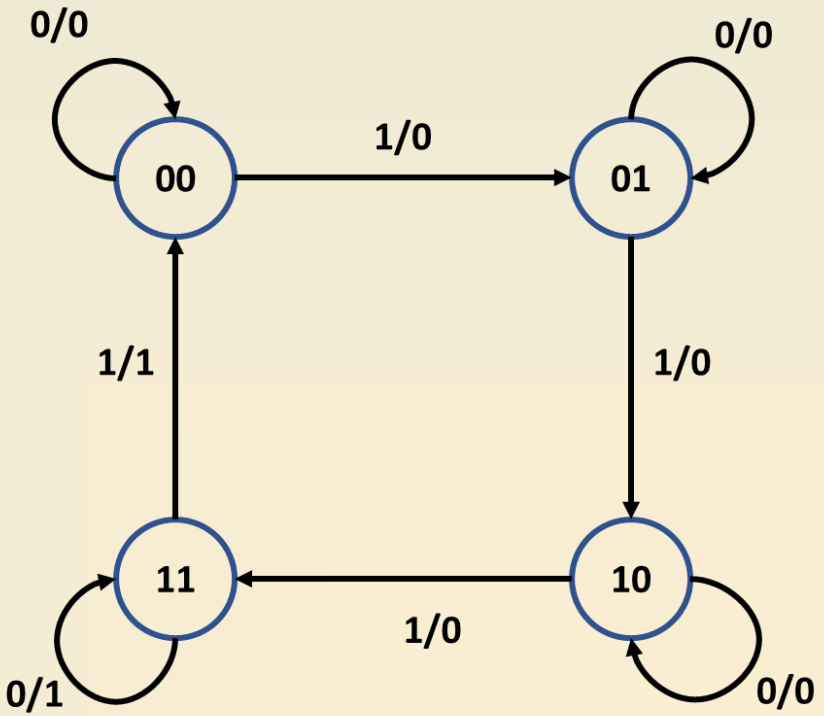
- Decoder Correct Behavior

Select				D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

FSM 1

◆FSM 1 Correct Behavior

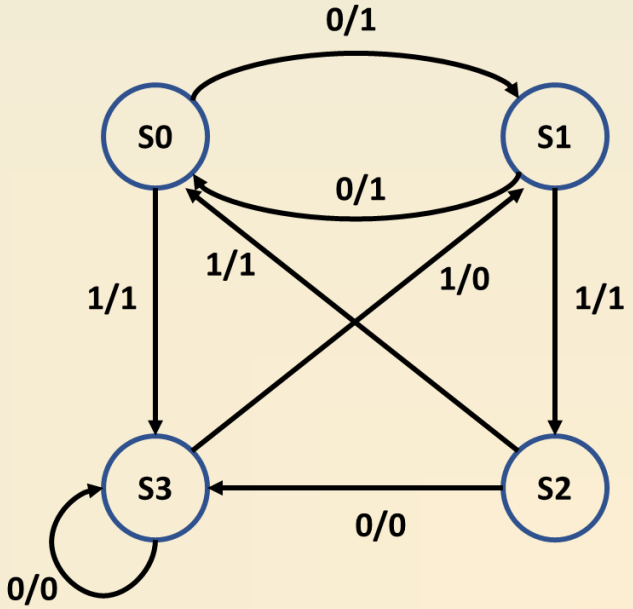
Current State		Input	Next State		Output
A	B	in	A	B	out
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	1



FSM 2

- FSM 2 Correct Behavior
  - FSM 2 can be divided into two parts, the first is normal operation mode and the second is idle mode.
    - Normal mode:

Current State	Next State		out	
	in=0	in=1	in=0	in=1
S0	S1	S3	1	1
S1	S0	S2	1	1
S2	S3	S0	0	1
S3	S3	S1	0	0



FSM 2

- FSM 2 Correct Behavior
  - FSM 2 can be divided into two parts, the first is normal operation mode and the second is idle mode.
    - Idle mode:
      - After 10 clock periods of normal mode, the current state will remain for 6 clock periods, that is, idle mode.
    - Normal mode and idle mode will rotate with each other