VLSI Testing
積體電路測試

Boolean Testing without Fault Model
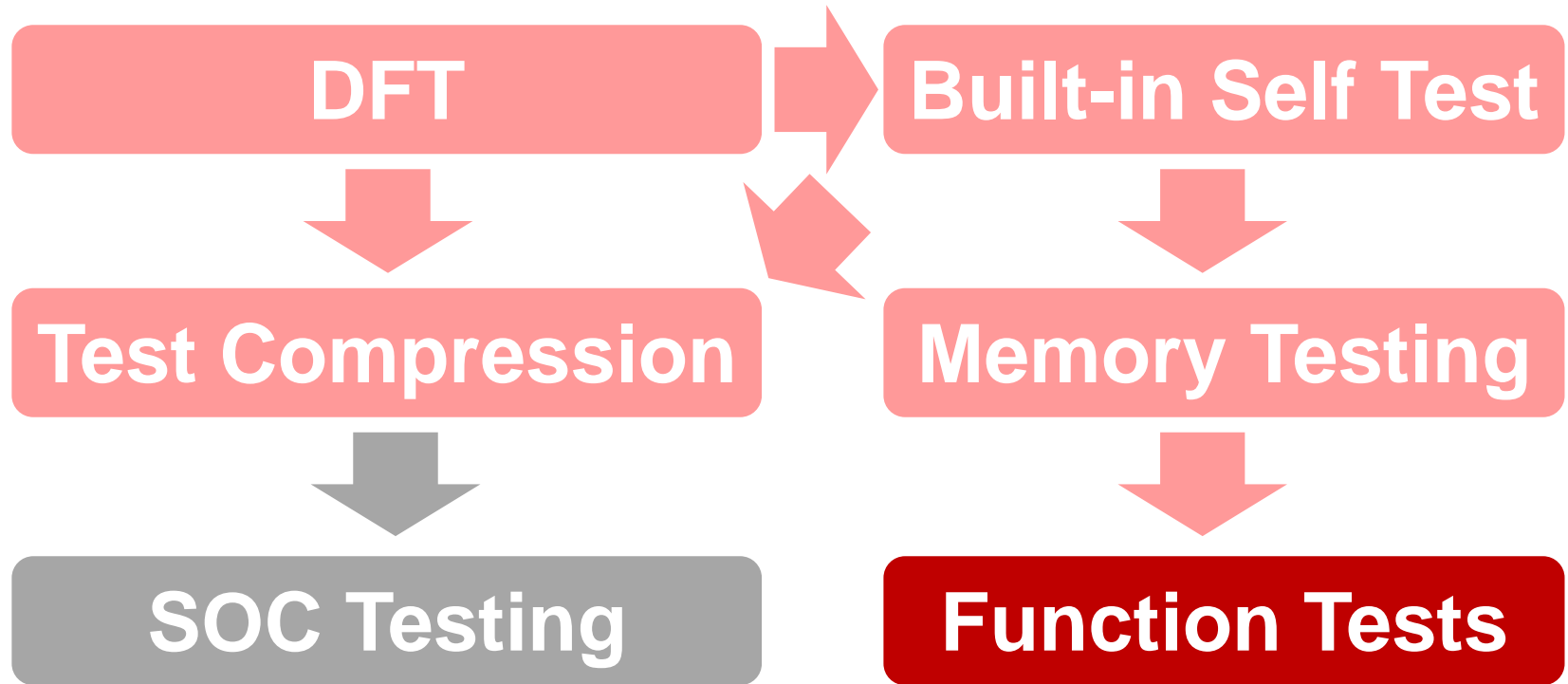
Professor James Chien-Mo Li 李建模

Lab. of Dependable Systems

Graduate Institute of Electronics Engineering

National Taiwan University

(courtesy of Prof. McCluskey, Stanford Univ.)

1

Course Roadmap (Design Topics)

DFT → Built-in Self Test

DFT → Test Compression

Test Compression → SOC Testing

Built-in Self Test → Memory Testing

Memory Testing → Function Tests

Why Am I Learning This?

- **Functional Testing is important because**
    1) Many circuits still reply on designers to generate test patterns
    2) Functional test applied at speed, important for delay defects
    3) Helps to debug design errors

When you see a worthy person, emulate him.
When you see an unworthy person, examine your inner self.
(Confucius)

3

Test without Fault Model

- **Introduction**
- **Boolean Tests without Fault Model**
  - ♦ **Toggle Test**
  - ♦ **Design Verification**
  - ♦ **Exhaustive Test**
  - ♦ **Pseudo Exhaustive Test (PET)**
- **Conclusions**
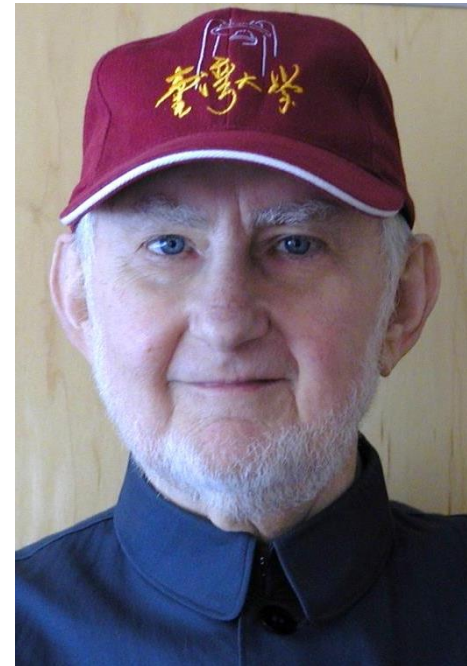
Many slides in this chapter are in memorial of Prof. McCluskey, CRC, Stanford University

Test Generation

Fault Models	Combinational Circuits (seq. ckt. w/ scan)	Sequential Circuits
No fault model	Toggle Functional Verification Exhaustive Pseudo exhaustive	Checking experiment
Single Stuck-at Fault Model	D PODEN FAN	Extended D 9-valued
Delay Fault Model	Path delay Transition delay	Launch on capture Launch on shift

Testing w/wO Fault Models (review 3.1)

Comparison	Functional testing test ckt functionality w/o fault model	Structural testing test ckt structure with fault model
Test pattern generation	☹ manual	☺ automatic
Fault coverage	☹ low	☺ high
Test speed	☺ at-speed testing test at specified circuit speed helps to defect delay faults	☹ slow speed testing exercise ckt in different ways from functional mode
Test power	☺ low power	☹ high power
Verification / silicon debug	☺ helps to debug	☹ does not help

Two Tests Complement Each Other

6

Brief History

- **1970~1990**
  - ♦ **Silicon expensive, DFT not widely used**
  - ♦ **ATPG not mature**
  - ♦ **Design simple**
  - ♦ **Functional verification tests without fault model popular**
- **1990~2010**
  - ♦ **Silicon not so expensive, DFT became standard**
  - ♦ **ATPG was mature**
  - ♦ **Design very complex, manual test generation infeasible**
  - ♦ **Structural tests with fault model popular**
- **2010~**
  - ♦ **Delay defects requires at-speed testing**
  - ♦ **Test power became serious problem**
  - ♦ **Functional verification test become popular again**

Func. and Structural Tests Both Needed

7

Test without Fault Model

- **Introduction**
- **Boolean Tests without Fault Model**
  - ♦ **<u>Toggle Test</u>**
  - ♦ **Design Verification Test**
  - ♦ Exhaustive Test
  - ♦ Pseudo Exhaustive Test (PET)
- Conclusions

McCluskey and his collection of hats

8

Toggle Coverage (DEF-1)

$$toggle\ coverage = \frac{\sum\limits_{all\ nodes\ i} \#\ of\ different\ \underline{values}\ of\ node\ i}{2 \times total\ \#\ of\ nodes}$$

 # = numbers
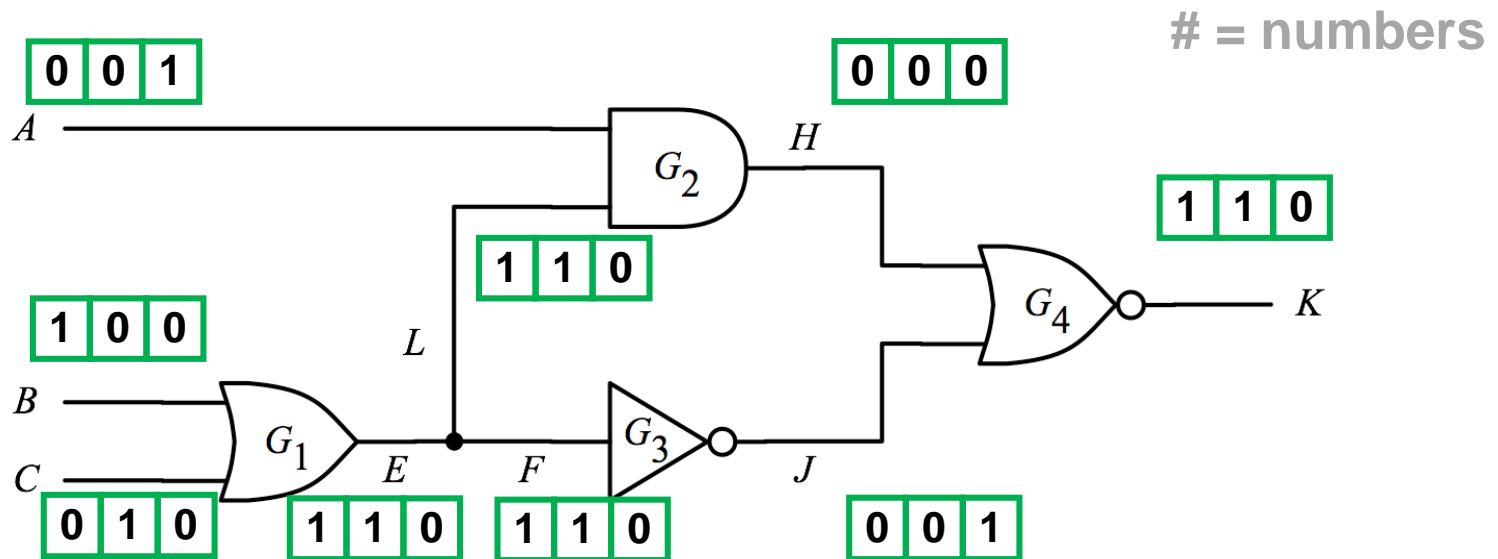
Toggle Coverage = 17/18 = 94%

Toggle Coverage (DEF-2)

$$toggle\ coverage = \frac{\sum_{all\ nodes\ i} \#\ of\ different\ \underline{transitions}\ of\ node\ i}{2 \times total\ \#\ of\ nodes}$$
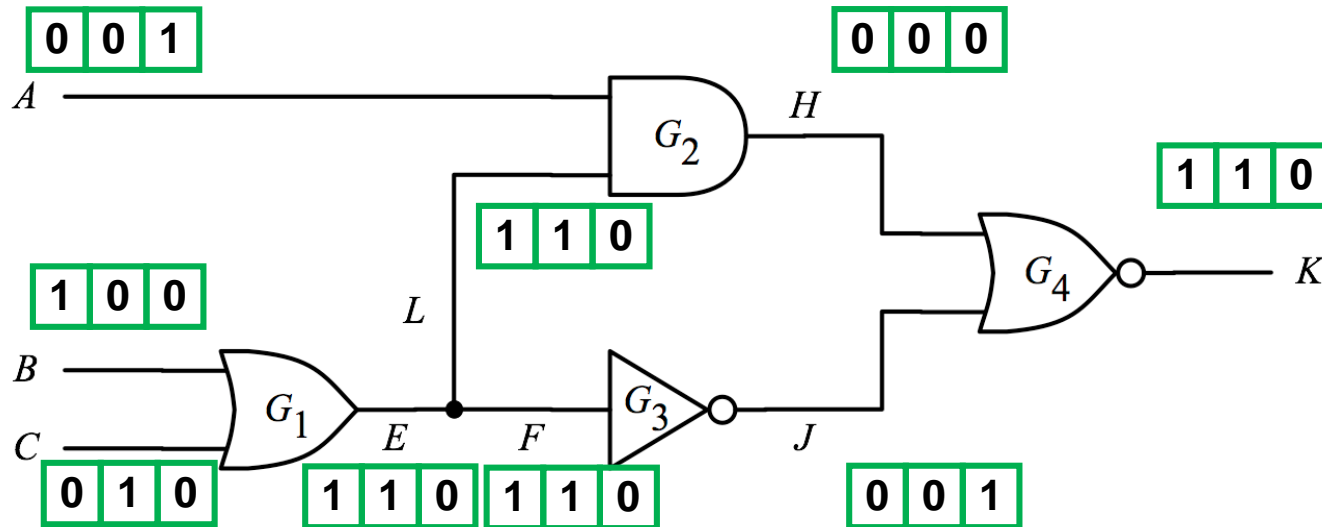
Toggle Coverage = 9/18 =50%
DEF-2 more stringent

Toggle Test: Pros and Cons

☺ **Advantage: Toggle coverage is easy to obtain**
  ♦ **Logic simulation only**, NO fault simulation
  ♦ **Short test length**
☹ **Disadvantage: Toggle coverage is very optimistic**
  ♦ **Fault activation only, NO fault propagation**
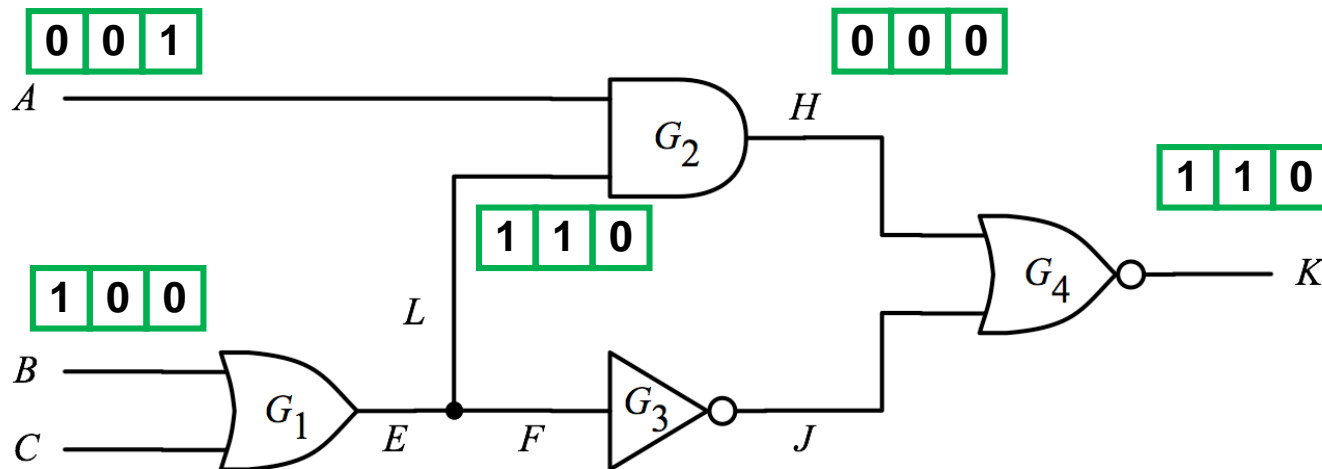
Quiz

Q: Apply 3 patterns to this circuit of 7 nodes.
   What is toggle coverage ? Use DEF-2 (transition).

A:
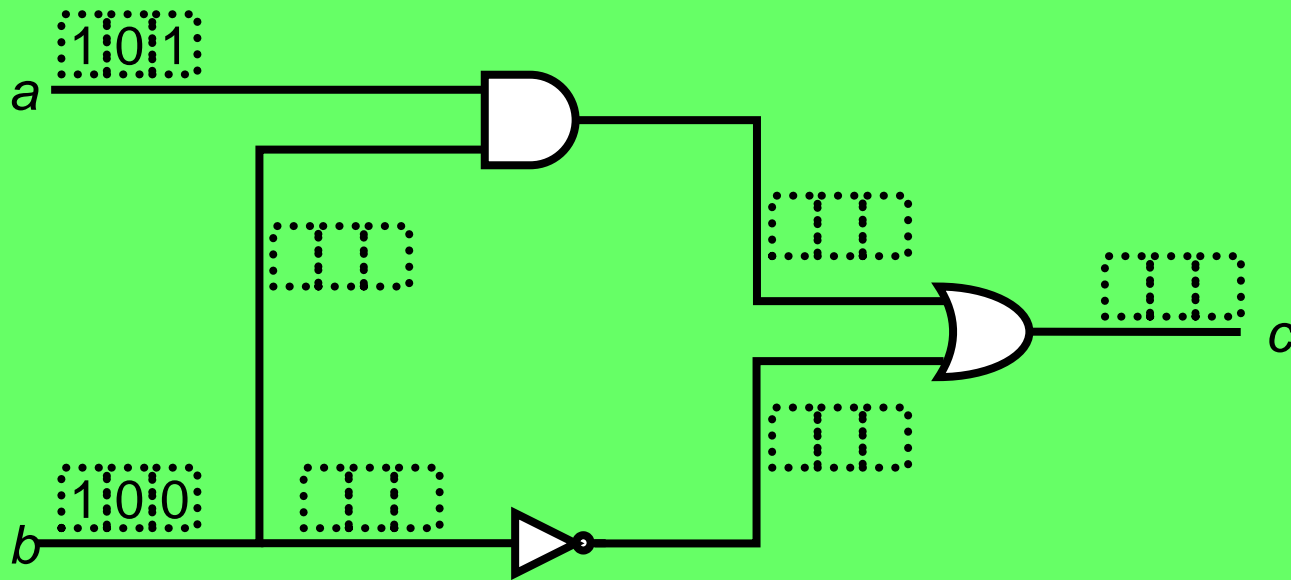
Test without Fault Model

- **Introduction**
- **Boolean Tests without Fault Model**
  - ♦ **Toggle Test**
  - ♦ **Design Verification Test**
  - ♦ Exhaustive Test
  - ♦ Pseudo Exhaustive Test (PET)
- Conclusions

McCluskey and his collection of hats

Design Verification

- **Purpose**
  - **Establish a design correctly implement a behavior specification**
- **Created for design verification**
  - **May not good enough for detecting defects**
- **Example: Dual 4-to-1 MUX**
  - **XW=00 selects A1 A2, XW=01 selects B1 B2   etc**

14
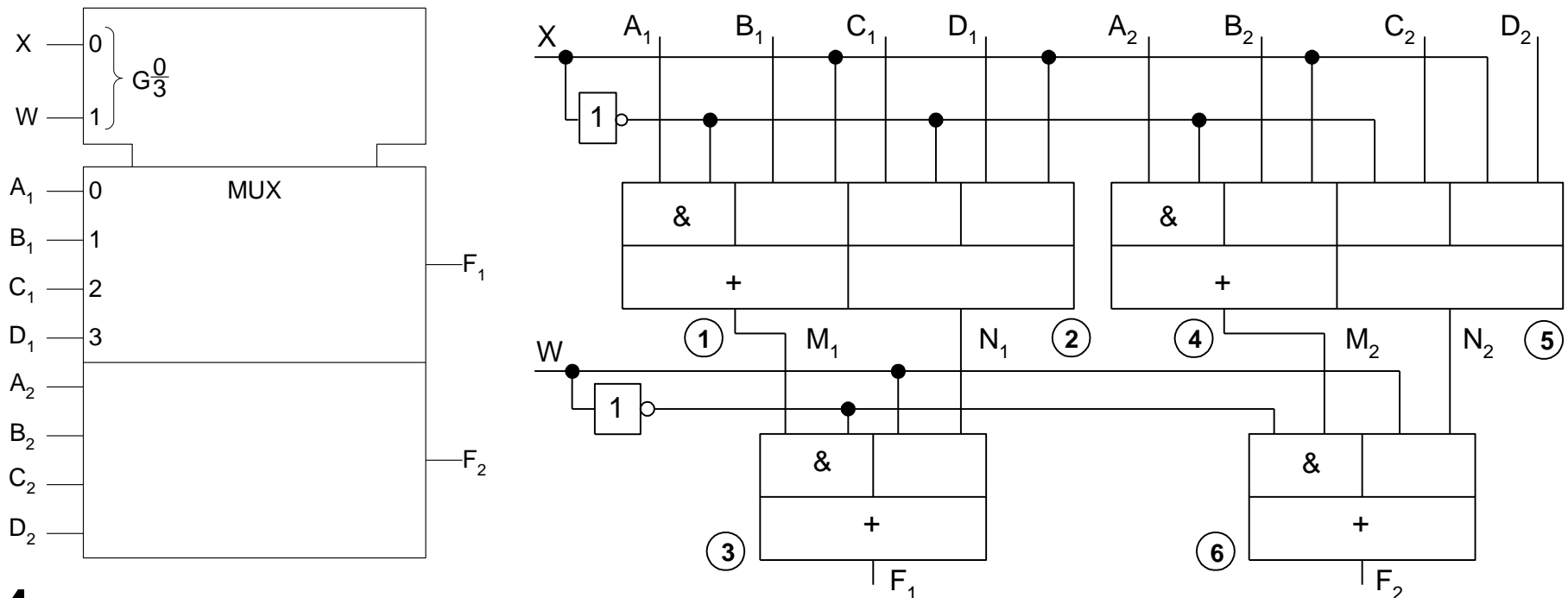
Low Fault Coverage Problem

- **Design verification patterns**
  - **Test length =8**
  - **100% toggle coverage**
  - **Only 68% SSF coverage**

- **ATPG test patterns**
  - **Same test length**
  - **100% toggle coverage**
  - **100% SSF coverage**

W X	A_1	B_1	C_1	D_1	A_2	B_2	C_2	D_2	F_1	F_2
0 0	0	0	0	0	0	0	0	0	0	0
0 1	0	0	0	0	0	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	0	0	0
0 0	1	0	0	0	1	0	0	0	1	1
0 1	0	1	0	0	0	1	0	0	1	1
1 0	0	0	1	0	0	0	1	0	1	1
1 1	0	0	0	1	0	0	0	1	1	1

W X	A_1	B_1	C_1	D_1	A_2	B_2	C_2	D_2	F_1	F_2
0 0	0	1	1	d	0	1	1	d	0	0
0 1	1	0	d	1	1	0	d	1	0	0
1 0	1	d	0	1	1	d	0	1	0	0
1 1	d	1	1	0	d	1	1	0	0	0
0 0	1	d	d	d	1	d	d	d	1	1
0 1	d	1	d	d	d	1	d	d	1	1
1 0	d	d	1	d	d	d	1	d	1	1
1 1	d	d	d	1	d	d	d	1	1	1

d = don't care

15

Long Test Length Problem

- **Alternative design verification test set**
  - ♦ **20 test patterns , 100% SSF coverage**

W X	A_1	B_1	C_1	D_1	A_2	B_2	C_2	D_2	F_1	F_2
0 0	0	0	0	0	0	0	0	0	0	0
0 1	0	0	0	0	0	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	0	0	0
0 0	1	0	0	0	1	0	0	0	1	1
0 1	1	0	0	0	1	0	0	0	0	0
1 0	1	0	0	0	1	0	0	0	0	0
1 1	1	0	0	0	1	0	0	0	0	0
0 0	0	1	0	0	0	1	0	0	0	0
0 1	0	1	0	0	0	1	0	0	1	1
1 0	0	1	0	0	0	1	0	0	0	0
1 1	0	1	0	0	0	1	0	0	0	0
0 0	0	0	1	0	0	0	1	0	0	0
0 1	0	0	1	0	0	0	1	0	0	0
1 0	0	0	1	0	0	0	1	0	1	1
1 1	0	0	1	0	0	0	1	0	0	0
0 0	0	0	0	1	0	0	0	1	0	0
0 1	0	0	0	1	0	0	0	1	0	0
1 0	0	0	0	1	0	0	0	1	0	0
1 1	0	0	0	1	0	0	0	1	1	1

Summary

- **Functional test without fault models**
  - ☺ **at-speed testing**
  - ☺ **low power**
  - ☺ **helps to debug**
  - ☹ **manually generated**
  - ☹ **Low fault coverage**
  - ☹ **Long test length**

- **Toggle test**
  - ♦ **Easy to evaluate**
  - ♦ **Two definitions: value, transition**
- **Design verification test**
  - ♦ **Long test length but low FC**

17