Basic Test Methodology for Logic Designers

Design for Testability

Created: Yasushi Ogawa

Updated: Hai Pham, Thanh Le

Outline Design For Testability (DFT)

1. Importance of DFT

2. DFT for Logic Parts

3. DFT for Embedded Memory

Outline Design For Testability (DFT)

1. Importance of DFT

(a) Purpose of Test

(b) Test Challenge and Methodology

(c) Defect and Fault Coverage

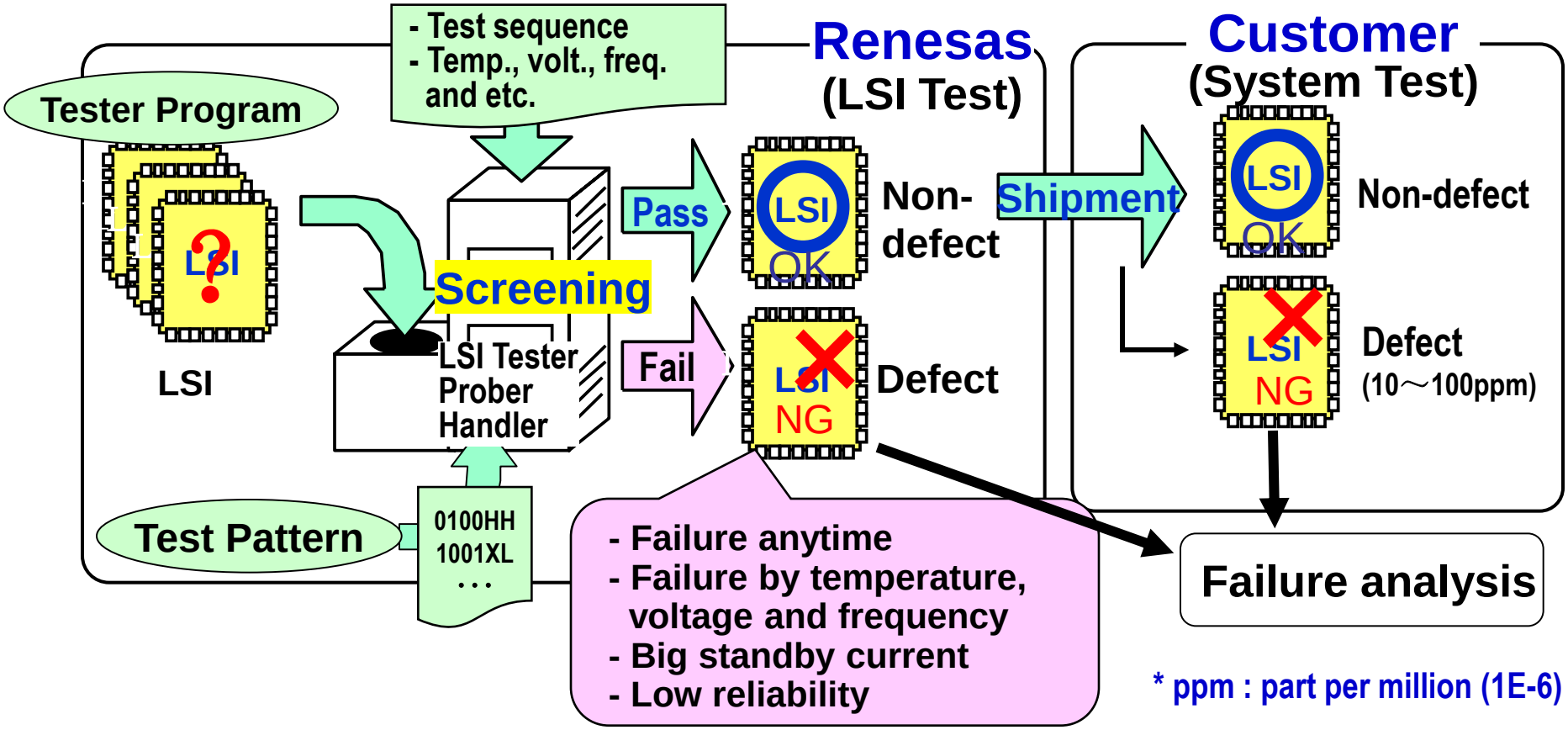
(d) Required Quality and Test Level

2. DFT for Logic Parts

3. DFT for Embedded Memory

Purpose of Test

- Screening of manufacturing defects
- i.e. "*last defense for LSI quality assurance*"



Purpose of Test

Test cost is increasing year by year

Purpose of test is as follows

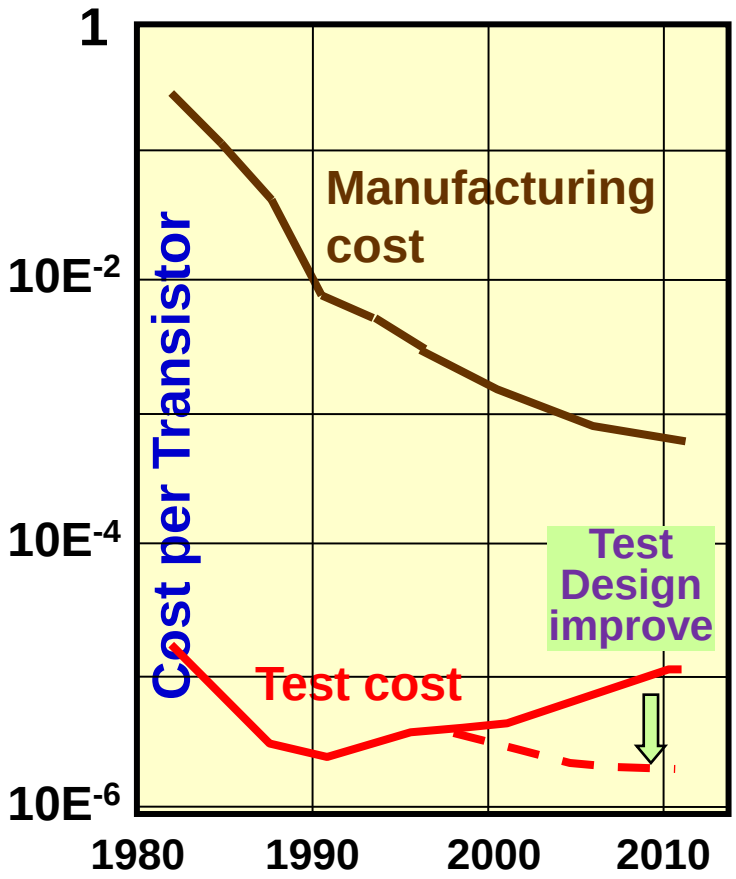
- Maximum quality assurance of LSI
- Adequate test cost

Test Cost (*tester-related cost*)

- Equipment (*tester, prober, handler*)
- Operation (*labor cost*)
  - ☞ increase of ratio to manufacturing cost

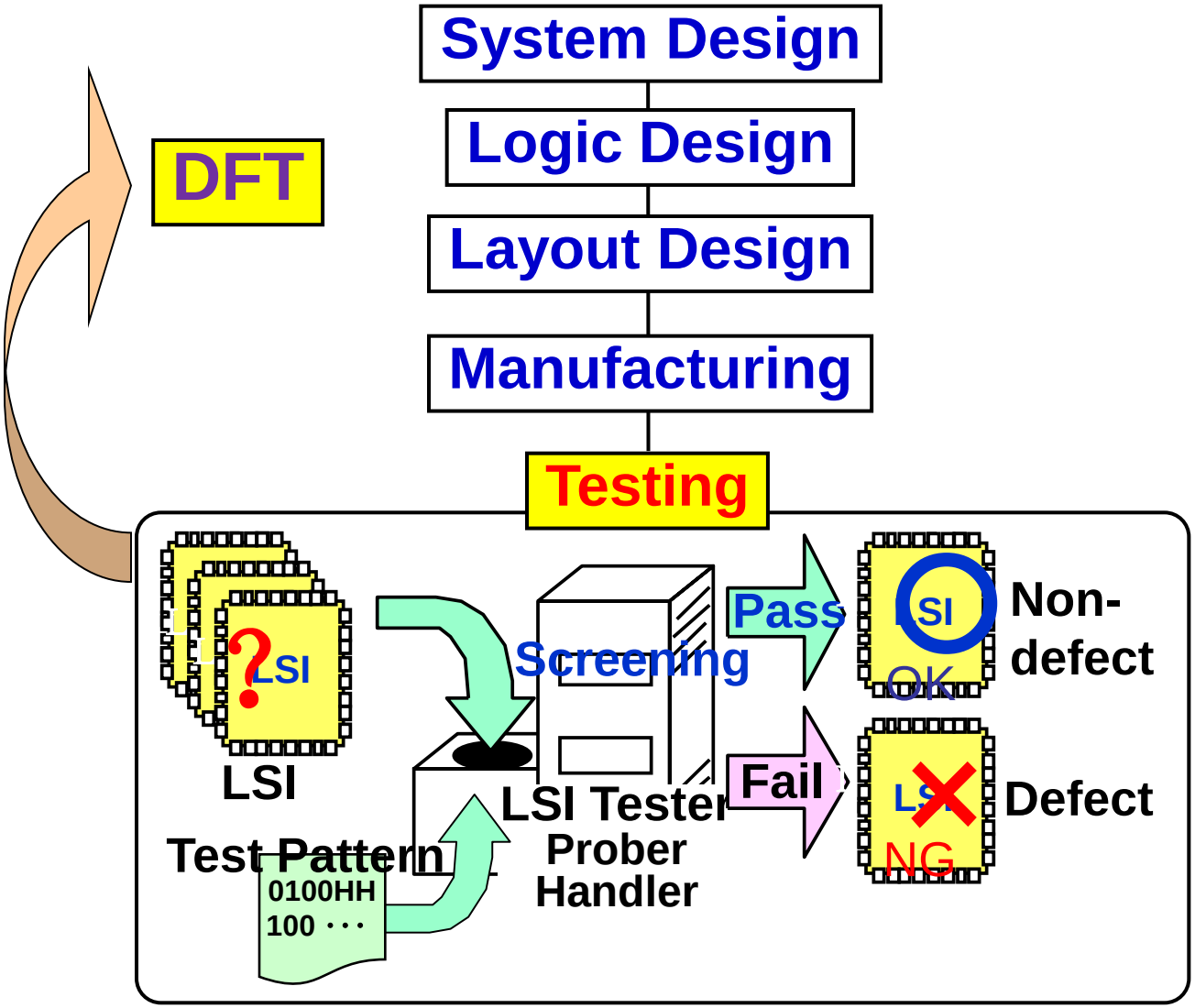
Test Cost (*cost other than tester*)

- Test design cost (*labor cost*)
- Test circuit cost (*chip area*)
- Failure analysis cost (*labor test*)



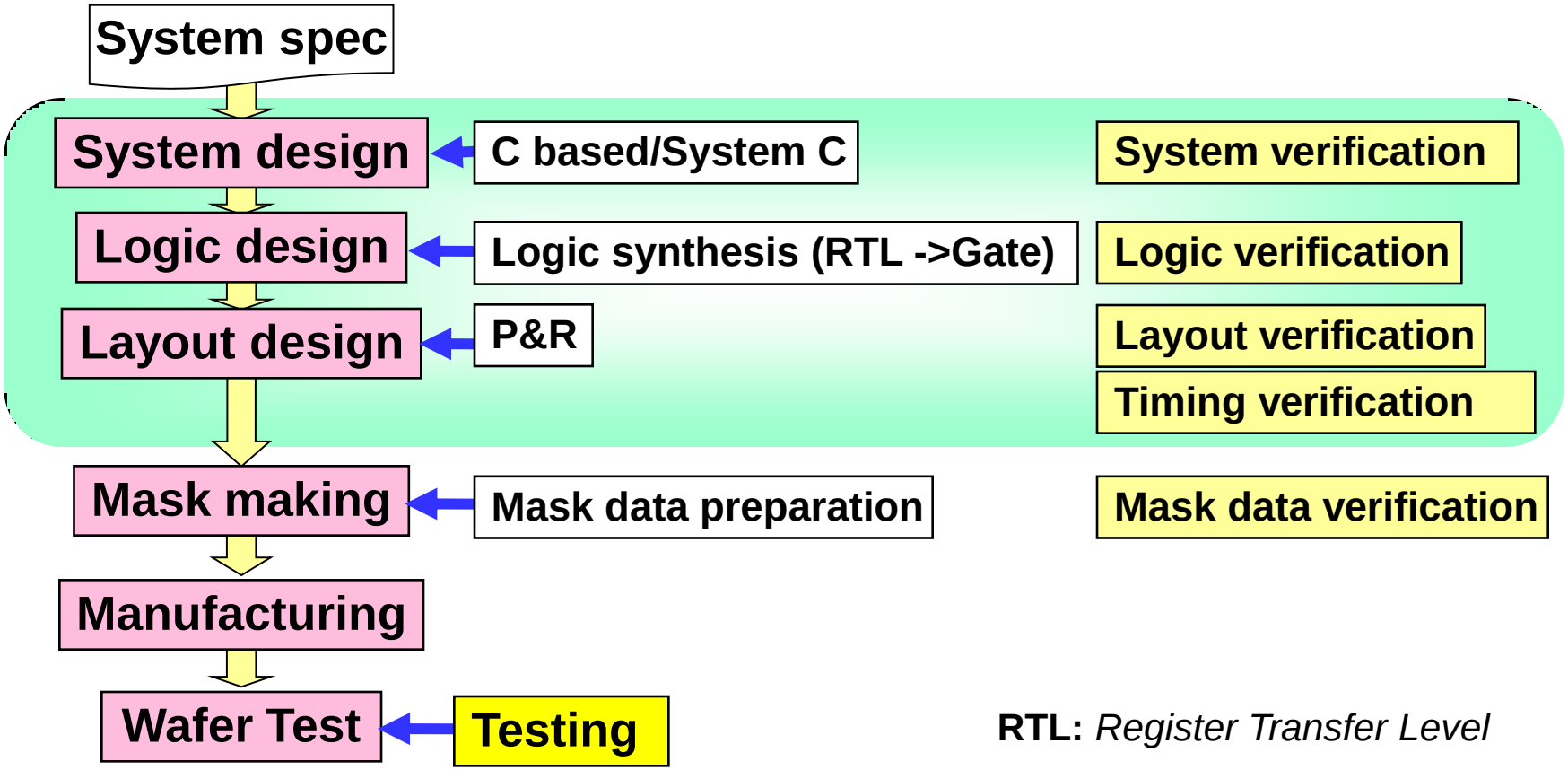
(Source: ITRS2002 Public Conference)

DFT : Design For Testability



LSI Design Flow

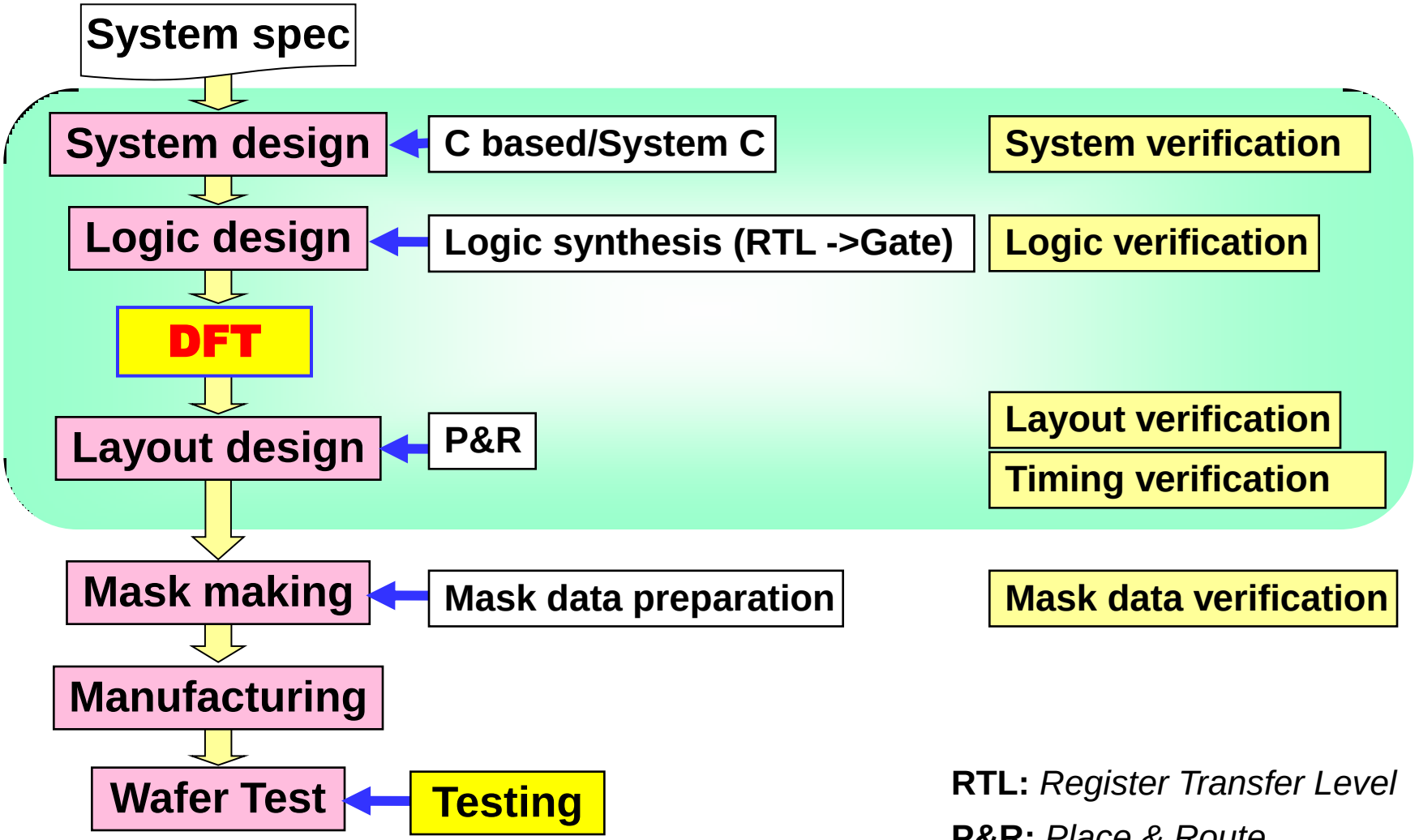
QUIZ: To which step DFT is applied?



RTL: *Register Transfer Level*

P&R: *Place & Route*

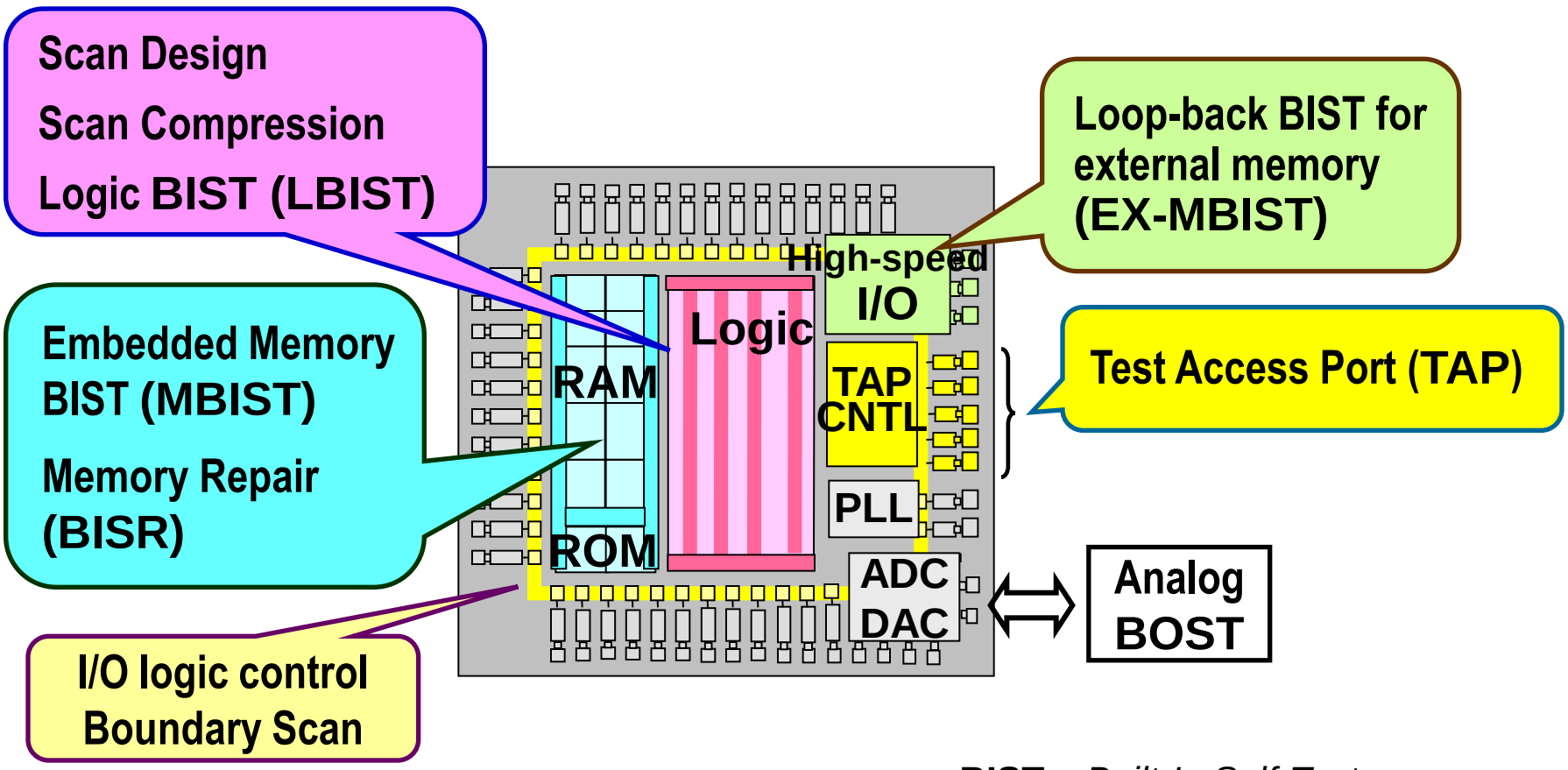
LSI Design Flow



RTL: *Register Transfer Level*

P&R: *Place & Route*

Application Example of DFT



BIST: *Built-In Self-Test*
BOST: *Built-Out Self-Test*
BISR: *Built-In Self-Repair*

What is DFT?

DFT is a design technique that adds test circuits inside the LSI hardware. These added circuits permit to easily apply tests on the designed LSI after manufacturing.

DFT consists of:

- Creating test circuits
- Inserting test circuits into LSI hardware
- Preparing test data for analyzing failure due to physical defect

Purpose of DFT:

- Guarantee the LSI product quality
- Improve manufacture process
- Optimize test cost

Target of DFT:

- Minimize area overhead
- Increase test/fault coverage
- Minimize the test time
- Minimize the test cost

Outline Design For Testability (DFT)

1. Importance of DFT

(a) Purpose of Test

(b) Test Challenge and Methodology

(c) Defect and Fault Coverage

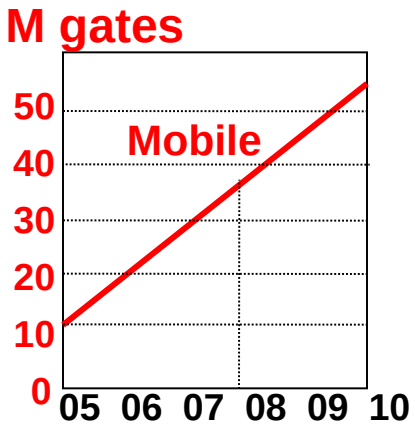
(d) Required Quality and Test Level

2. DFT for Logic Parts

3. DFT for Embedded Memory

Impact of SOC Design Trend

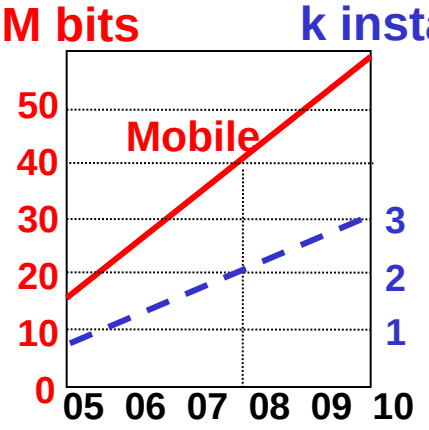
(1) Logic size



> 30% increase a year
-> test time increases much

LBIST or Compression Test

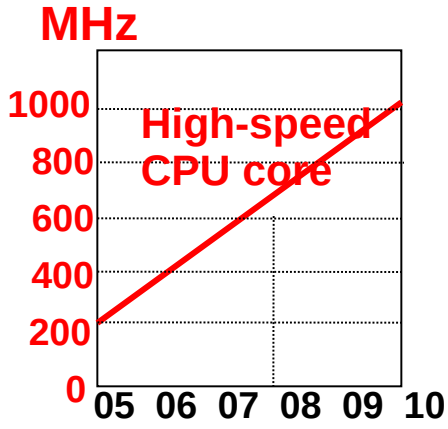
(2) Memory



> 30% increase a year
-> test time increases
(proportion to size)

Concurrent measurement using MBIST

(3) Frequency



> 30% increase a year
-> High-speed tester

Enable to lower external clock freq. using PLL (around 30MHz)

Apply DFT:

- Reduces an increase in test time;
- Reduce high requirement of tester capacity.

Test Methodology

	Functional Test	Structural Test
Concept	Actual LSI operation = Pay attention to Function Check whether LSI functions in the same conditions to the actual condition in use	Pay attention to defects during manufacturing and resulting faults Cover all faults by modeling faults completely
Pass/Fail Decision	Decide by operating functionally or not	Decide the existing of defects during manufacturing or not
Coverage Measure	Stuck-at fault + critical path check by manual	Evaluating coverage quantitatively for each fault models by tools
Total coverage capability	Difficult to evaluate quantitatively (no fault modeling ➔ no definition of coverage)	Easy to evaluate quantitatively (Summary for each fault models)
Test quality	There is no scientific improvement because there is no quantification	Evaluate scientifically by quantification (*important to choose models*)

Outline Design For Test (DFT)

1. Importance of DFT

(a) Purpose of Test

(b) Test Challenge and Methodology

(c) Defect and Fault Coverage

(d) Required Quality and Test Level

2. DFT for Logic Parts

3. DFT for Embedded Memory

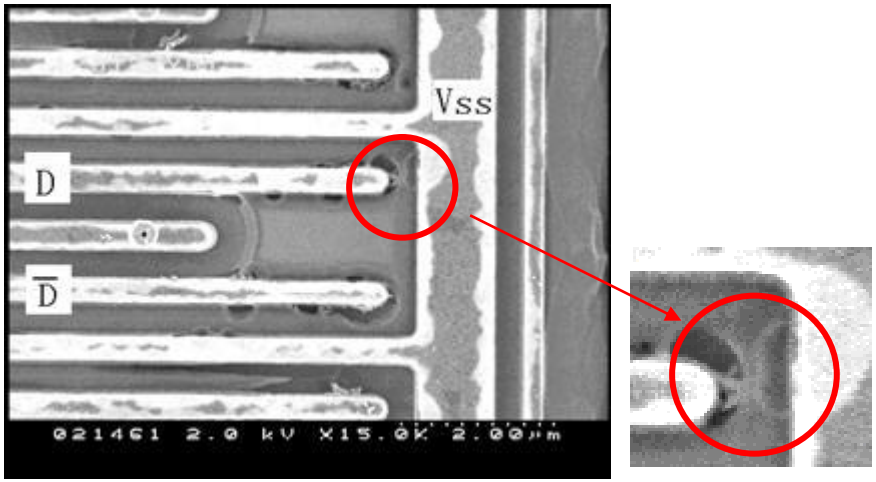
Physical Defects

Physical defects may happen during manufacturing

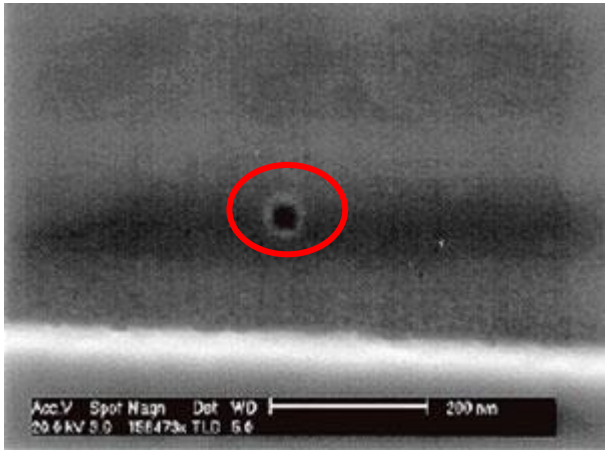
- Abnormality in MOS Transistor
  - ☞ Abnormality in Gate Oxide
  - ☞ Shortage in MOS Transistor
  - ☞ Abnormality in wire width, space and thickness
  - ☞ Contamination due to foreign particle
- Shortage in wiring
  - ☞ Etching residuum
  - ☞ Bridge due to foreign particle
  - ☞ Abnormal wider width and thicker thickness
- Disconnection in wiring
  - ☞ Non-conductive via and higher resistance of via
  - ☞ Abnormal narrower width and thinner thickness
  - ☞ Half disconnection due to foreign particle and void

Physical Defects

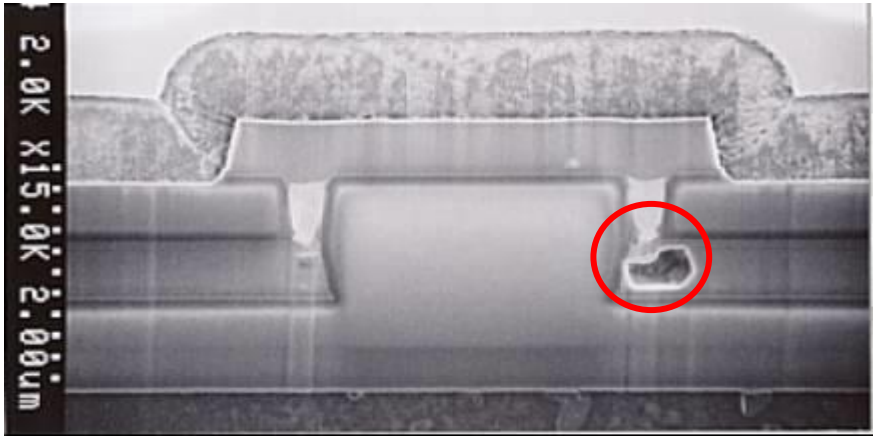
There are a variety of LSI defects in manufacture phase



Shortage due to foreign particle



Gate oxide pinhole



Disconnection due to EM (Electro-migration)

Fault Coverage

A Fault is a representation of a defect reflecting a physical condition that causes a circuit to fail to perform in a required manner.

Fault Coverage: the percentage of total faults for which test patterns have been generated

$$Fault\ Coverage = \frac{Number\ of\ Detected\ Faults}{Total\ Number\ of\ Faults\ in\ the\ Circuit} \times 100$$

Fault Coverage is also called **Failure Detection Rate** and can be estimated by using **Fault Models**

Outline Design For Test (DFT)

1. Importance of DFT

(a) Purpose of Test

(b) Test Challenge and Methodology

(c) Defect and Fault Coverage

(d) Required Quality and Test Level

2. DFT for Logic Parts

3. DFT for Embedded Memory

Required Quality and Test Level

Required quality (quality level of products)

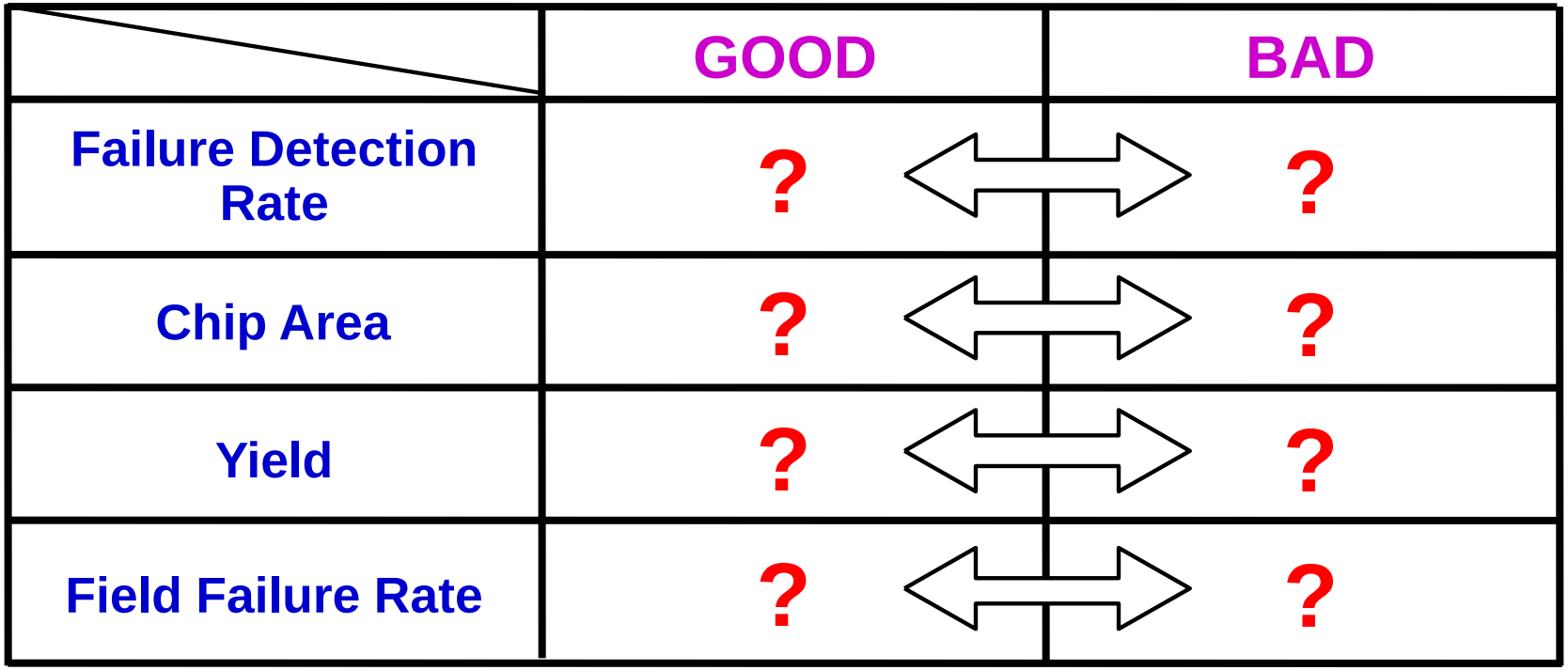
- Zero defect
- Guarantee of operation under warranty condition

Test level

- Test items
- Test conditions
  (*temperature, voltage, frequency/timing, etc.*)
- Test model (*measure representing test sufficiency*)
  and target fault coverage (*quantitative evaluation of test sufficiency*)

Relation between Field Failure Rate and Detection Rate, Area and Yield

QUIZ: Fill the table below by using **High** or **Low** or **Large** or **Small**



Failure Detection Rate: *it is also called "**Fault Coverage**"*

Yield: *ratio of the number of good LSIs to the number of manufactured LSIs*

Field Failure Rate: *ratio of LSIs returned from customers to shipped LSIs, it is also called "**Reject Rate**"*

Relation between Field Failure Rate and Detection Rate, Area and Yield

	GOOD	BAD
Failure Detection Rate	High	Low
Chip Area	Small	Large
Yield	High	Low
Field Failure Rate	Low	High

Failure Detection Rate: *it is also called "**Fault Coverage**"*

Yield: *ratio of the number of good LSIs to the number of manufactured LSIs*

Field Failure Rate: *ratio of LSIs returned from customers to shipped LSIs, it is also called "**Reject Rate**"*

Target Quality of Renesas Products

Standardization of common quality level requested by Consumer (RS-4001)

Quality Classification		Guaranteed Lifetime	Main products
High Reliability	Q1A	20 years	Automobile parts (*Engine controller*, *etc.*) General traffic equipment
	Q1B	10 years	Car electronics (*accessories*: *genuine goods*), etc.
Industry	Q2	10 years	Car electronics, Factory Automation equipment, etc.
Consumer	Q3	10 years	PC, Consumer electronics, Mobile equipment
Custom	QX	Determine individually for each product	Game instruments, Ultra high reliable equipment

Yield and Field Failure Rate

QUIZ: Find the word corresponding to XX , YY , ZZ ?

ASA method

A formula to model the relation between yield and fault coverage over field failure rate

Field failure rate

$$RR(f) = \frac{(1 - f) * (1 - y) * \exp(-(n_0 - 1) * f))}{y + (1 - f) * (1 - y) * \exp(-(n_0 - 1) * f))}$$

Field failure rate: RR (Reject rate)
Yield: y
Fault Coverage : f
N_0 is a mean failure counts per 1 defect
(Example values in the paper 8, 1.94, 4-5)

Example1

Lower XX causes increase of YY

Variation of RR depending on yield fluctuation while fault coverage remains 95%.

Yield(%)	90	80	70	60	50	40	30	20	10
Fault coverage(%)	95	95	95	95	95	95	95	95	95
RR(ppm)	40	89	153	238	258	536	834	1429	3209

ppm : $1E^{-6}$

Assumption $n_0 = 6.2$

Example2

Need to improve ZZ

Variation of fault coverage due to yield fluctuation to maintain RR=50ppm

Yield(%)	90	80	70	60	50	40	30	20	10
Fault coverage(%)	94.0	96.9	98.1	98.7	99.1	99.4	99.6	99.7	99.9
RR(ppm)	50	50	50	50	50	50	50	50	50

Assumption $n_0 = 6.2$

Source:V.D.Agrawal, S.C.Seth, P.Agrawal,"Fault Coverage Requirements in Production Testing of LSI Circuits" IEEE Journal of Solid State Circuits Vol.SC-17, pp57-61, 1982

Yield and Field Failure Rate

ASA method

A formula to model the relation between yield and fault coverage over field failure rate

Field failure rate

$$RR(f) = \frac{(1 - f) * (1 - y) * \exp(-(n_0 - 1) * f))}{y + (1 - f) * (1 - y) * \exp(-(n_0 - 1) * f))}$$

Field failure rate: RR (Reject rate)
Yield: y
Fault Coverage : f
N_0 is a mean failure counts per 1 defect
(Example values in the paper 8, 1.94, 4-5)

Example1

Lower Yield causes increase of Reject Rate

Variation of RR depending on yield fluctuation while fault coverage remains 95%.

Yield(%)	90	80	70	60	50	40	30	20	10
Fault coverage(%)	95	95	95	95	95	95	95	95	95
RR(ppm)	40	89	153	238	258	536	834	1429	3209

ppm : $1E^{-6}$

Assumption $n_0 = 6.2$

Example2

Need to improve ZZ

Variation of fault coverage due to yield fluctuation to maintain RR=50ppm

Yield(%)	90	80	70	60	50	40	30	20	10
Fault coverage(%)	94.0	96.9	98.1	98.7	99.1	99.4	99.6	99.7	99.9
RR(ppm)	50	50	50	50	50	50	50	50	50

Assumption $n_0 = 6.2$

Source:V.D.Agrawal, S.C.Seth, P.Agrawal,"Fault Coverage Requirements in Production Testing of LSI Circuits" IEEE Journal of Solid State Circuits Vol.SC-17, pp57-61, 1982

Yield and Field Failure Rate

ASA method

A formula to model the relation between yield and fault coverage over field failure rate

Field failure rate

$$RR(f) = \frac{(1 - f) * (1 - y) * \exp(-(n_0 - 1) * f))}{y + (1 - f) * (1 - y) * \exp(-(n_0 - 1) * f))}$$

Field failure rate: **RR** (Reject rate)
Yield: **y**
Fault Coverage : **f**
N_0 is a mean failure counts per 1 defect
(Example values in the paper 8, 1.94, 4-5)

Example1

Lower **Yield** causes increase of **Reject Rate**

Variation of RR depending on yield fluctuation while fault coverage remains 95%.

Yield(%)	90	80	70	60	50	40	30	20	10
Fault coverage(%)	95	95	95	95	95	95	95	95	95
RR(ppm)	40	89	153	238	258	536	834	1429	3209

ppm : $1E^{-6}$

Assumption $n_0 = 6.2$

Example2

Need to improve **Fault Coverage**

Variation of fault coverage due to yield fluctuation to maintain RR=50ppm

Yield(%)	90	80	70	60	50	40	30	20	10
Fault coverage(%)	94.0	96.9	98.1	98.7	99.1	99.4	99.6	99.7	99.9
RR(ppm)	50	50	50	50	50	50	50	50	50

Assumption $n_0 = 6.2$

Source:V.D.Agrawal, S.C.Seth, P.Agrawal,"Fault Coverage Requirements in Production Testing of LSI Circuits" IEEE Journal of Solid State Circuits Vol.SC-17, pp57-61, 1982

Importance of DFT - Summary

- Purpose and target of Test
  - ☞ Guarantee of quality ⇨ Classification of Pass and Fail
  - ☞ Improvement of quality ⇨ Time-To-Market, Time-To-Volume
  - ☞ Optimization of cost ⇨ necessary and sufficient (minimum) test
  - ☞ Based on yield ⇨ Balancing of test quality and cost
  - ☞ Data gathering and storage to make quantitative standard
  - ☞ The relations among Field Failure Rate, coverage, area and yield
- Evolve from Functional Test to Structural Test

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

(a) Purpose of SCAN Design

(b) Outline of SCAN Design

(c) Logic Fault Model

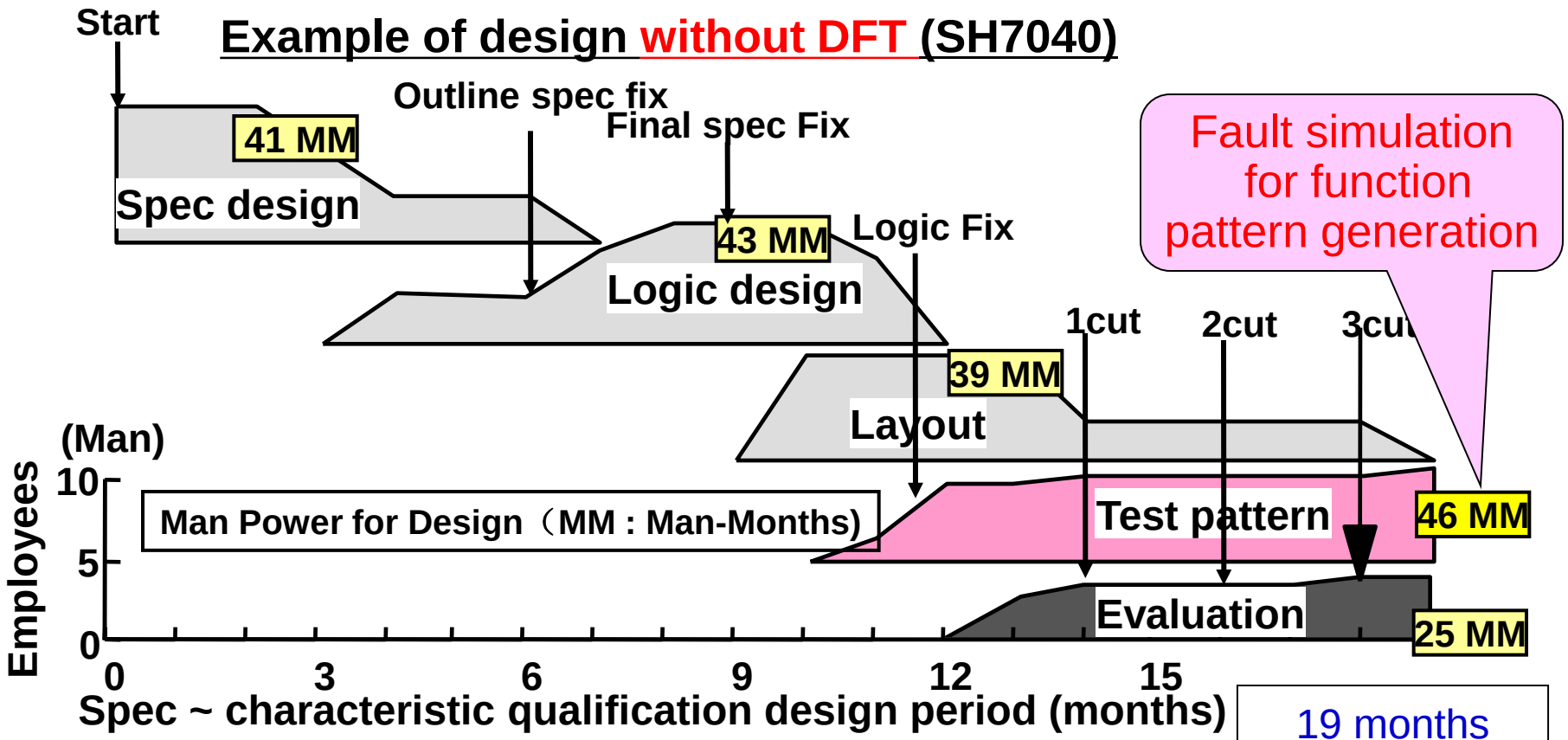
(d) Applied SCAN Design (LBIST etc.)

(e) IDDQ Test

3. DFT for Embedded Memory

Purpose of SCAN Design - Man Power (1)

Resource reduction for test design
(A limitation of function test pattern generation)



19 months by qualification

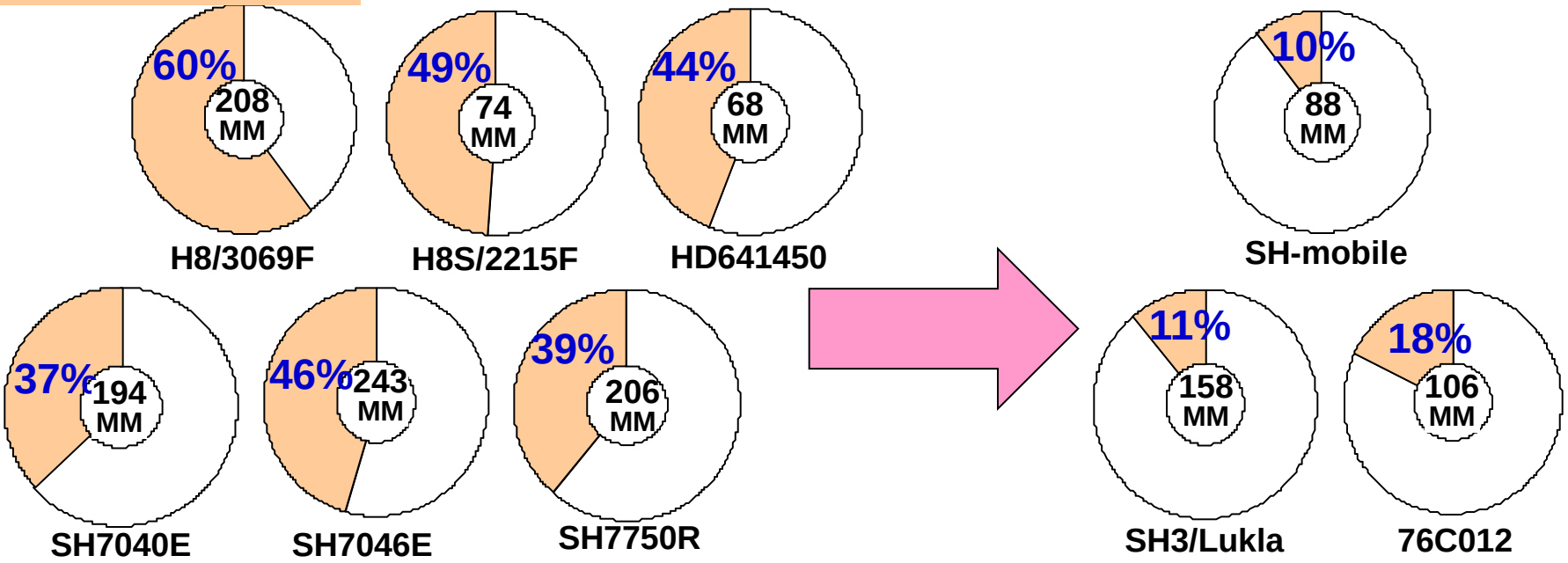
Need to promote a drastic automation

Purpose of SCAN Design - Man Power (2)

Resource reduction for test design
(A limitation of function test pattern generation)

Test & qualification resources

MM=Man-Months



Products without DFT

Products with DFT

Purpose of SCAN Design - Quality

A limitation of test quality evaluation method for function test base design

Test type	Function Test	Scan Test
Fault Model (quality)	Only stuck-at fault	Stuck-at fault, Transition fault, Short fault, Path delay fault
Test Coverage	No increase	Continuous increase as additional model
Pattern Generation	Manual generation (Depends on designers' skill and experience)	Automatic generation by ATPG (An additional model may be supported by ATPG if the model is adopted)

ATPG: Automatic Test Pattern Generation Tool

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

(a) Purpose of SCAN Design

(b) Outline of SCAN Design

(c) Logic Fault Model

(d) Applied SCAN Design (LBIST etc.)

(e) IDDQ Test

3. DFT for Embedded Memory

Outline of SCAN Design

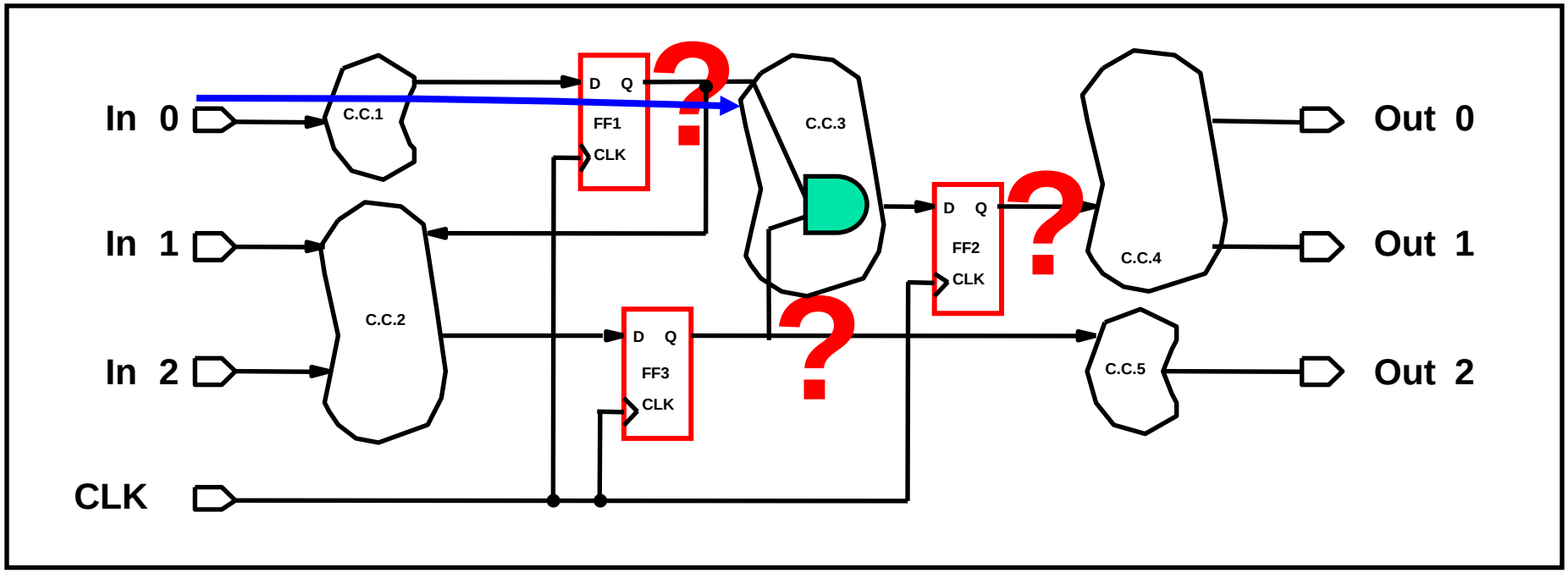
Basic concept

- A difficulty for automatic test pattern generation
  - ☞ Need to manage internal states
- "Removal of internal states" is the solution
  - ☞ Treat FFs with internal states as virtual input-output ports
  - ☞ Changing the entire logic circuit to a combination circuit without states
  - ☞ Easy to generate test patterns automatically by DA (Design Automation)
- To treat FFs as virtual input-output ports
  - ☞ Connect all FFs as if "shift registers" besides usual logic
  - ☞ Use input-output ports of FFs to control and observe data
  - ☞ Call the operation to transfer data as if shift registers do "a scan"

Example of SCAN - Original Circuit

Example of original net (has 3 Flip-flops (FFs))

We can not test combinational logics if FF operates as a sequential circuit

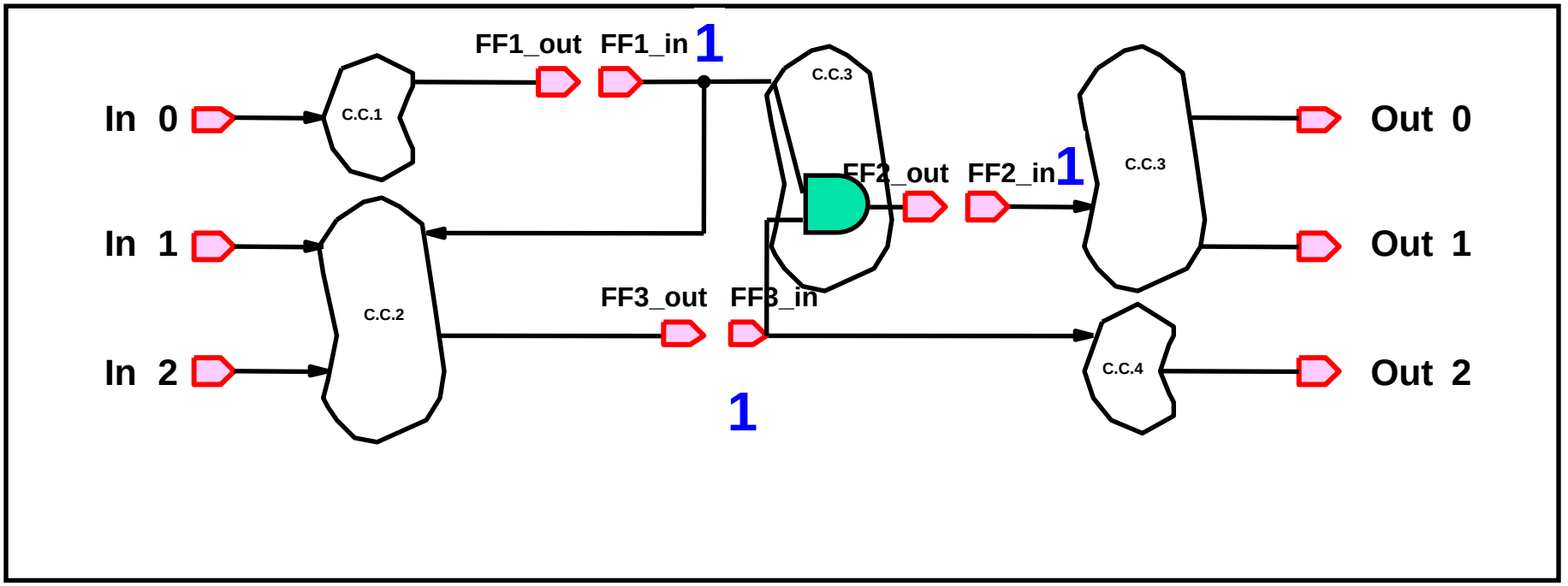


Testability is lowered because flip-flops(FF_1,FF_2 and FF_3) is a memory unit

Example of SCAN - Virtual Port

Treat FF as virtual input-output port

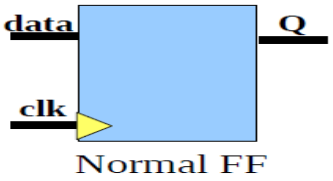
⇨ **Combination circuit without internal states**



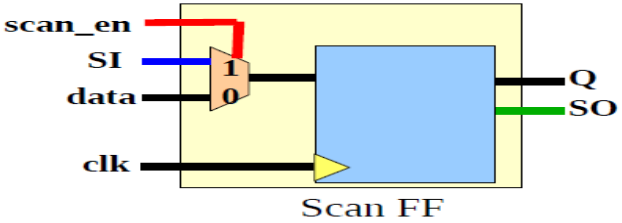
Treat FF_1 - FF_3 as virtual input-output ports and handle the entire circuit as combination circuits.

Example of SCAN - Actual Circuit

Normal FF



Scan FF

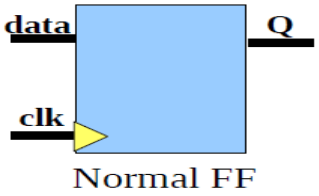


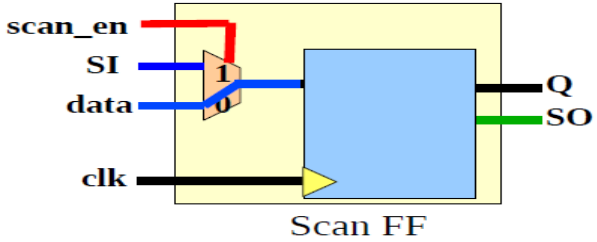
+ Normal FF (before apply SCAN)

TUHNQKDFFAQRZFC1X10U reg_3_9 (.DATA(N1), .CLK(N2), .Q(N4) ;

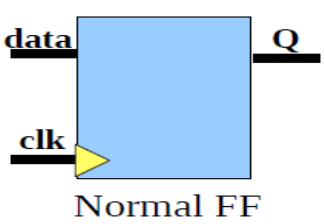
+ SCAN FF (after apply SCAN): new pin SI, SO, SMC (scan_enable) and their's connection are added.

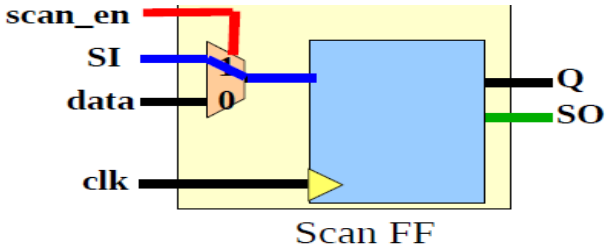
TUHNQKDFFAQRZFC1X10 reg_3_9 (.DATA(N1), .CLK(N2), . Q(N4), .SIN(ZQQ7IO008), .SMC(ZQQ7SEN_CLK), .SO(ZQQ7IO009));





+ scan_en=0: The data input to FF from data port.
(Scan-FF operates as Normal-FF)



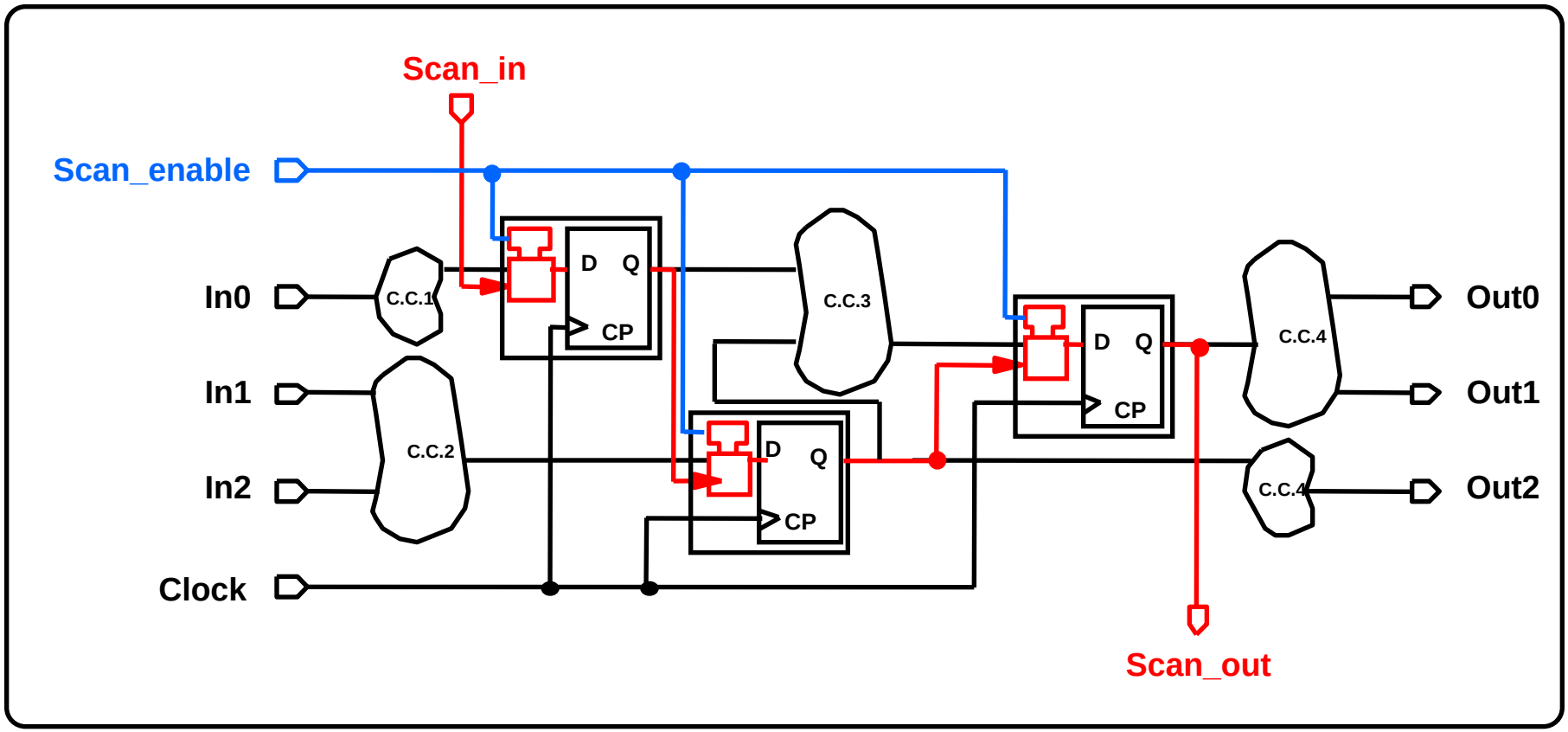


+ scan_en=1: The data input to FF from SI port.

Example of SCAN - Actual Circuit

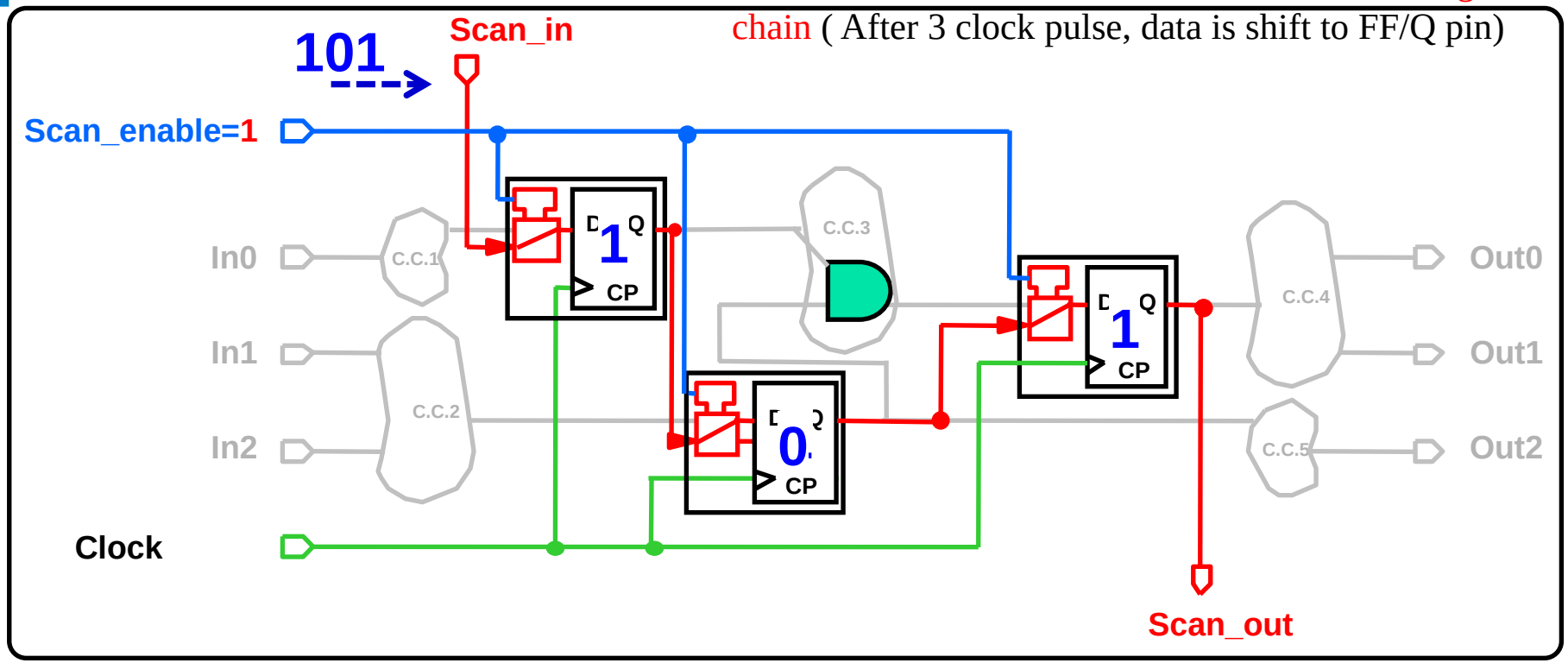
Blue Normal/SCAN operation switching circuit

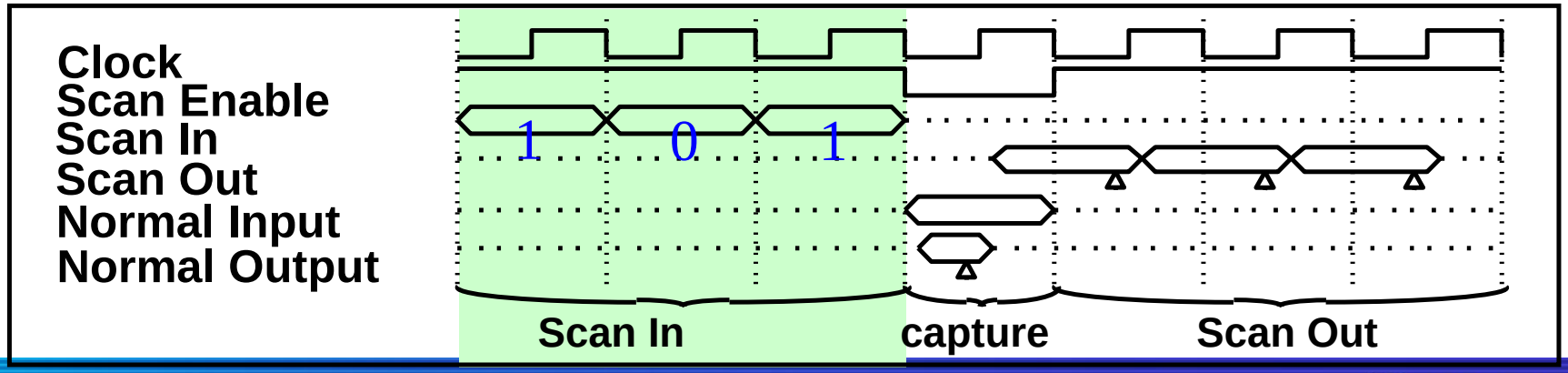
Red SCAN circuit



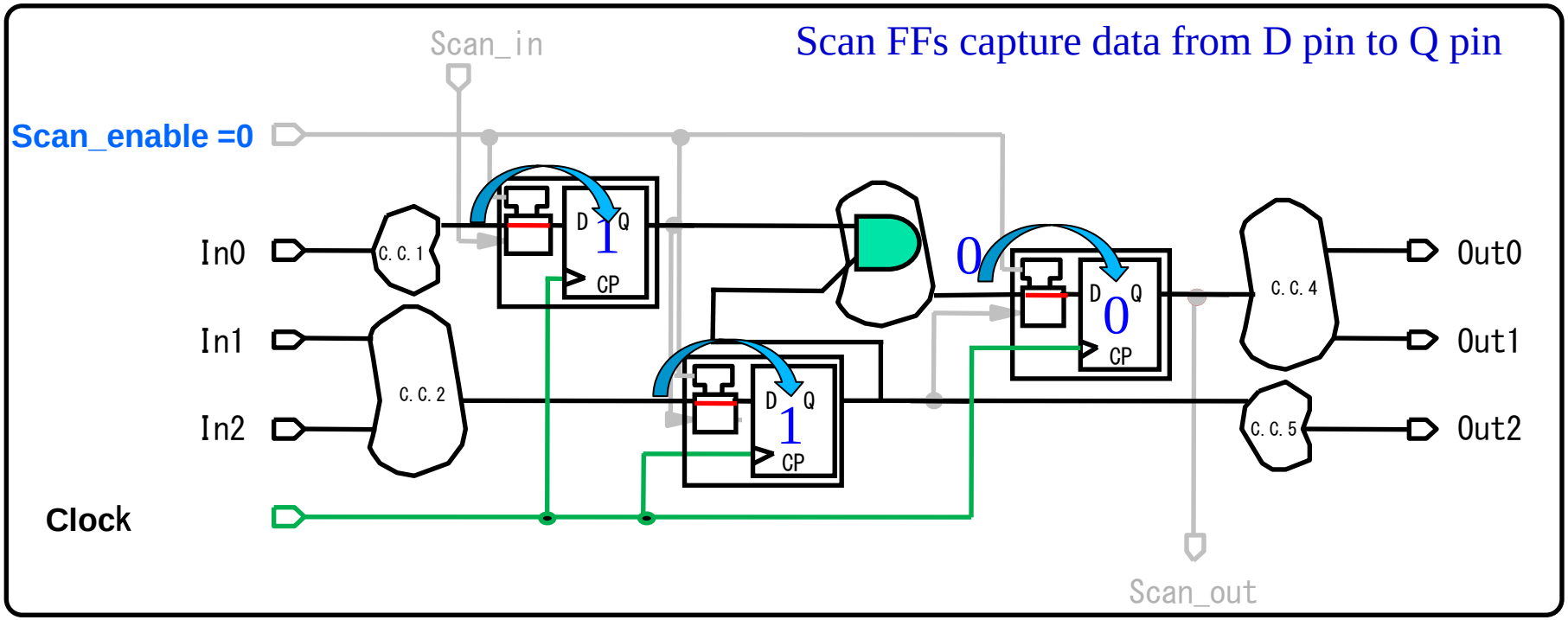
Example of SCAN - Scan In Operation

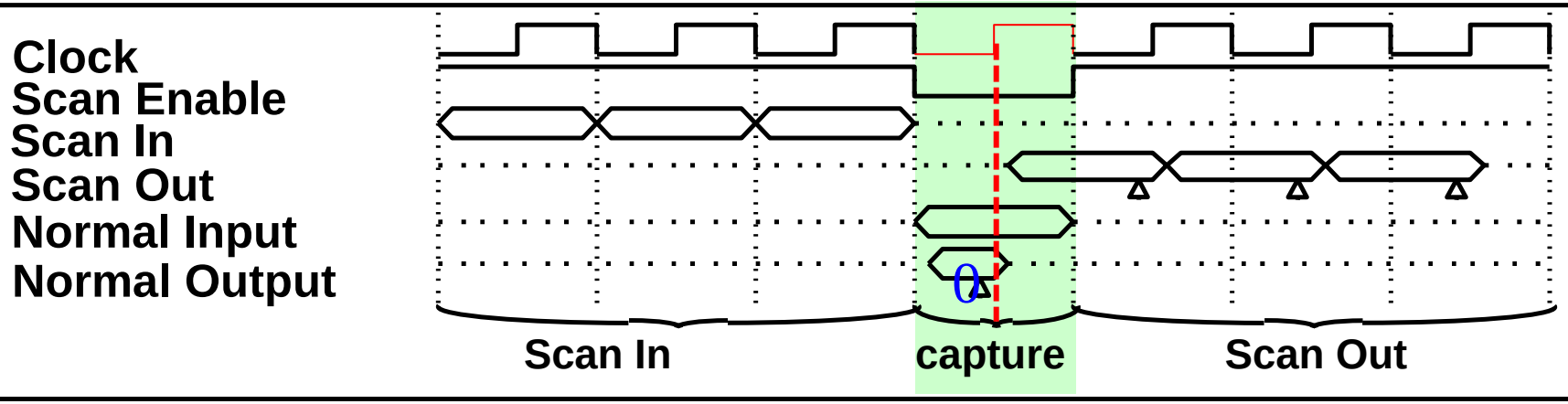
Data is shifted in from Scan_in to scan FFs through scan chain (After 3 clock pulse, data is shift to FF/Q pin)



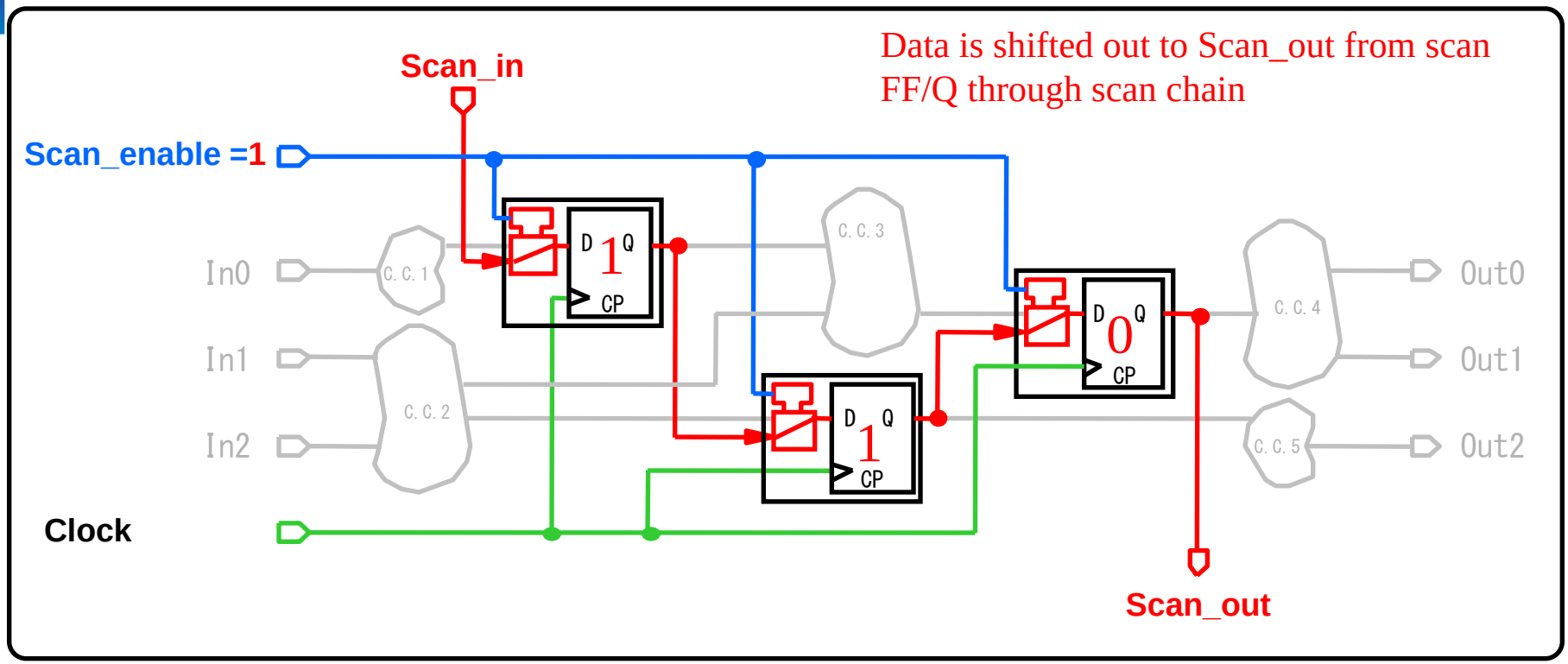


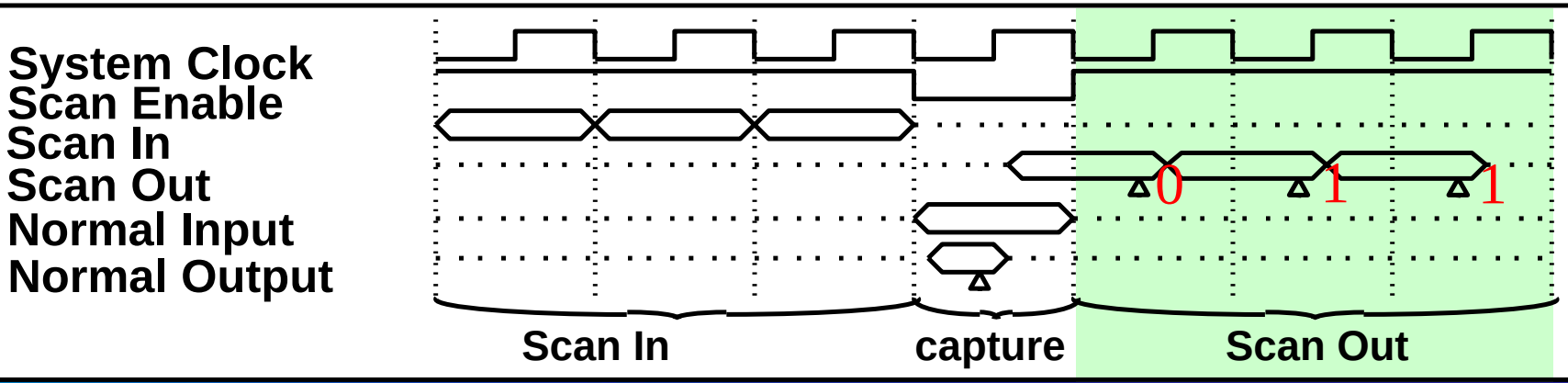
Example of SCAN - Capture Operation



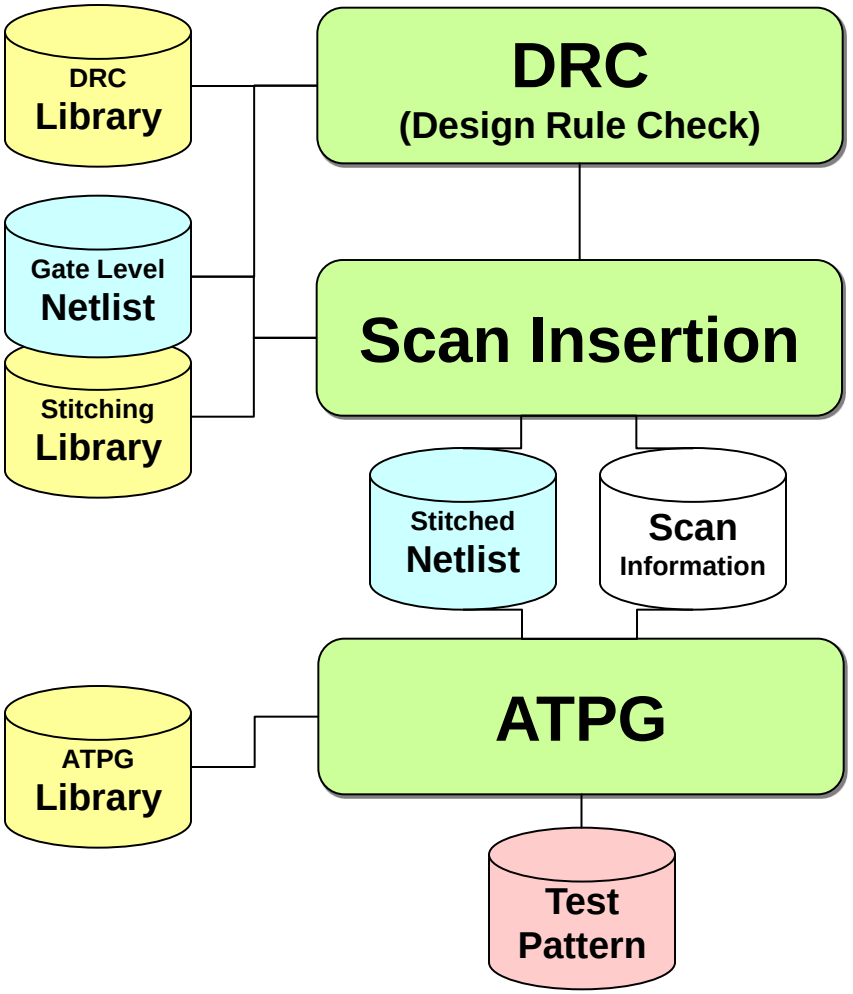


Example of SCAN - Scan Out Operation





SCAN Design Flow



DRC (Design Rule Check)

Verify a design satisfies design constraints for MUX scan method

Ex: clock/set/reset signal must be controllable during scan test

Scan Insertion

Construct scan chain

Compression and decompression circuits are also inserted in the use of scan compression

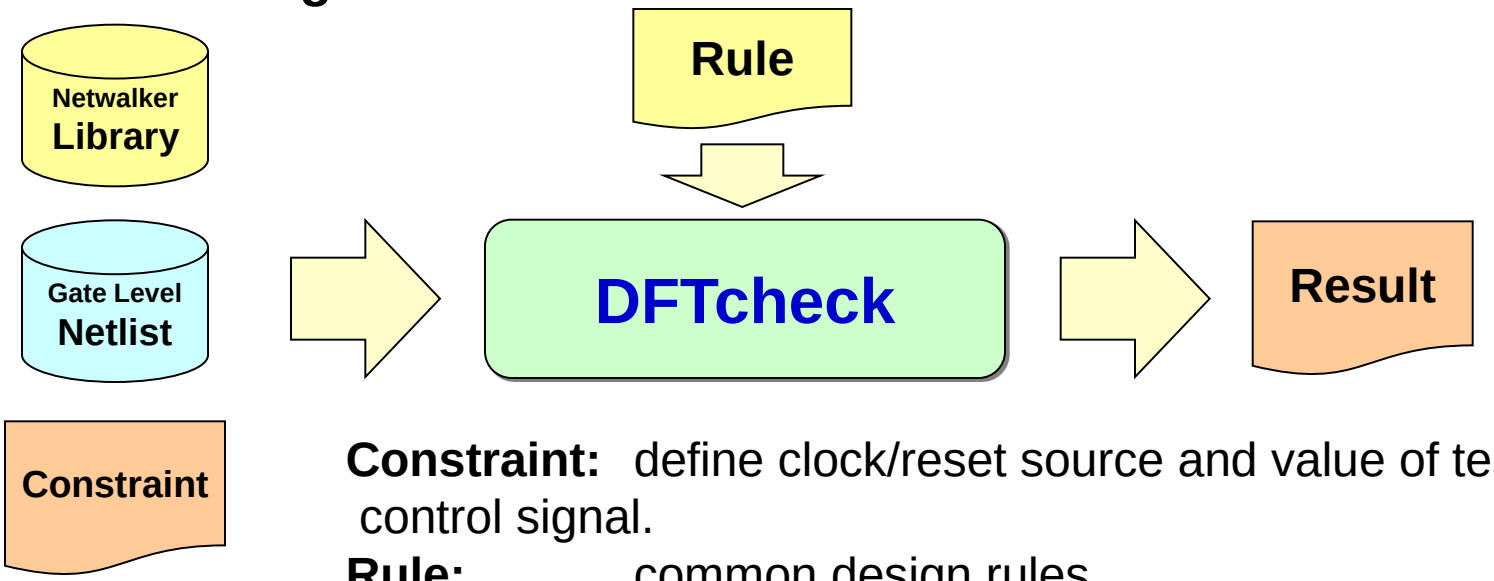
ATPG (Test pattern generation)

Generate test pattern

DRC - Design Rule Check

- A tool which can verify design constraints for MUX scan method is satisfied (In this figure, DFTcheck is in-house tool of RENESAS)
- Necessary constraints for MUX scan are covered
- Easy error analysis by detail error report and debugging functionality

IO Configuration



Constraint: define clock/reset source and value of test mode control signal.

Rule: common design rules

Result: check result/error summary. If design has no error, apply scan insertion. If it has error, feedback to logic designer

Scan Tools

Renesas introduced the following tools in use

- Tools from **Mentor Graphics** (MUX Scan) (Using in RVC)
  - ☞ **DFTAdvisor** Scan insertion and ATPG tool
  - ☞ **Tessent** Scan insertion and ATPG tool

- Tools from **Synopsys** (MUX Scan)
  - ☞ **DFTCompiler** Scan insertion tool
  - ☞ **DFTMAX**/**TetraMax** ATPG tool

- Tools from **Cadence** (MUX Scan)
  - ☞ Encounter Design Implementation (Scan insertion tool)
  - ☞ Encounter Test ATPG (ATPG tool)

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

(a) Purpose of SCAN Design

(b) Outline of SCAN Design

(c) Logic Fault Model

- *Stuck-at Fault*
- Transition Fault
- Path Delay Fault

(d) Applied SCAN Design (LBIST etc.)

(e) IDDQ Test

3. DFT for Embedded Memory

Stuck-at Fault Model in Logic Parts - 1

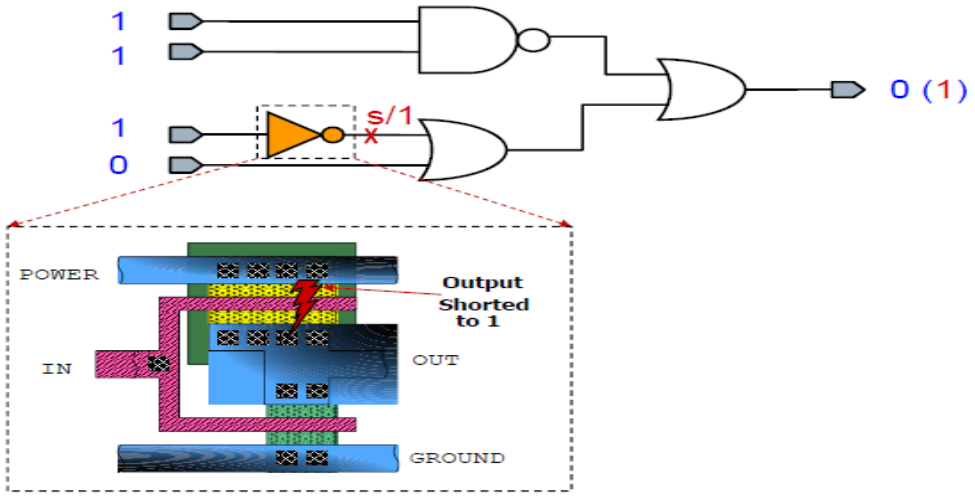
Fault models are necessary for generating and evaluating a set of test vectors. Generally, a good fault model should satisfy two criteria:

- Accuracy reflect the behavior of defects
- Computationally efficient in terms of fault simulation and test pattern generation

Many fault models have been proposed but no single fault model accurately reflects the behavior of all possible defects that can occur. As a result, a combination of different fault models is often used.

Single Stuck-at Fault Model:

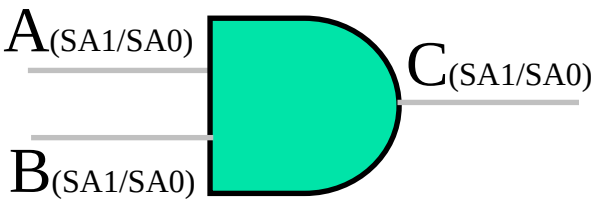
- A circuit with single stuck-at fault



- 01 line in the circuit is faulty at a time
- The faulty line is permanently set to 0 or 1 (low or high voltage)
- The fault can be at an input or output of a gate

Stuck-at Fault Model in Logic Parts - 2

Equivalent Fault: two faults are called equivalent if every pattern that detects one of the faults also detects the other.



- SA1: stuck-at 1, SA0: stuck-at 0

Original stuck-at fault list: {A/0, A/1, B/0, B/1, C/0, C/1}

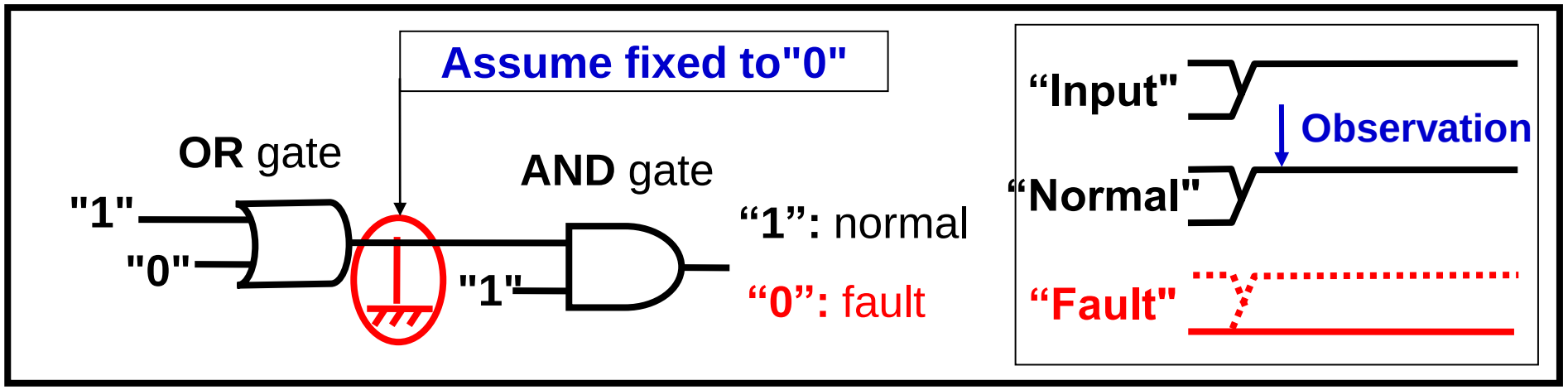
A/0, B/0, C/0 are equivalent. Therefore, reduced fault list: {A/1, B/1, C/1, (C/0 or B/0 or A/0)}

Reducing equivalent fault can help improve both test generation and simulation time.

- AND gate: input stuck-at 0 is equivalent to output stuck-at 0
- OR gate: input stuck-at 1 is equivalent to output stuck-at 1
- NAND gate: output stuck-at 1 is equivalent to any input stuck-at 0
- NOR gate: output stuck-at 0 is equivalent to any intput stuck-at 1

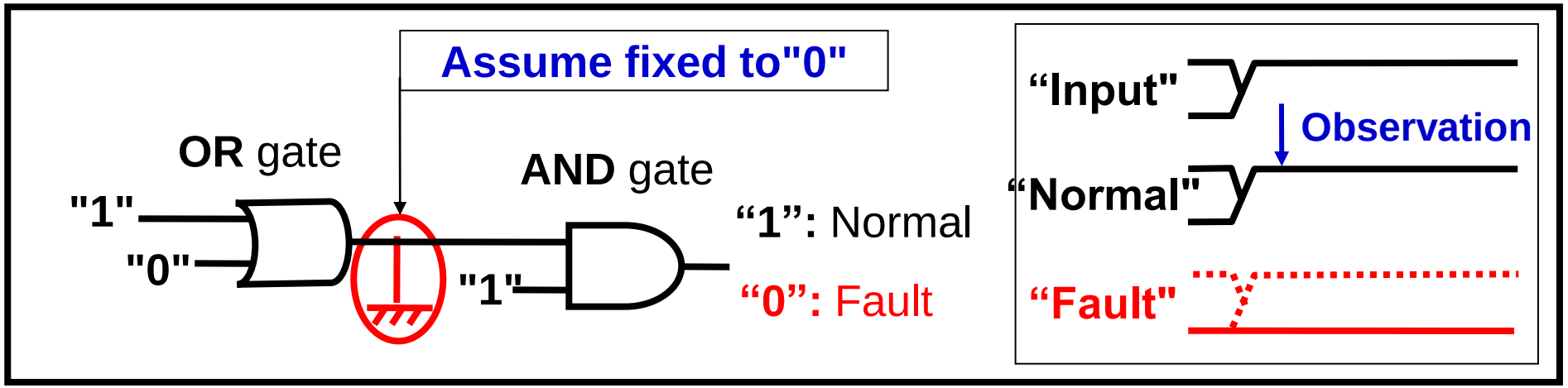
Stuck-at Fault Model in Logic Parts - 3

Single Stuck-at Fault Model: only 01 line in the circuit is faulty at a time



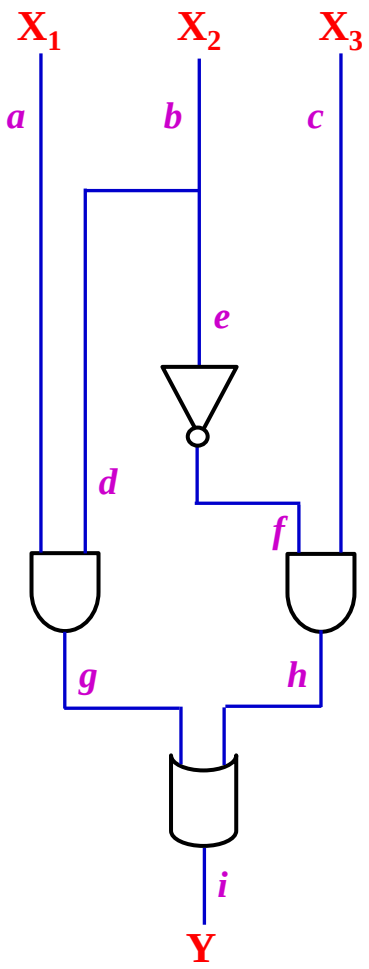
- **Fault assumption**
  - ☞ Assuming **1/0** fault to all pins of a cell, and make report without reduction (specified in QS9000)
  - ☞ There exists logically equivalent faults
- **Fault detection**
  1. Set reverse value of **1/0** at the position where **0/1** fault is assumed
  2. If the change is observed at the observation point (output pin or scan FF), then the fault can be detected

Stuck-at Fault Model in Logic Parts - 4



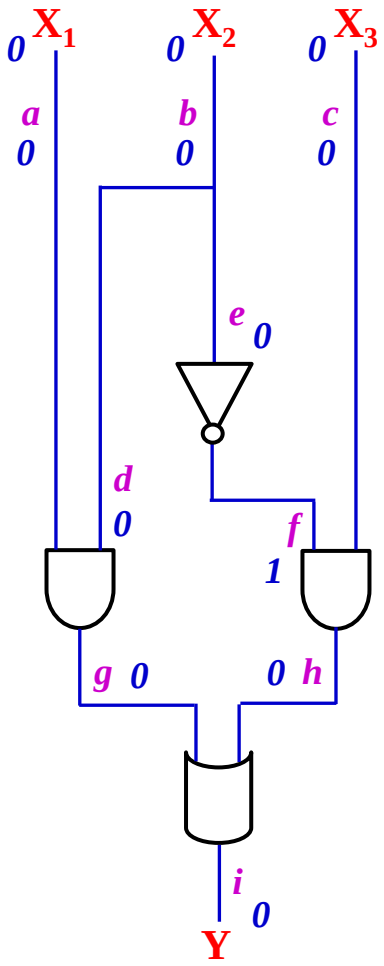
- **Measurement condition**
  - ☞ Since the model does not consider delay and timing, there is no specific measurement condition.

Stuck-at Fault Model - Example (1/6)



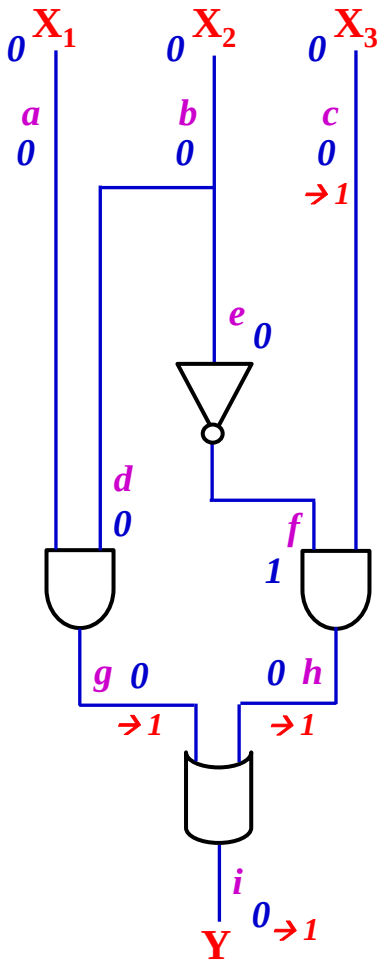
$X_1\ X_2\ X_3$		000	001	010	011	100	101	110	111
Y		0	1	0	0	0	1	1	1
a	SA0								
	SA1								
b	SA0								
	SA1								
c	SA0								
	SA1								
d	SA0								
	SA1								
e	SA0								
	SA1								
f	SA0								
	SA1								
g	SA0								
	SA1								
h	SA0								
	SA1								
i	SA0	0	0	0	0	0	0	0	0
	SA1	1	1	1	1	1	1	1	1

Stuck-at Fault Model - Example (2/6)



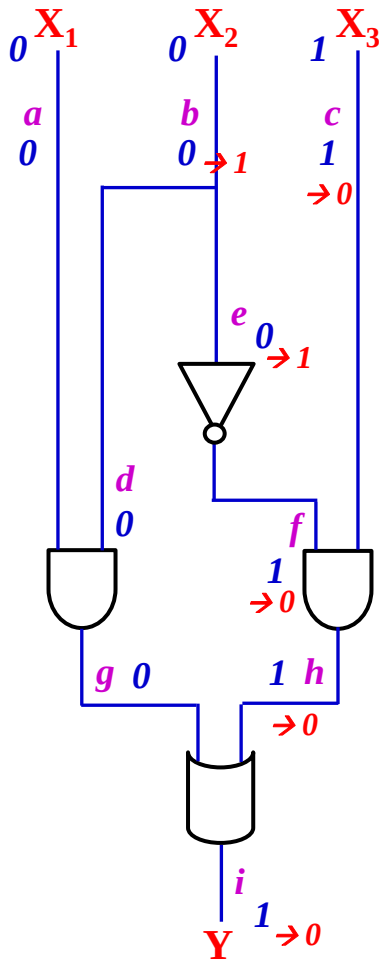
$X_1\ X_2\ X_3$		000	001	010	011	100	101	110	111
Y		0	1	0	0	0	1	1	1
a	SA0	0							
	SA1								
b	SA0	0							
	SA1								
c	SA0	0							
	SA1								
d	SA0	0							
	SA1								
e	SA0	0							
	SA1								
f	SA0								
	SA1	1							
g	SA0	0							
	SA1								
h	SA0	0							
	SA1								
i	SA0	0							
	SA1								

Stuck-at Fault Model - Example (3/6)



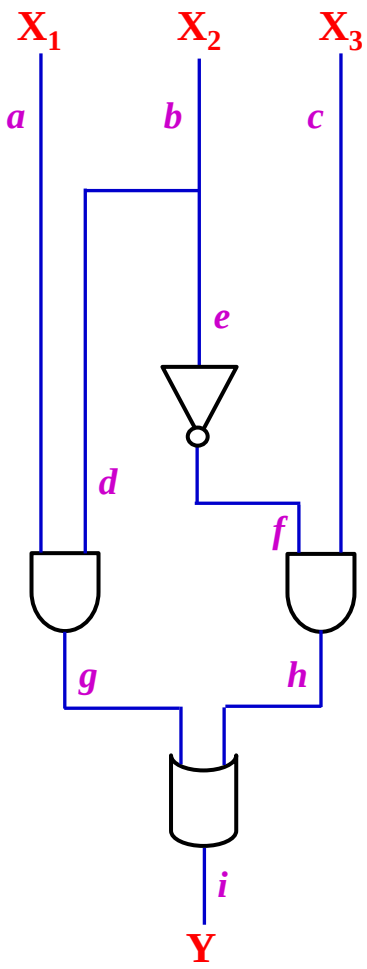
$X_1\ X_2\ X_3$		000	001	010	011	100	101	110	111
Y		0	1	0	0	0	1	1	1
a	SA0	0							
	SA1								
b	SA0	0							
	SA1								
c	SA0	0							
	SA1	1							
d	SA0	0							
	SA1								
e	SA0	0							
	SA1								
f	SA0								
	SA1	1							
g	SA0	0							
	SA1	1							
h	SA0	0							
	SA1	1							
i	SA0	0							
	SA1	1							

Stuck-at Fault Model - Example (4/6)



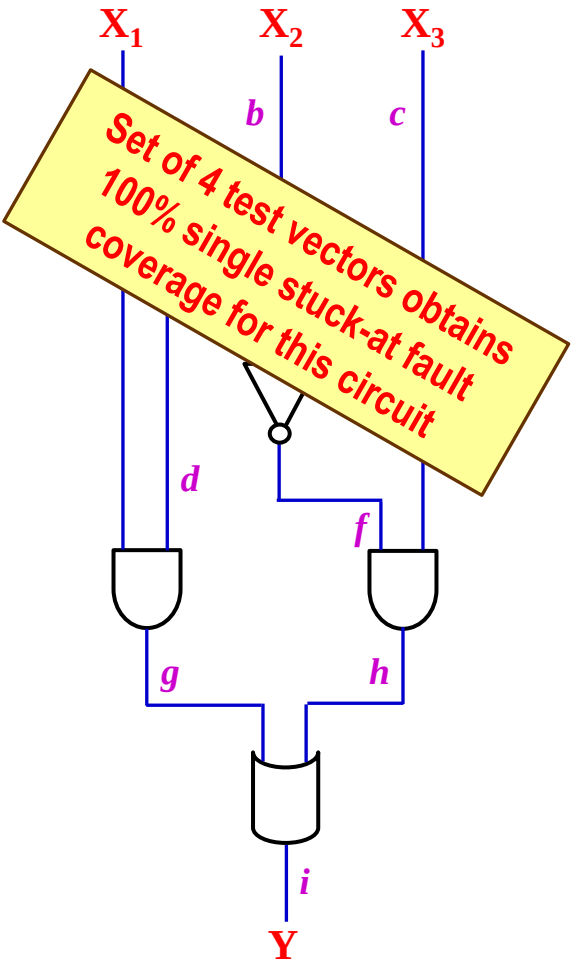
$X_1\ X_2\ X_3$		000	001	010	011	100	101	110	111
Y		0	1	0	0	0	1	1	1
a	SA0	0	0						
	SA1								
b	SA0	0	0						
	SA1		1						
c	SA0	0	0						
	SA1	1	1						
d	SA0	0	0						
	SA1								
e	SA0	0	0						
	SA1		1						
f	SA0		0						
	SA1	1	1						
g	SA0	0	0						
	SA1	1							
h	SA0	0	0						
	SA1	1	1						
i	SA0	0	0						
	SA1	1	1						

Stuck-at Fault Model - Example (5/6)



$X_1\ X_2\ X_3$		000	001	010	011	100	101	110	111
Y		0	1	0	0	0	1	1	1
a	SA0	0	0	0	0			0	0
	SA1			1	1	1	1	1	1
b	SA0	0	0		0	0	0	0	0
	SA1		1	1	1	1	1	1	1
c	SA0	0	0	0		0	0	0	
	SA1	1	1		1	1			1
d	SA0	0	0			0	0	0	0
	SA1			1	1	1		1	1
e	SA0	0	0		0	0	0		
	SA1		1	1	1		1	1	1
f	SA0		0	0	0		0	0	0
	SA1	1	1		1	1	1		
g	SA0	0	0	0	0	0	0	0	0
	SA1	1		1	1	1		1	1
h	SA0	0	0	0	0	0	0	0	0
	SA1	1	1	1	1	1	1		
i	SA0	0	0	0	0	0	0	0	0
	SA1	1	1	1	1	1	1	1	1

Stuck-at Fault Model - Example (6/6)



$X_1\ X_2\ X_3$		000	001	010	011	100	101	110	111
Y		0	1	0	0	0	1	1	1
a	SA0	0	0	0	0			0	0
	SA1			1	1	1	1	1	1
b	SA0	0	0		0	0	0	0	0
	SA1		1	1	1	1	1	1	1
c	SA0	0	0	0		0	0	0	
	SA1	1	1		1	1			1
d	SA0	0	0			0	0	0	0
	SA1			1	1	1		1	1
e	SA0	0	0		0	0	0		
	SA1		1	1	1		1	1	1
f	SA0		0	0	0		0	0	0
	SA1	1	1		1	1	1		
g	SA0	0	0	0	0	0	0	0	0
	SA1	1		1	1	1		1	1
h	SA0	0	0	0	0	0	0	0	0
	SA1	1	1	1	1	1	1		
i	SA0	0	0	0	0	0	0	0	0
	SA1	1	1	1	1	1	1	1	1

Outline Design For Test (DFT)

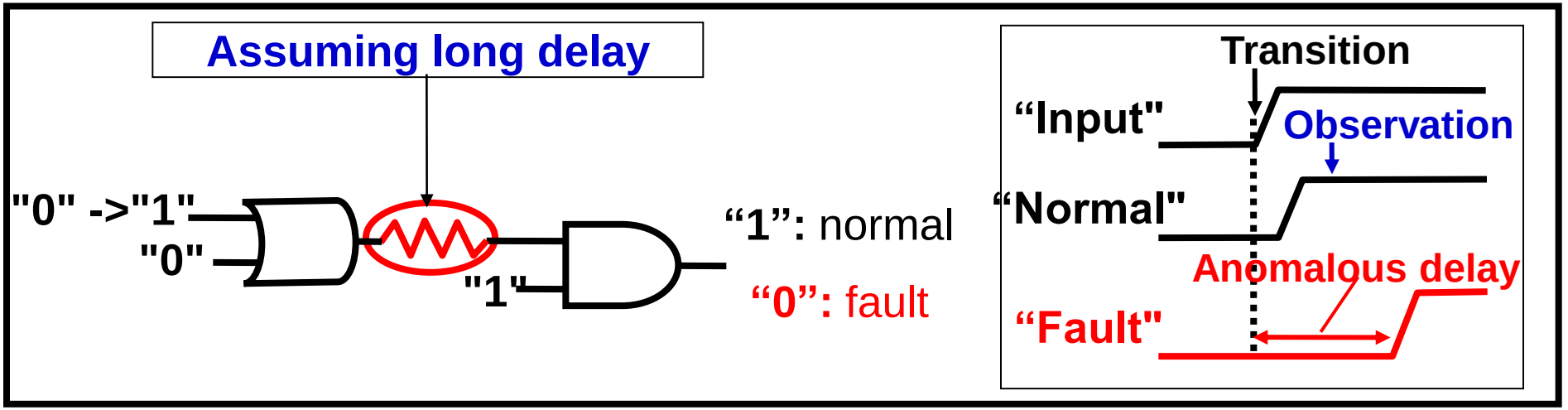
1. Importance of DFT

2. DFT for Logic Parts

- (a) Purpose of SCAN Design
- (b) Outline of SCAN Design
- *(c) Logic Fault Model*
  - Stuck-at Fault
  - *Transition Fault*
  - Path Delay Fault
- (d) Applied SCAN Design (LBIST etc.)
- (e) IDDQ Test

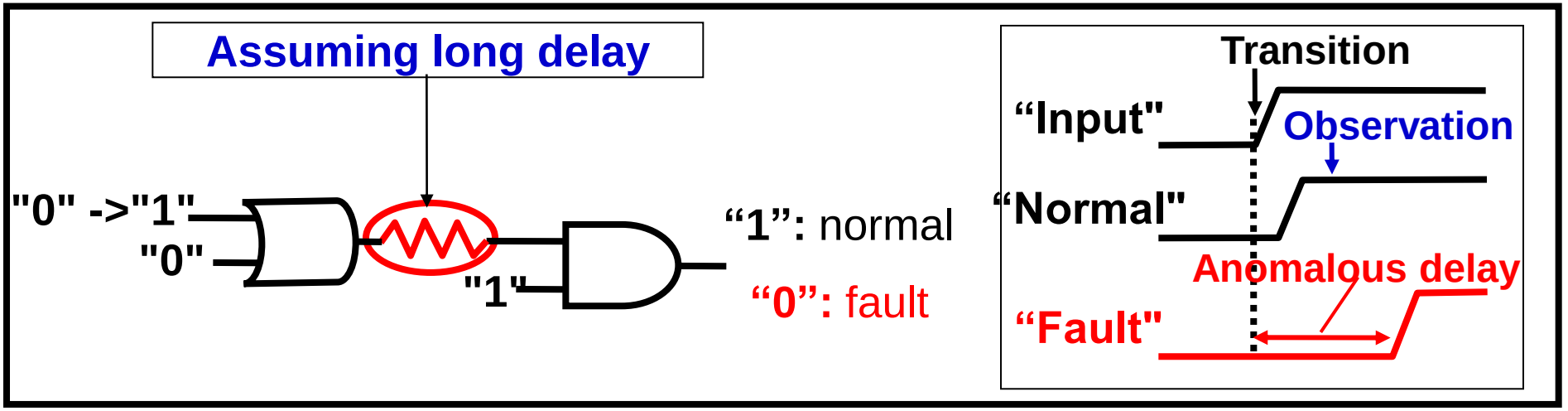
3. DFT for Embedded Memory

Transition Fault Model in Logic Parts - 1



- **Fault assumption**
  - ☞ Assume a long delay that can not be propagated within one clock cycle at all pins of a cell. The equivalent faults are reduced by a tool
- **Fault detection**
  - ☞ If the transition (signal change of **0**->**1** or **1**->**0**) occurs at the position where the delay fault is assumed is detected at the observation point, then the fault is detected. (In case of the transition of **0**->**1**, if “**1**” is observed, it is normal, and if “**0**” is observed, it is the transition fault).

Transition Fault Model in Logic Parts - 2



- **Measurement condition**
  - ☞ Test Frequency

  The detective size of delay fault depends on the time difference between “timing of input transition” and “timing of transition observation”. The detection capability is enhanced, if the Test Frequency is close to the actual operation, but we need to determine the Test Frequency considering false path and multi-cycle path.

Outline Design For Test (DFT)

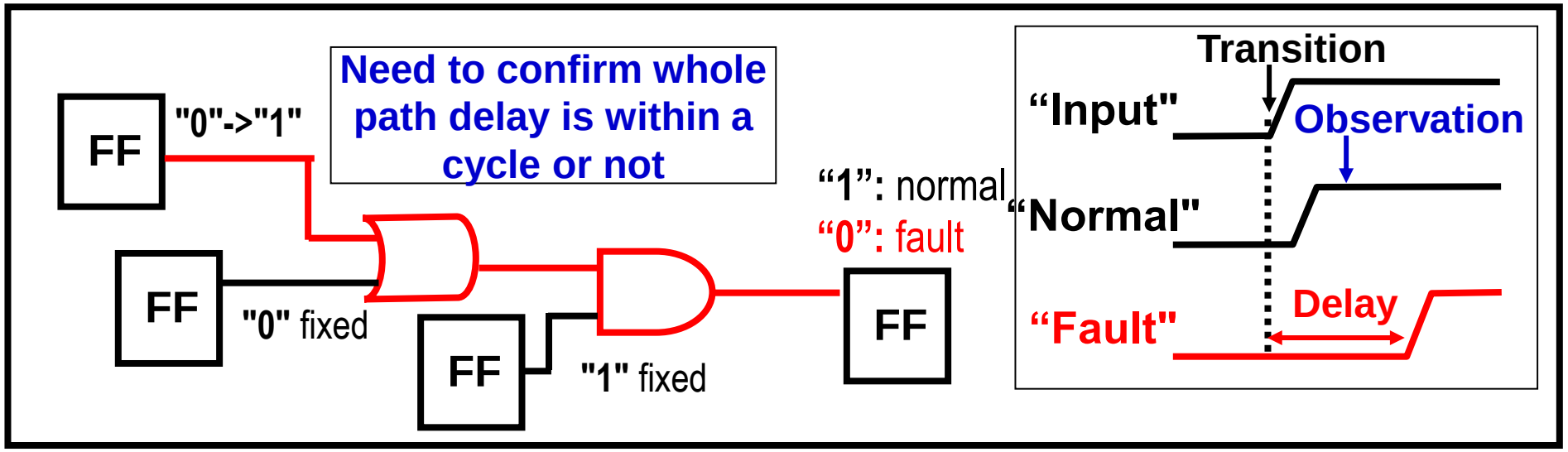
1. Importance of DFT

2. DFT for Logic Parts

- (a) Purpose of SCAN Design
- (b) Outline of SCAN Design
- *(c) Logic Fault Model*
  - Stuck-at Fault
  - Transition Fault
  - *Path Delay Fault*
- (d) Applied SCAN Design (LBIST etc.)
- (e) IDDQ Test

3. DFT for Embedded Memory

Path Delay Fault Model in Logic Parts



- **Fault assumption**
  - ☞ Assuming a fault to a specific path by considering the result of STA (*Static Timing Analysis*) and critical paths. It is not practical to assume faults to all paths (huge numbers of paths) and N pieces of critical worst paths are chosen
- **Fault detection**
  - ☞ Maintaining that the whole path is not affected by other signal changes during test, and evaluate that the signal transition is transferred through the tested path within a cycle. (can test at fixed frequency and detect of fixed delay with the control of clock cycles).

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

(a) Purpose of SCAN Design

(b) Outline of SCAN Design

(c) Logic Fault Model

(d) Applied SCAN Design (LBIST, etc.)

- ***AC-SCAN***
- Compression SCAN
- LBIST (Logic Built-In Scan Test)

(e) IDDQ Test

3. DFT for Embedded Memory

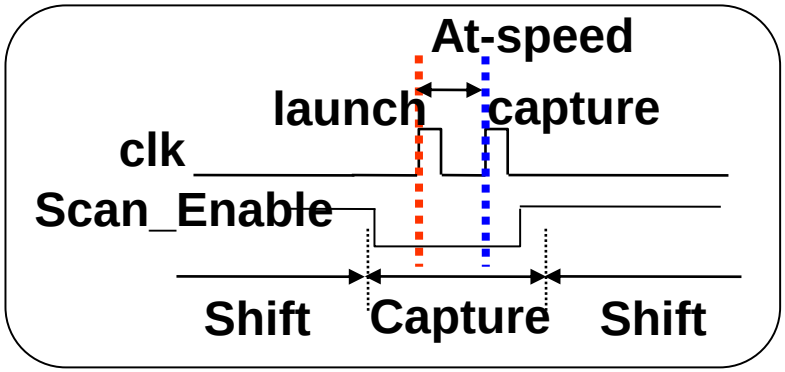
AC SCAN Test

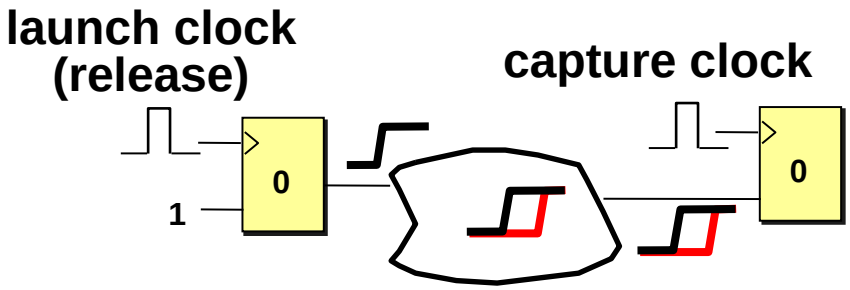
Delay-induced failure increases due to miniaturization

- Difficult to detect it using Stuck-at Fault Model
- Minimal delay failure causes critical problems in high-speed LSI

Apply AC SCAN test

- Adopt MUX SCAN test method
- Use Transition Fault Model
- Input the test pattern and observe at speed
- Test pattern generation methods
  a) Launch-off-Shift (**LOS**)
  b) Launch-off-Capture (**LOC**)

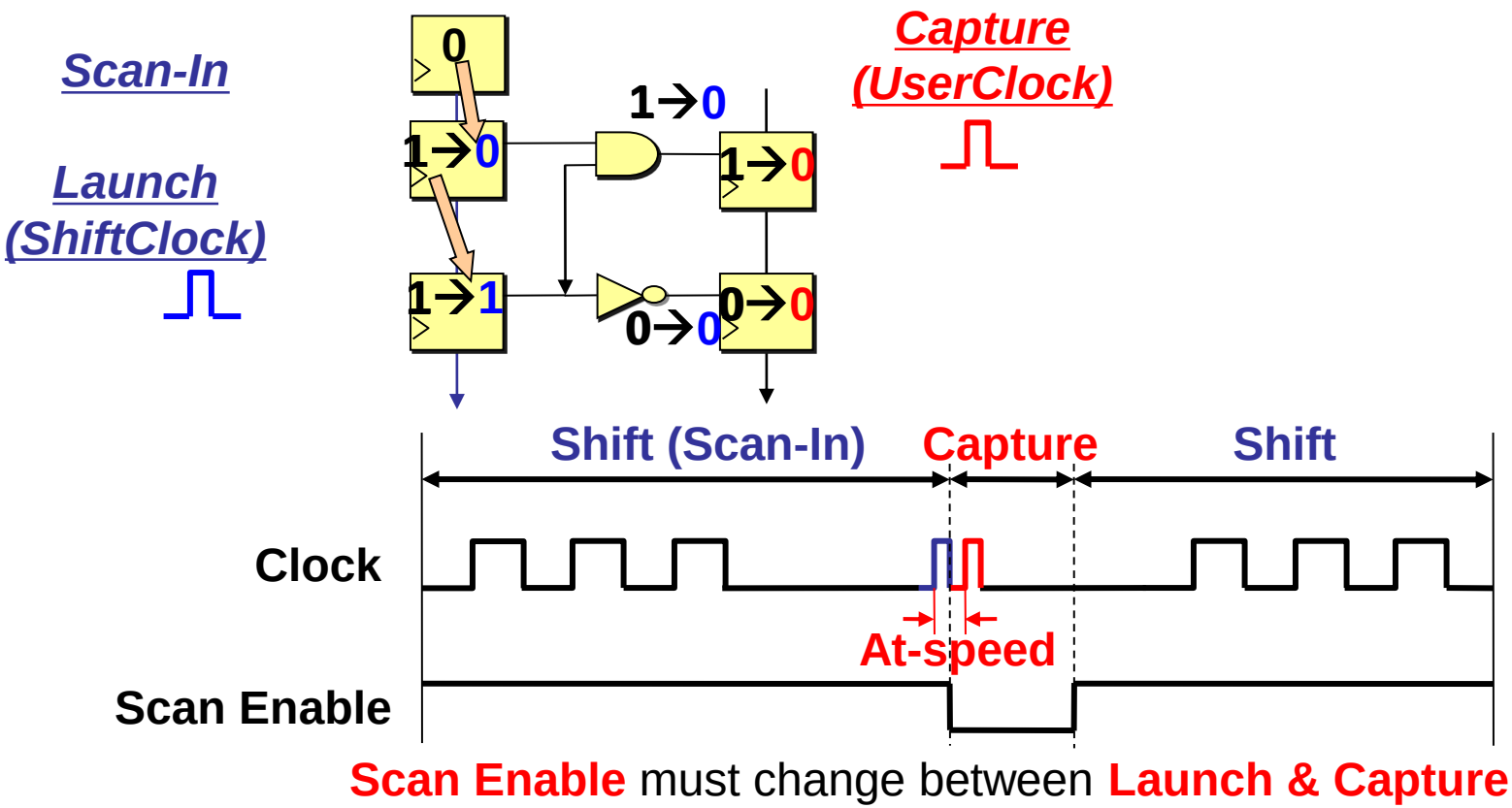




Synchronous circuits

Launch-off-Shift (LOS) / Skewed Load

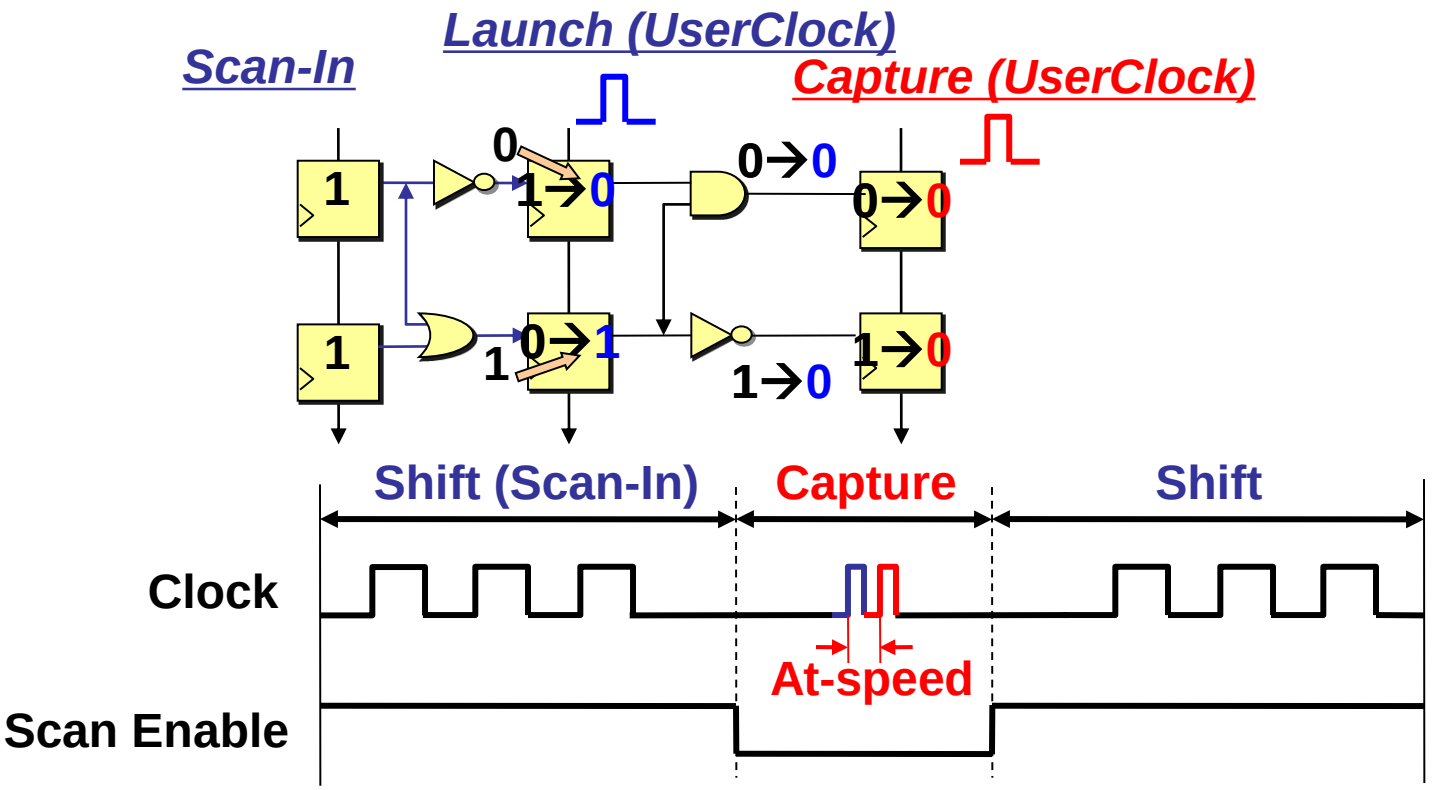
- Enable to generate signal change directly through SCAN chain
  - ☞ Expectation of high test coverage
- Difficult to control timing of scan enable signal



Scan Enable must change between **Launch & Capture**

Launch-off-Capture (LOC) / Broad Side

- Pattern must be set through user function path
  - ☞ Test coverage not higher than expectation
- Easy to control timing of scan enable signal



Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

(a) Purpose of SCAN Design

(b) Outline of SCAN Design

(c) Logic Fault Model

(d) Applied SCAN Design (LBIST, etc.)

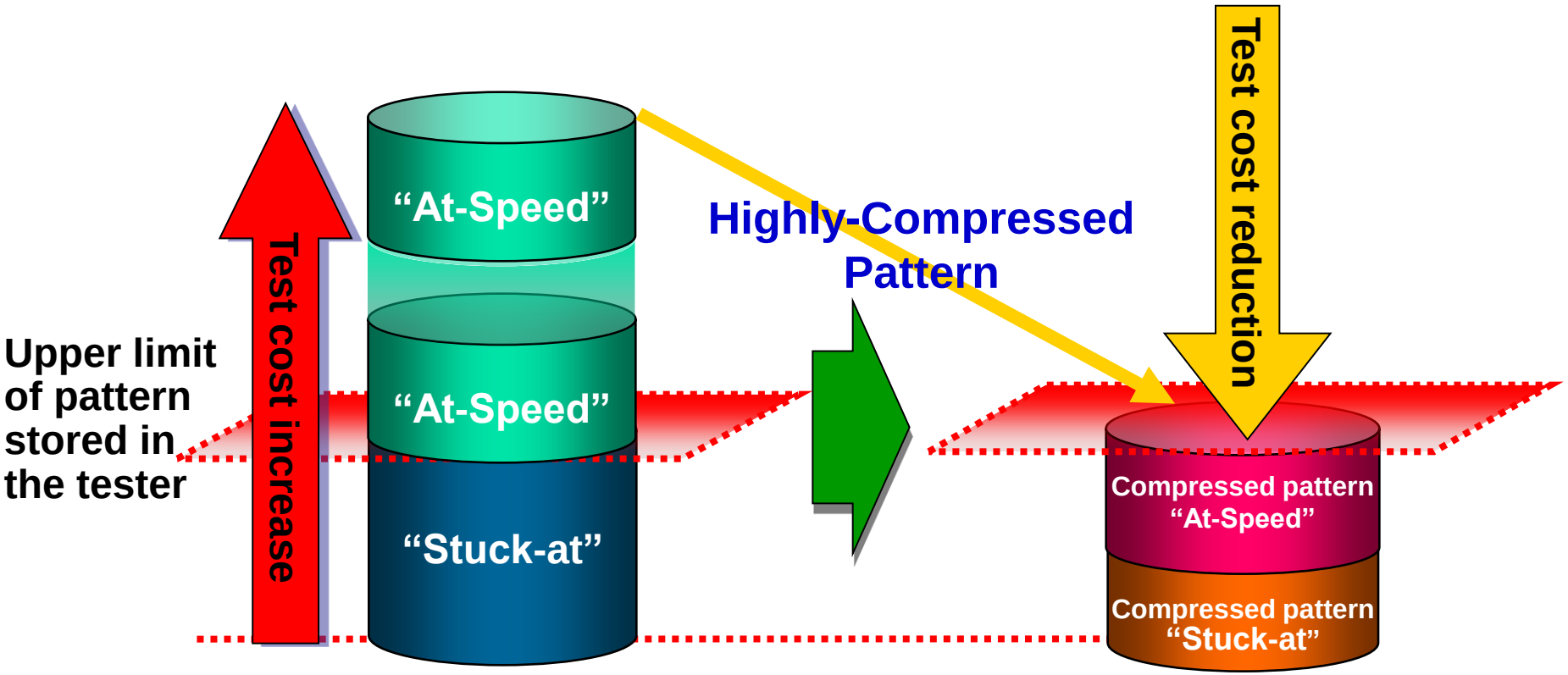
- AC-SCAN
- ***Compression SCAN***
- LBIST (Logic Built-In Scan Test)

(e) IDDQ Test

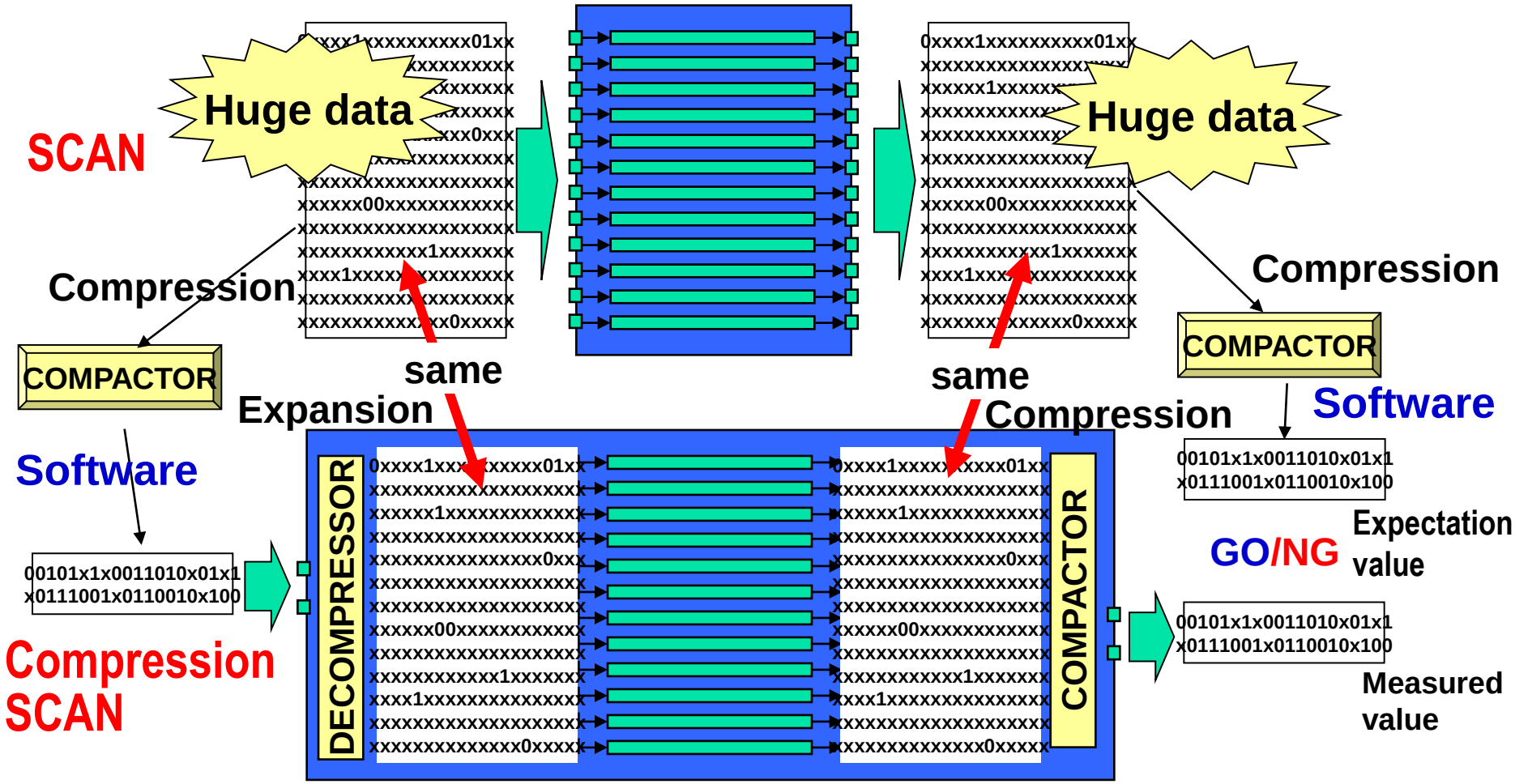
3. DFT for Embedded Memory

Necessity for Highly-Compressed Pattern

- Need more pattern generated for AC SCAN test (3 to 5 times)
- Drastically Highly-Compressed Pattern is necessary to cope with large scale while maintaining the higher test quality



Basic Concept of Compression SCAN



Data compression/expansion circuits (**DECOMPRESSOR/COMPACTOR**) in a LSI for SCAN input/output can reduce pattern data size

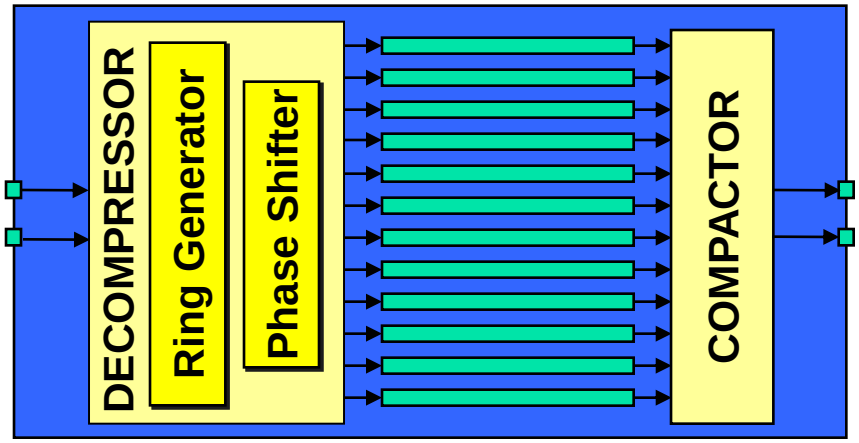
Basic Concept of Compression SCAN

- Advantage:
  - Reduce scan in/out port
  - Reduce test pattern's volume

  → Reduce test time, test cost
- Disadvantage:
  - Increase chip size (~ 2-3% total chip)
  - More complicated design flow in DFT implementation

Compression SCAN: TESTKOMPRESS (Mentor)

- Ensure the same fault coverage as the normal ATPG
- Enable pattern generation for Stuck-at, Transition, and Path Delay Faults
- Automatically insert SCAN + Compression/Expansion Circuit IP
  - ☞ **DECOMPRESSOR**: Ring Generator (LFSR) + Phase Shifter (EX-OR)
  - ☞ **COMPACTOR**: EX-OR-Tree (with indefinite and aliasing block function)
  - ☞ Circuit overhead is about 17Gate per an internal chain
- Realize about 1/400 of pattern compressibility



LFSR: Linear **F**eedback **S**hift **R**egister

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

(a) Purpose of SCAN Design

(b) Outline of SCAN Design

(c) Logic Fault Model

(d) Applied SCAN Design (LBIST, etc.)

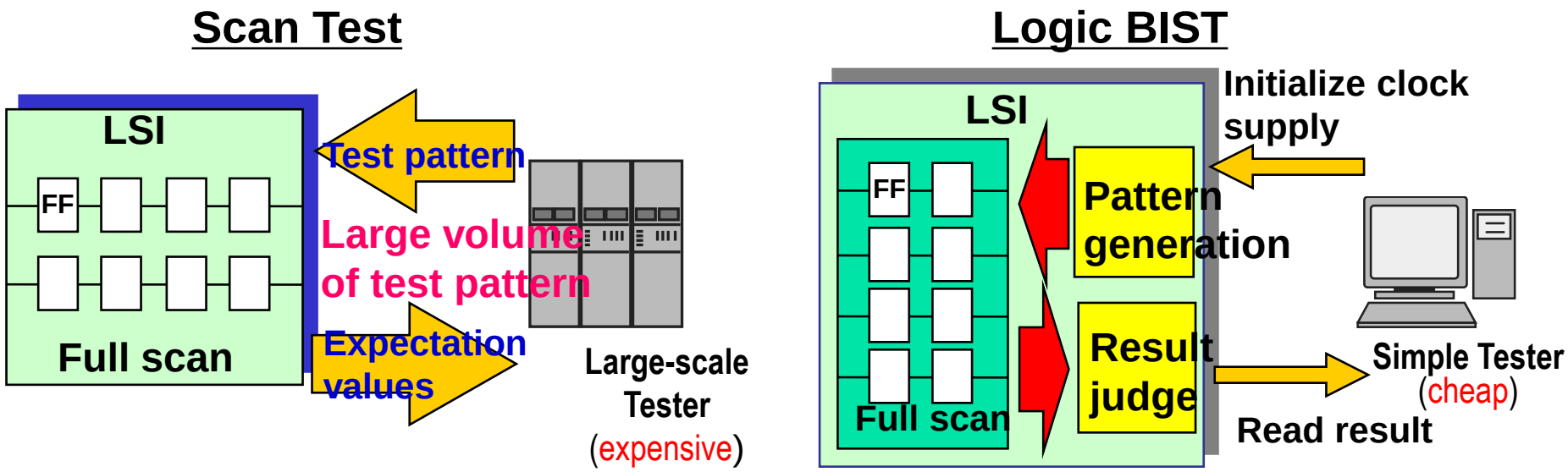
- AC-SCAN
- Compression SCAN
- *LBIST (Logic Built-In Scan Test)*

(e) IDDQ Test

3. DFT for Embedded Memory

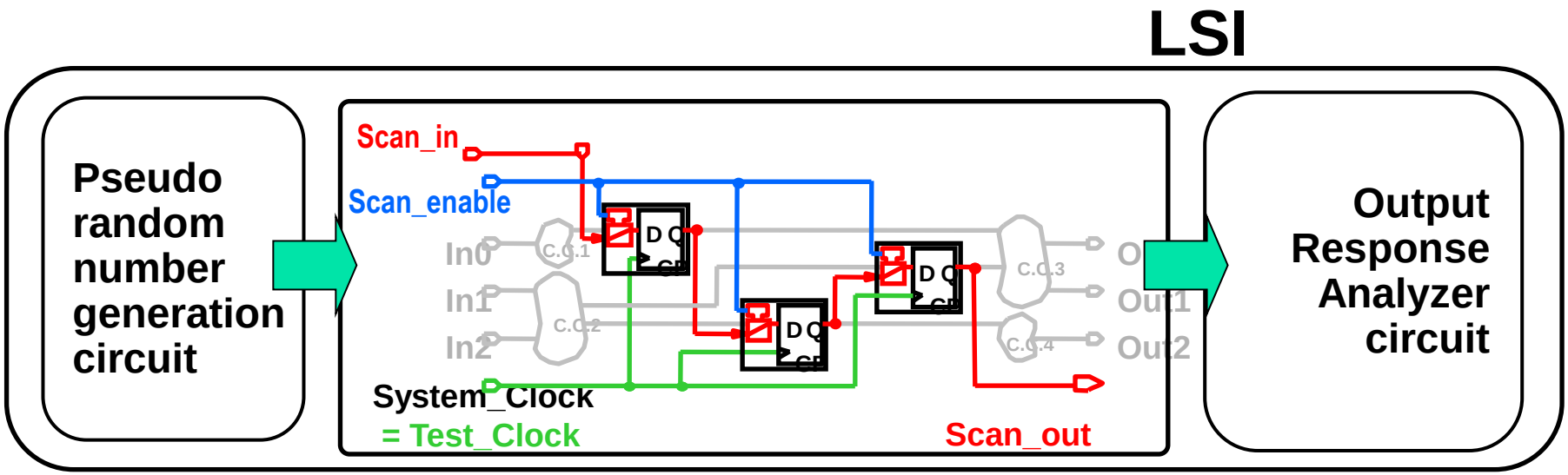
BIST (*Built-In Self Test*)

- Pattern generation circuits and pass/fail judge circuits are inserted into an LSI
- Can test an LSI using pseudo random number patterns
- Can judge pass/fail after a test completed



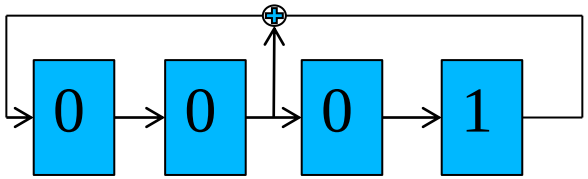
Principle of LBIST (1)

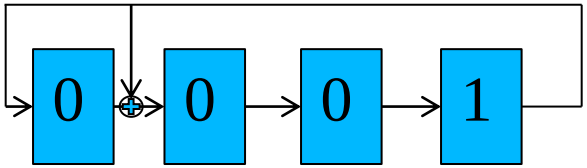
- Pattern generation circuits: provide pattern by pseudo random number generation circuits
- Output Response Analyzer circuits: compact output responses into a “signature” and compare with a “golden signature” by MISR (Multi Input Signature Register)



Principle of LBIST (2)

- Pattern generation circuits: provide pattern by pseudo random number generation circuits (TPG). For example: LFSR circuits





I1	I2	I3	I4
0	0	0	1
1	0	0	0
0	1	0	0
1	0	1	0
0	1	0	1
0	0	1	0
0	0	0	1
1	0	0	0
0	1	0	0
1	0	1	0
0	1	0	1
0	0	1	0
0	0	0	1
1	0	0	0
0	1	0	0
1	0	1	0

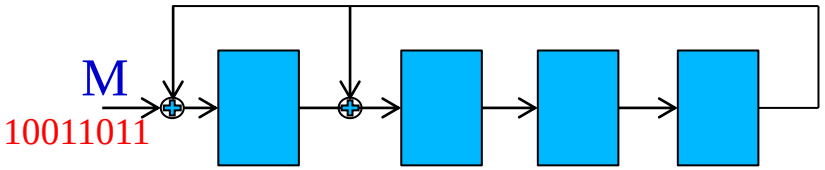
Initial value

I1	I2	I3	I4
0	0	0	1
1	1	0	0
0	1	1	0
0	0	1	1
1	1	0	1
1	0	1	0
0	1	0	1
1	1	1	0
0	1	1	1
1	1	1	1
1	0	1	1
1	0	0	1
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

TPG: Test Pattern Generation
LFSR: Linear Feedback Serial Register

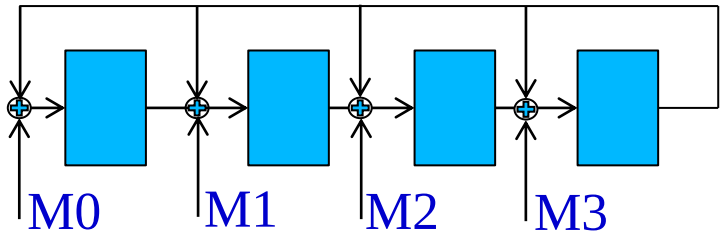
Principle of LBIST (3)

- Output Response Analyzer circuits: compact output responses into a “signature” and compare with a “golden signature” (Golden signature is signature for the fault-free circuit).
- Below are 2 kind of signature analysis schemes:



M (input)	r0	r1	r2	r3
1	0	0	0	0
1	1	0	0	0
0	1	1	0	0
1	0	1	1	0
1	1	0	1	1
0	0	0	0	1
0	1	1	0	0
1	0	1	1	0
R (signature)	1	0	1	1

Serial signature analysis
(SISR: single-input signature register)



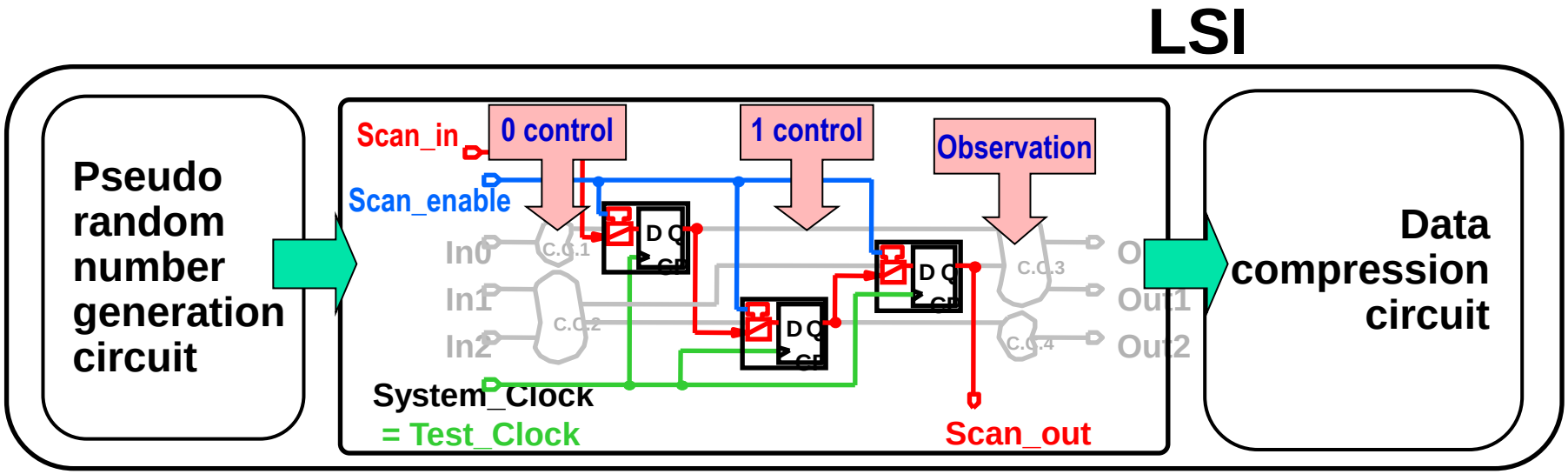
M0, M1, M2, M3: parallel inputs

Parallel signature analysis
(MISR: multiple-input signature register)

Principle of LBIST (4)

- Fault coverage improvement: **TPI** (*Test Point Insertion*)
  - ☞ It is not easy to improve fault coverage only by pseudo random number.
  - ☞ Insert control/observation circuits (test points) to the parts whose fault coverage is low when a test using random number is done to improve its coverage.

    ➔ Insert TPI: control FF and observer FF. These FFs are stitched in scan chain.



Effect of LBIST

- Reduce test data volume and the tester resource
  - Apply only clock for BIST during a test
    - ☞ Repeat and loop expressions in the test program are available for test pattern
    - ☞ At-speed test with easy IO timing constraint
      (I/F for scan in/out or user logic during a test is not needed.)
- Concurrent test for multi LSIs
  - Enable to control LBIST by only few BIST control pins

Application Guide for DFT for Logic Parts (Basic Concept)

Tool	FastScan (mentor) TetraMAX(synopsys)	DFTMAX(synopsis)	TestKompress (Mentor)	Singen (Renesas)	Tessent (Mentor)
Methodology	Scan	Compression Scan	Compression Scan	LBIST	LBIST + SCAN
Design Size	< 1M Gate	< 2M Gate	< 20M Gate	20M Gates <	flexible
Design Constraint	Easy (MUX-Scan)	Easy (MUX-Scan)	Easy (MUX-Scan)	Hard	Hard
Area Overhead	Scan	Scan + Adaptive Scan (0.1%-0.5%)	Scan + EDT (0.3%-1.0%)	Scan + LBIST + TP (2.0%)	Scan + LBIST + TP (5.0%)
Compression ratio	1	10x - 60x	10x - 260x	100x - 1000x	100x - 1000x
Low Pin Test	Not available	Not available	Available	Available	Available
At-Speed Test	Single Clock domain	Single Clock domain	Multiple Clock domain (< 3 Clock domain)	Multiple Clock domain	Multiple Clock domain

In RVC, Singen and TestKompress have been applied in many projects.
Tessent becomes more popular in recently project

TP: Test Points
EDT: Embedded Deterministic Testing Circuit

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

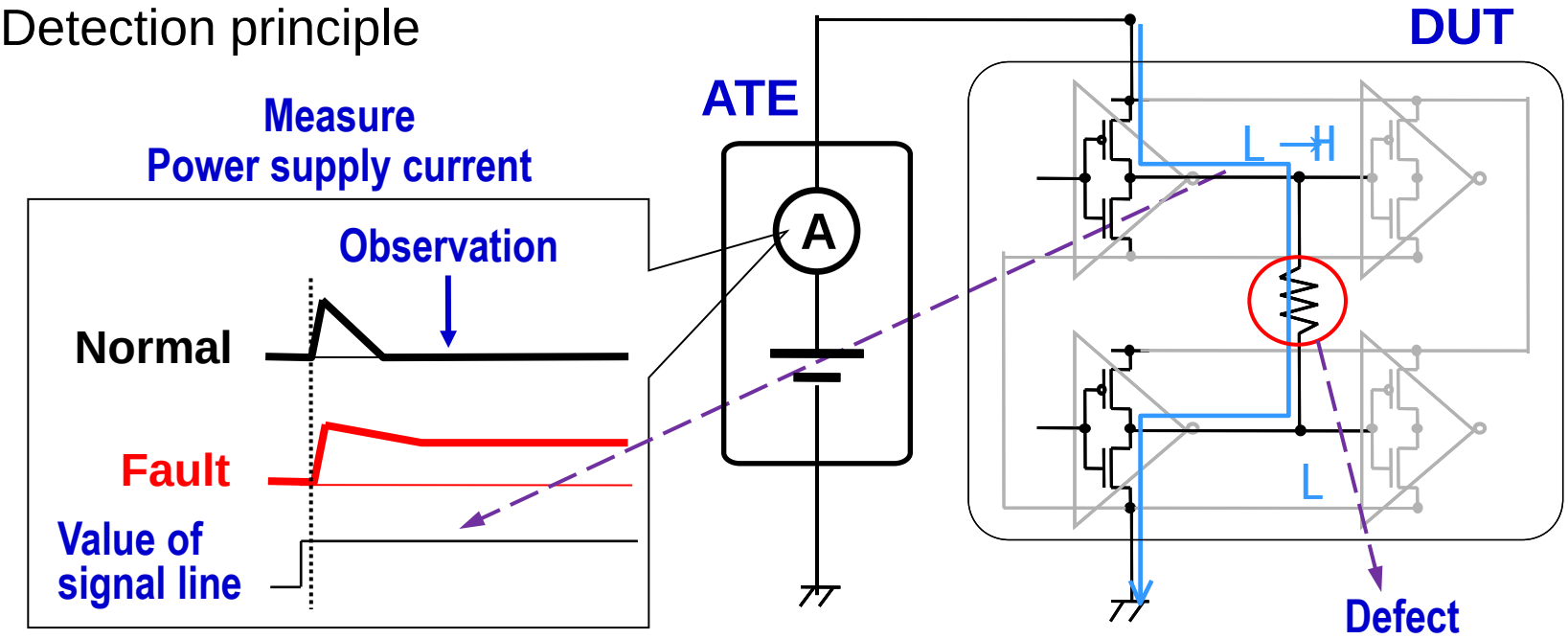
- (a) Purpose of SCAN Design
- (b) Outline of SCAN Design
- (c) Logic Fault Model
- (d) Applied SCAN Design (LBIST etc.)
- *(e) IDDQ Test*

3. DFT for Embedded Memory

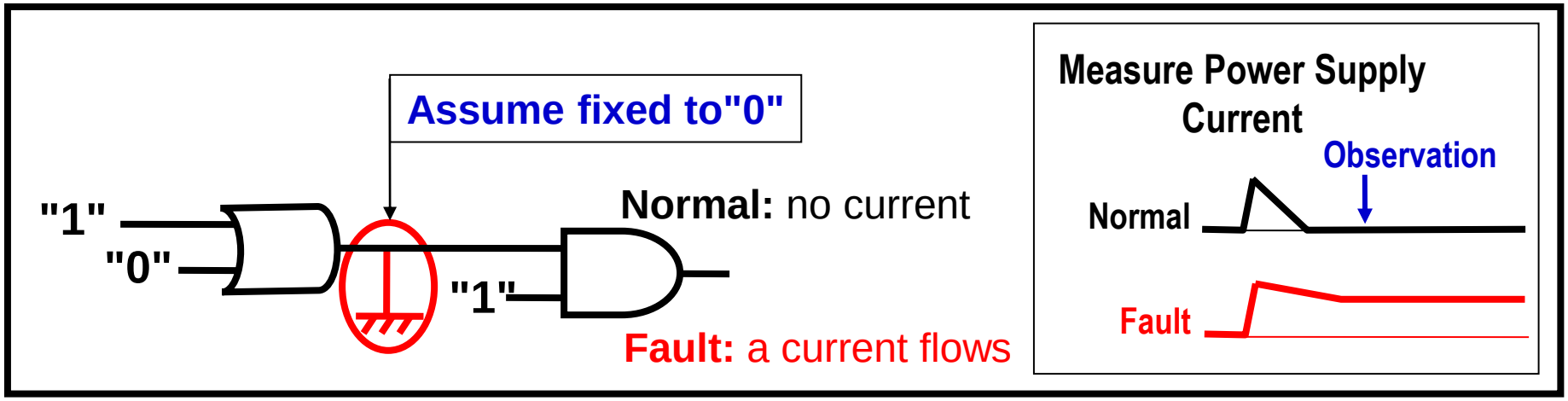
IDDQ Test (Idd at quiescent state)

- Basic concept
  - Quiescent state: Circuit are not switched and inputs are held at static value
  - No current flows through a transistor during operation stop of CMOS
  - Current flow with clock stop and no penetrating current path means failures
  - Fault can be detected by current measurement. Criteria is an analog value of current

- Detection principle

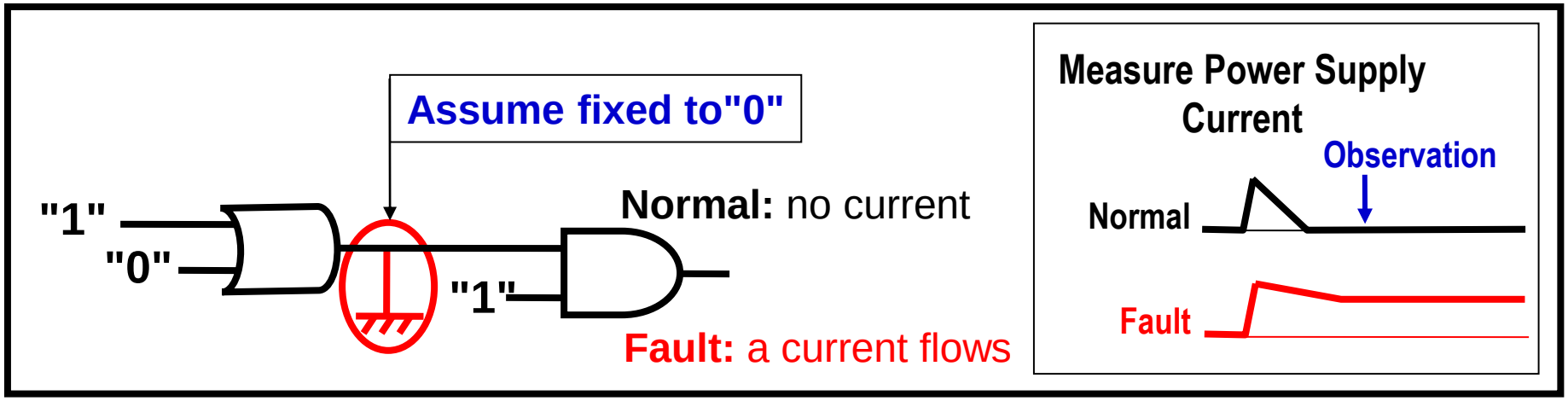


IDDQ Fault Model (1) - IDDQ Toggle Fault Model



- **Fault assumption**
  - ☞ Assuming 1/0 fault to all pins of a cell
- **Fault detection**
  - ☞ If reverse value (1/0) can be set at the position where 0/1 fault is assumed when IDDQ current is measured, the fault can be detected. When the reverse value is set, the power supply current due to the shortage increases and the fault is detected

IDDQ Fault Model (2) - IDDQ Toggle Fault Model

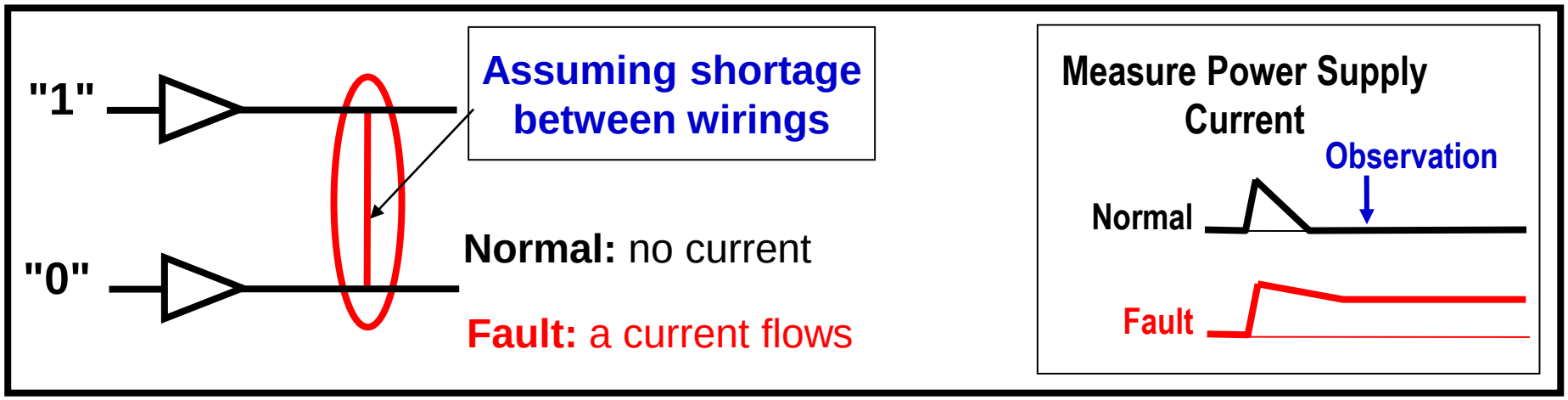


- **Measurement condition**
  - ☞ The value of threshold current

  The value of current is analog which depends on the resistance of shortage and the impedance of gate.

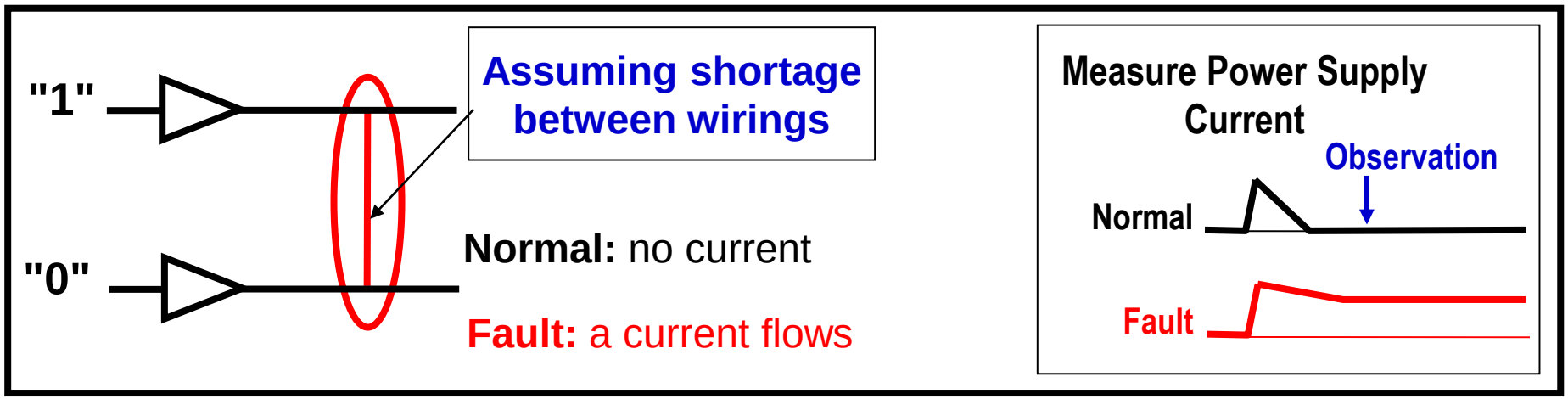
  It is necessary to review the value of current and the evaluation by measuring the actual chip to determine the threshold of current to be treated as a fault

IDDQ Fault Model (3) - IDDQ Bridge Fault Model



- **Fault assumption**
  - ☞ Assuming a fault of shortage between wirings which have a probability. Extract a pair of adjacent wirings which have a probability of shortage fault from the actual layout
- **Fault detection**
  - ☞ If a pair of wirings can be biased such as 1/0 reversely, the fault can be detected when IDDQ current is measured. (When the reverse values are set, the power supply current due to the shortage increases and the fault is detected)

IDDQ Fault Model (4) - IDDQ Bridge Fault Model



- **Measurement condition**
  - ☞ The value of threshold current

  The value of current is analog which depends on the resistance of shortage and the impedance of gate.

  It is necessary to review the value of current and the evaluation by measuring the actual chip to determine the threshold of current to be treated as a fault

IDDQ Test

- **Advantage**
  - Simple and direct test that can be identified physical defects.
  - Area overhead is low
  - Test pattern generation is fast.
  - Detect faults that other test like stuck-at fault could not (bridge, gate-oxide defect, ...)
- **Disadvantage**
  - Time consuming (measuring the current is much slower than measuring the voltage) and expensive.
  - As LSI shrink, the leakage current becomes much higher and less predictable. The noise current must also be dealt with to ensure the quality of the test application.

IDDQ is a supplement test to improve the coverage with small number of test pattern.

Target Fault Coverage (Guide)

- **Stuck-at Fault assumption (DC test)**
  - ☞ ≥ 97% with IDDQ test
  - ☞ ≥ 98% without IDDQ test
- **Transition Fault (AC test)**
  - ☞ ≥ 80% At-Speed
- **IDDQ Bridge Failure/Toggle Failure**
  - ☞ ≥ 80%

DFT in Logic Part - Summary

- **Fault Models**
  - Stuck-at Fault, Transition Fault, Path Delay Fault
  - Toggle Fault Model, Bridge Fault Model
- **SCAN design – MUX SCAN**
  - Conversion of sequential circuit to combinational circuit, automatic generation of pattern
- **LBIST**
  - Built-in input pattern generator and output pattern compression circuit
  - Advantage: applied to Large-scale circuit pattern data size is almost 0, can apply At-Speed test
  - Disadvantage: severe design restriction prohibition of undefined status, medium overhead of circuit
- **Compression SCAN**
  - Advantage: High pattern compression ratio and high coverage

Choosing DFT technique in function of products features and usage purpose

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic Parts

3. DFT for Embedded Memory

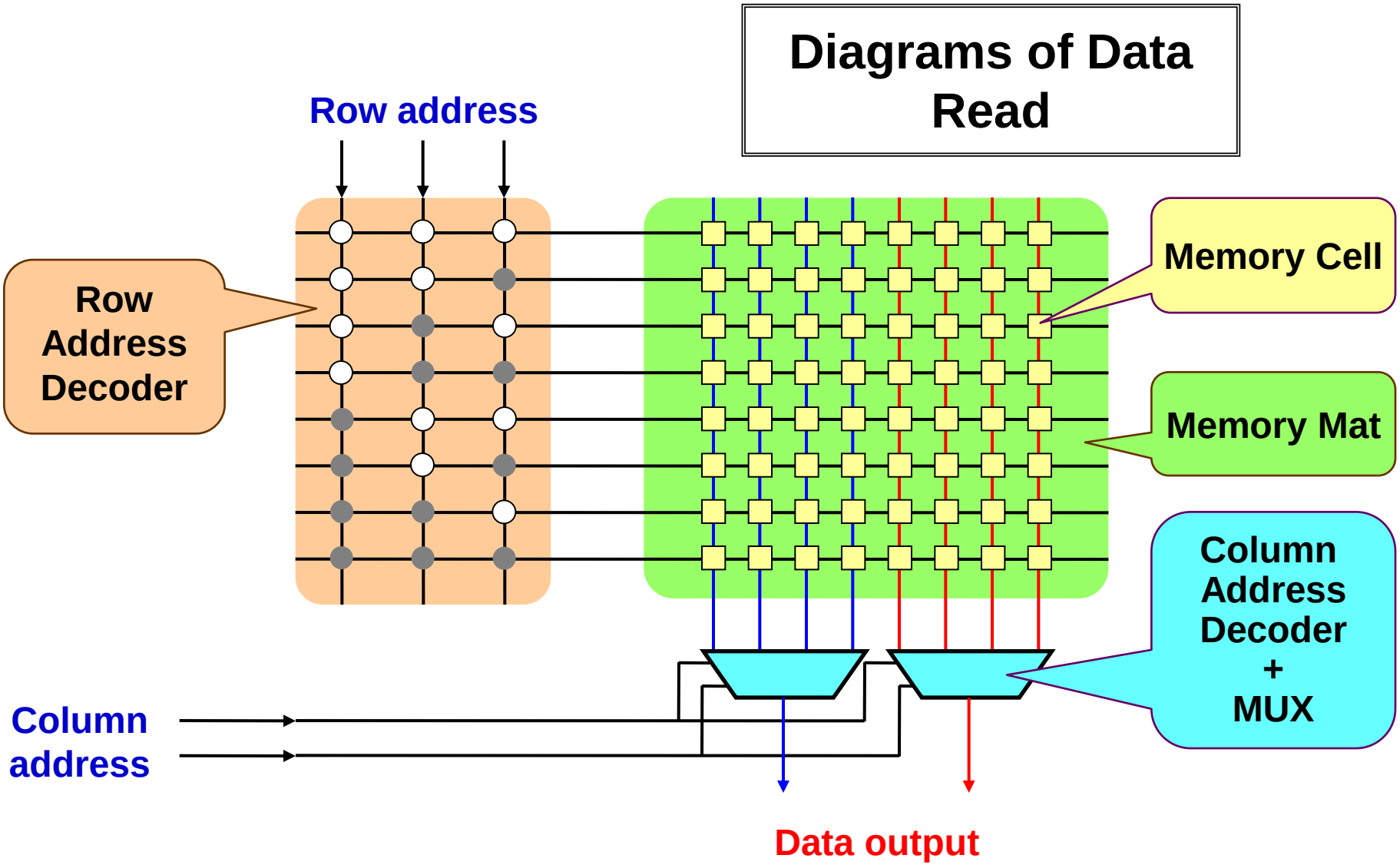
(a) Failure of Memory and Fault Model

(b) Test Algorithms

(c) MBIST (Memory BIST)

(d) Defect Repair and Fuse

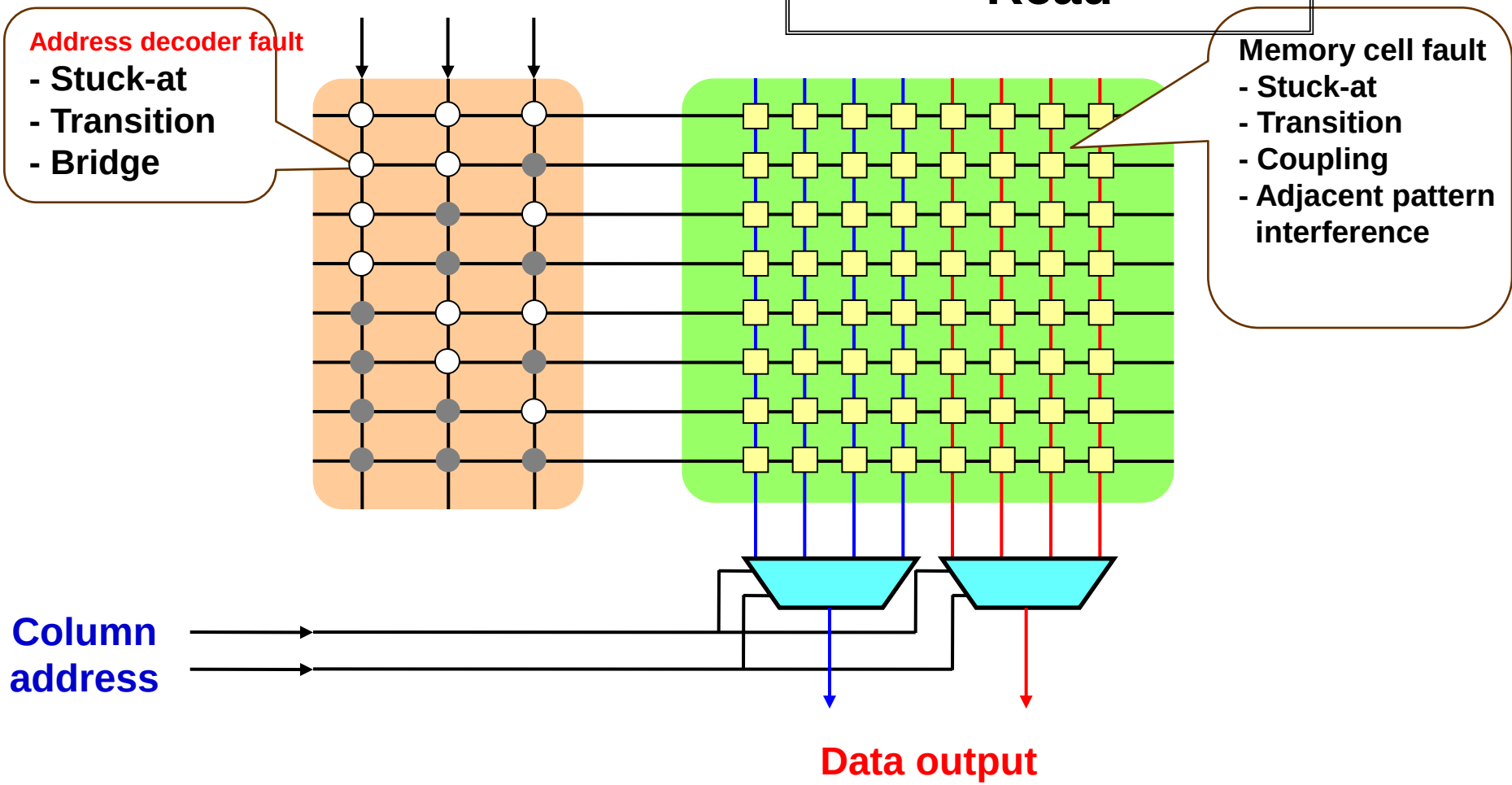
Structure of Memory



Structure of Memory

Manufacturing defect

Diagrams of Data Read



Failure of Address Decoder

Faults of the address decoder are categorized as follow:

- **Stuck-at fault**
  - **Definition:** Cannot decode a specific address
  - **Detect condition:**
    - Need to test all address space (detect all address lines)
    - Need to test with different data to each address (detection of decode failure)
- **Transition fault**
  - **Definition:** After address changes, the decode completion is delayed
  - **Detection condition:** Need to test to make each address line transit with the highest speed (Transition speed of upper address lines with change in simple descending and ascending orders is not sufficient.)
- **Bridge fault**
  - **Definition:** Short-circuit and interferes with the adjacent wiring.
  - **Detection condition:** Need to test adjacent address lines in all the states

Failure of Memory Mat (1)

Faults of the memory mat can be categorized as follow:

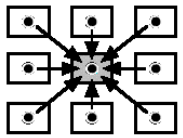
- **Stuck-at fault**
  - **Definition:** The output of memory cells is fixed to either “0” or “1”
  - **Detect condition:**
    - Write “0” to any memory cell, and confirm that “0” is read.
    - Write “1” to any memory cell, and confirm that “1” is read.

- **Transition fault**
  - **Definition:** Output of memory cell can not be changed “0”→”1” and “1”→”0”.
  - **Detect condition:**
    - Write “0” to any memory cell, then write “1” to that cell, and confirm that “1” is read.
    - Write “1” to any memory cell, then write “0” to that cell, and confirm that “0” is read.

Failure of Memory Mat (2)

- **Coupling fault**
  - **Definition:** The “write” operation to a certain memory cell influences the state of another memory cells.
    - The value of influenced objects (effected memory cells) can be same or reverted as the value of subject (certain memory cell).
    - The influenced objects is the adjacent cells.
  - **Detect condition:**
    - Need to test the memory with increment/decrement of address.
    - Detect the influence to lower addresses with the increment test.

    → The test of marching algorithm is effective.
- **Adjacent pattern interference fault**
  - **Definition:** According to the status (pattern) of adjacent memory cells, the value of a specific cell is reversed, does not transit, or changes to a specific value.



  - **Detect condition:**
    - Due to the variety of fault mechanism, it is difficult to test all patterns with single algorithm
    - If the weak point is predicted from memory cell architecture or wafer process, then the test can be focused to the point.

Outline Design For Test (DFT)

1. Importance of DFT

2. DFT for Logic parts

3. DFT for Embedded Memory

- (a) Failure of Memory and Fault Model
- *(b) Test Algorithms*
- (c) MBIST (Memory BIST)
- (d) Defect Repair and Fuse

The Checker Test Technique (Algorithm)

- **Outline**

The name comes from memory test patterns that are made like checker board of “0” and “1”. We need to use checker patterns considering a physical placement of memory cells.

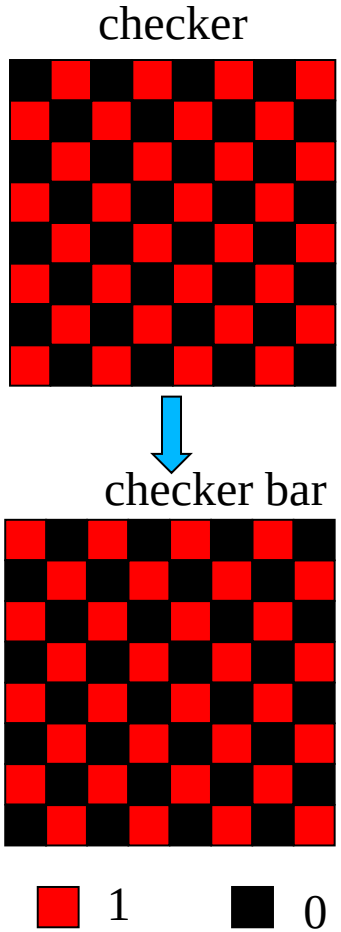
- **A general test procedure**
  - Generate the checker pattern
  - Read and confirm the checker pattern
  - Generate the reverse pattern of checker (checker-bar)
  - Read and confirm the reverse patterns of checker.

The order of test time is proportional to 4 x memory bit number.

- **Feature:**

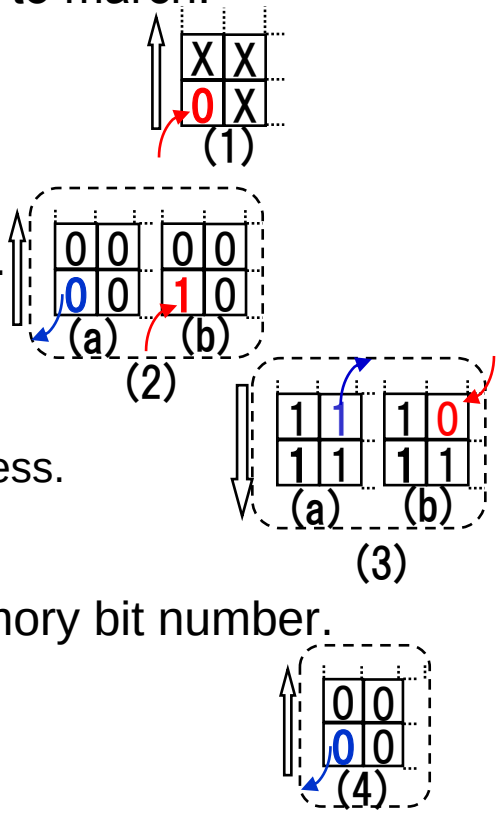
  The order of test time is proportional to 4 x memory bit number

- **Range of fault coverage**
  - Stuck fault
  - The majority of coupling fault
  - A part of adjacent pattern interference fault



Marching Test Technique (Algorithm)

- **Outline**
  During testing, the memory is accessed in increment/decrement address.
  It is called as “marching pattern” because bit patterns look to march.



- **General test procedure**

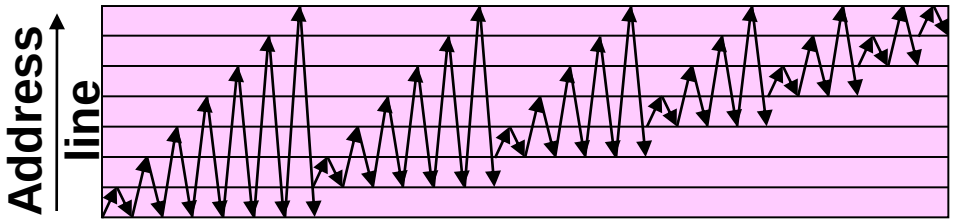
① 0 is written in address ascending order.
② 0 is read in address ascending order, and 1 is written.
③ 1 is read and 0 is written in the descending order of the address.
④ 0 is read in address ascending order.
⑤ 1 is written in address ascending order.
⑥ 1 is read in address ascending order, and 0 is written.
⑦ 0 is read, and one is written in the descending order of the address.
⑧ 1 is read in address ascending order.

- **Feature**
  The test time (read/write frequency) is proportional to memory bit number.
- **Range of fault cover**
  - Stuck fault
  - Transition fault
  - The majority of coupling fault
  - A considerable part of the address decoder test can be covered by devising the address transition.

Simple Address Decoder Test Technique (Algorithm)

- **Outline**
  The transition of the address line is positively generated in the direction of Row and Column.

- **Example of test procedure**
  Address transition of Ping-Pong type is shown as follows, but not a full address transition. Address line transition in only the direction of Row and Column focuses on testing the transition and bridge faults of address decoder.



- **Feature**
  Test time (read/write frequency) is proportional to the number of address lines squared. Then, test time does not so largely increase. It is a practical technique.

- **Range of fault cover**
  - Address decoder test

Outline Design For Test (DFT)

1. Importance of DFT

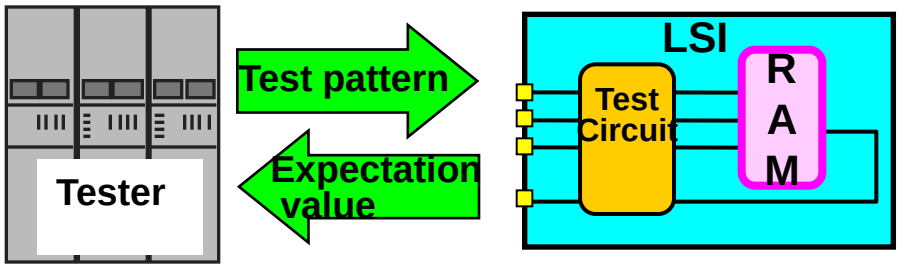
2. DFT for Logic Parts

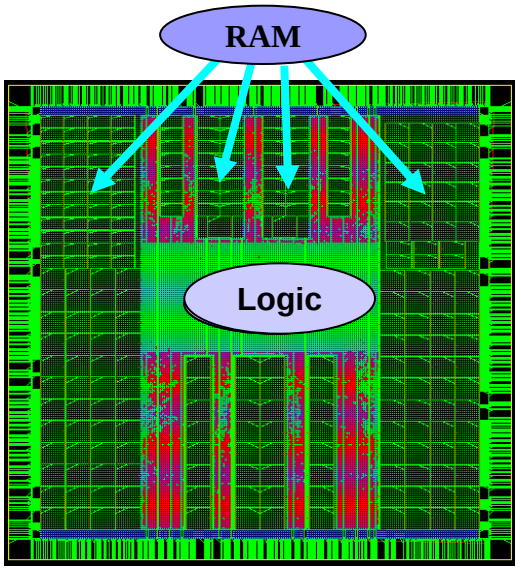
3. DFT for Embedded Memory

- (a) Failure of Memory and Fault Model
- (b) Test Algorithms
- *(c) MBIST (Memory BIST)*
- (d) Defect Repair and Fuse

Memory Test Method - External Access Method

- **Internal memories are accessed from the outside to test.**





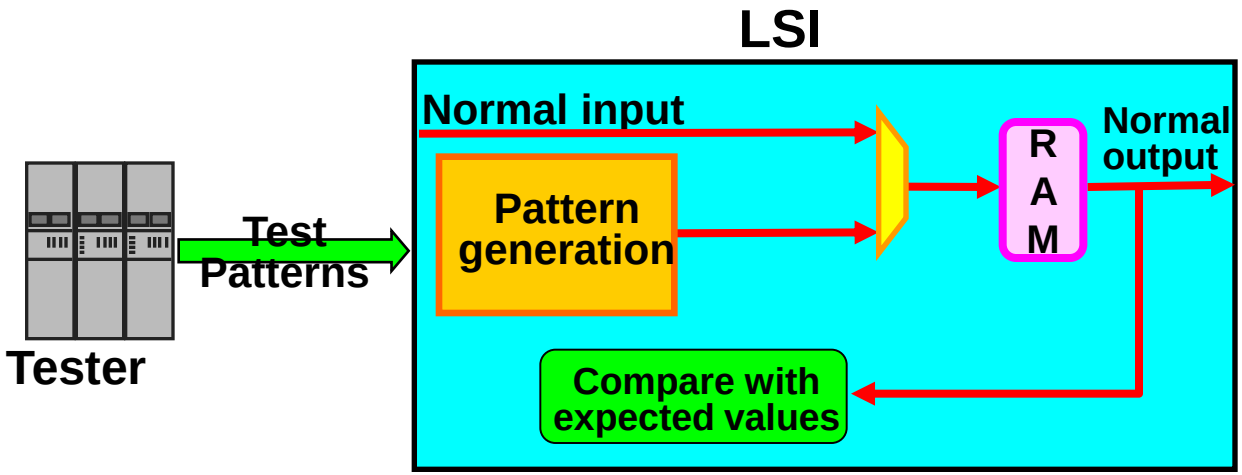
A lot of embedded RAM (250 pieces)

- Test frequency and quality depends on Tester
- Difficulty of high speed test
- Longer test patterns
- Test circuit (external access circuit) is necessary
- Design resource is increased
- Need wirings that run across LSI and waste wiring resources

This makes difficulty to design external access circuit itself.

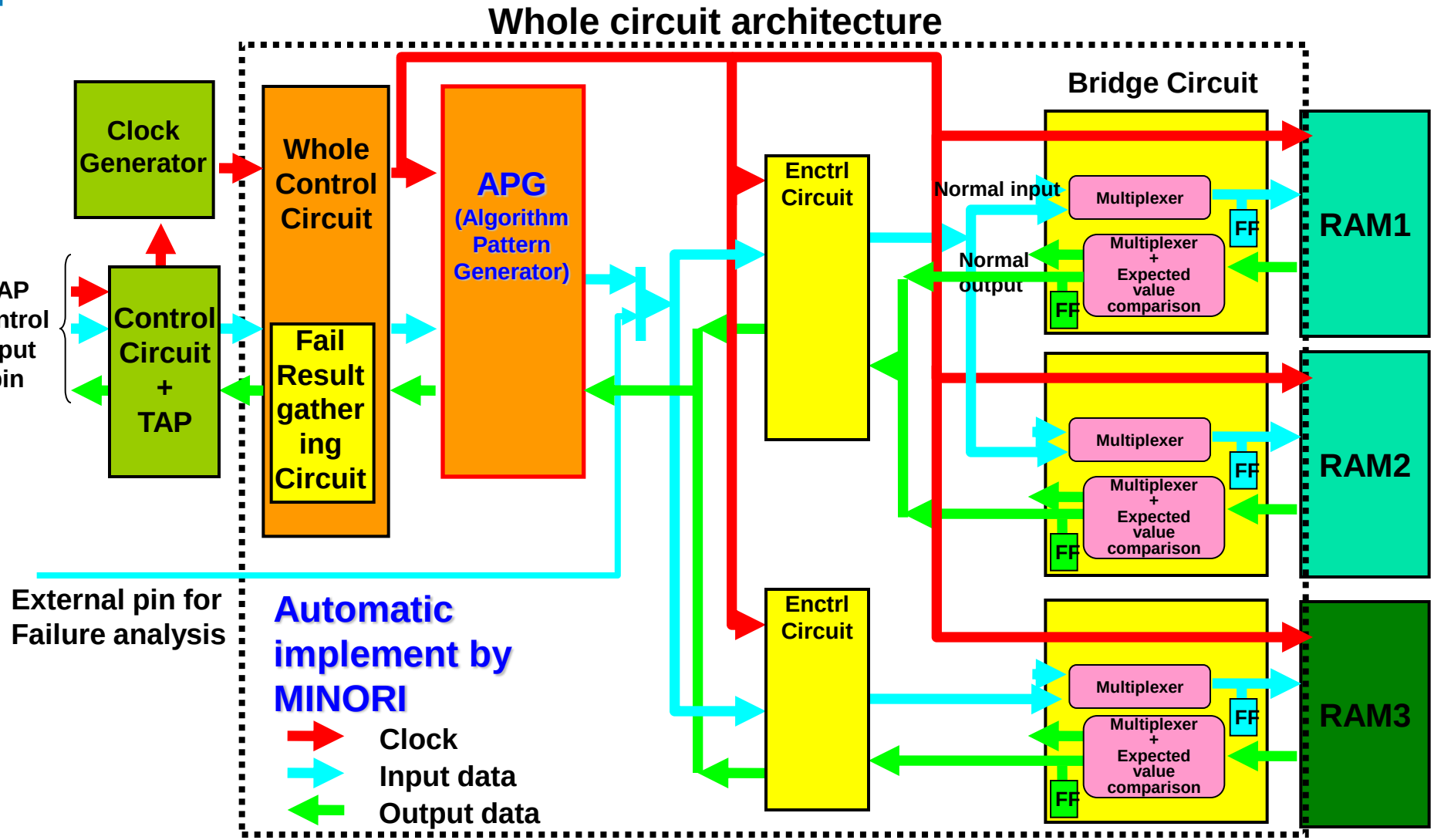
Memory Test Method - MBIST

- **Memory BIST (Built-In Self Test)**

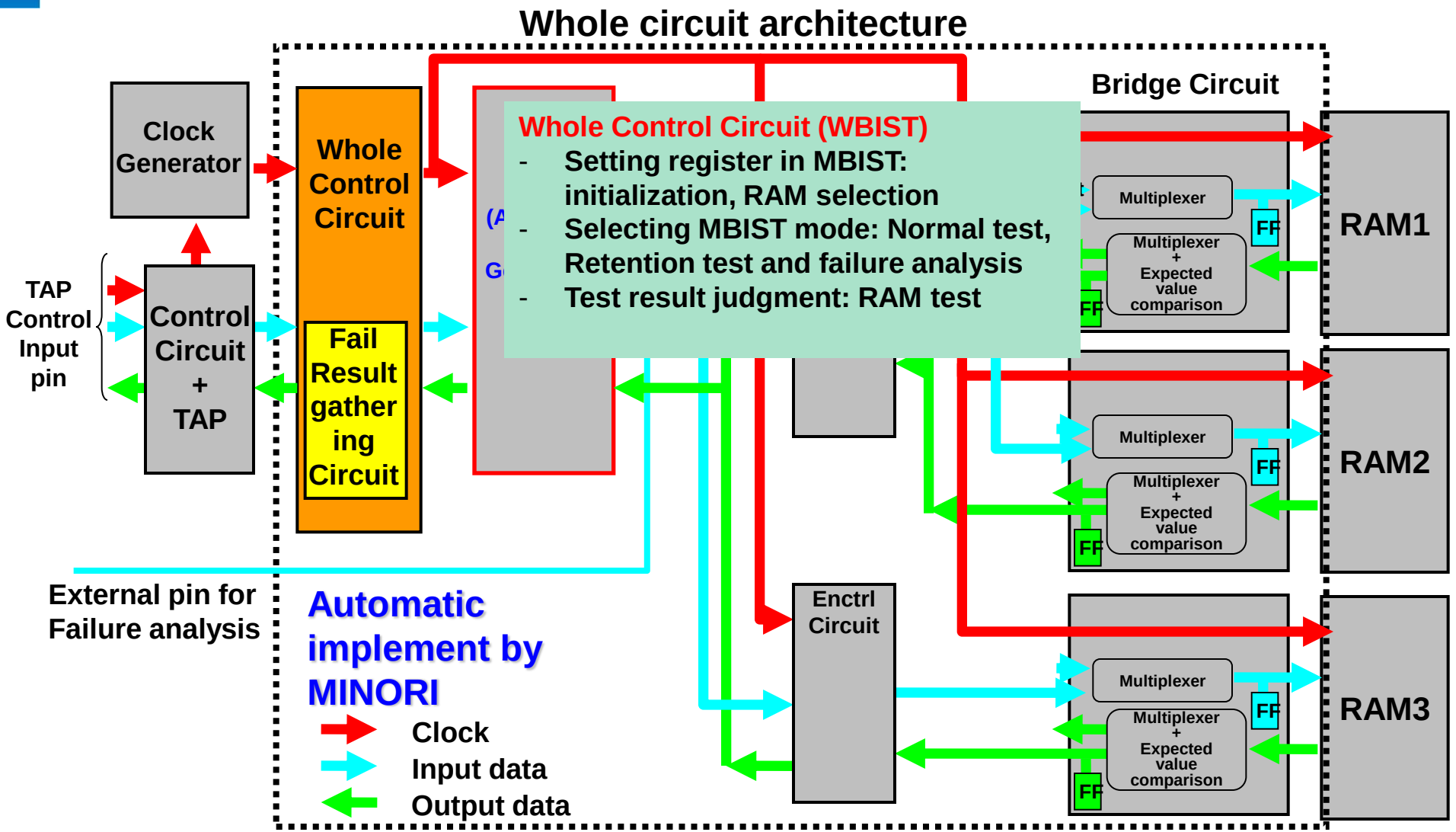


- Generate test patterns within LSI and compare with expectation values
→ Output only “Fail/Pass” info from LSI
- Test frequency is determined within LSI.
- Enable high speed test (Built-in PLL can be used)
- Shorter test patterns (only clock input and instructions are needed during operation)
- Enable to design test circuit (BIST circuit) automatically
- Design cost can be reduced
- Increase circuit size as BIST circuit becomes overhead

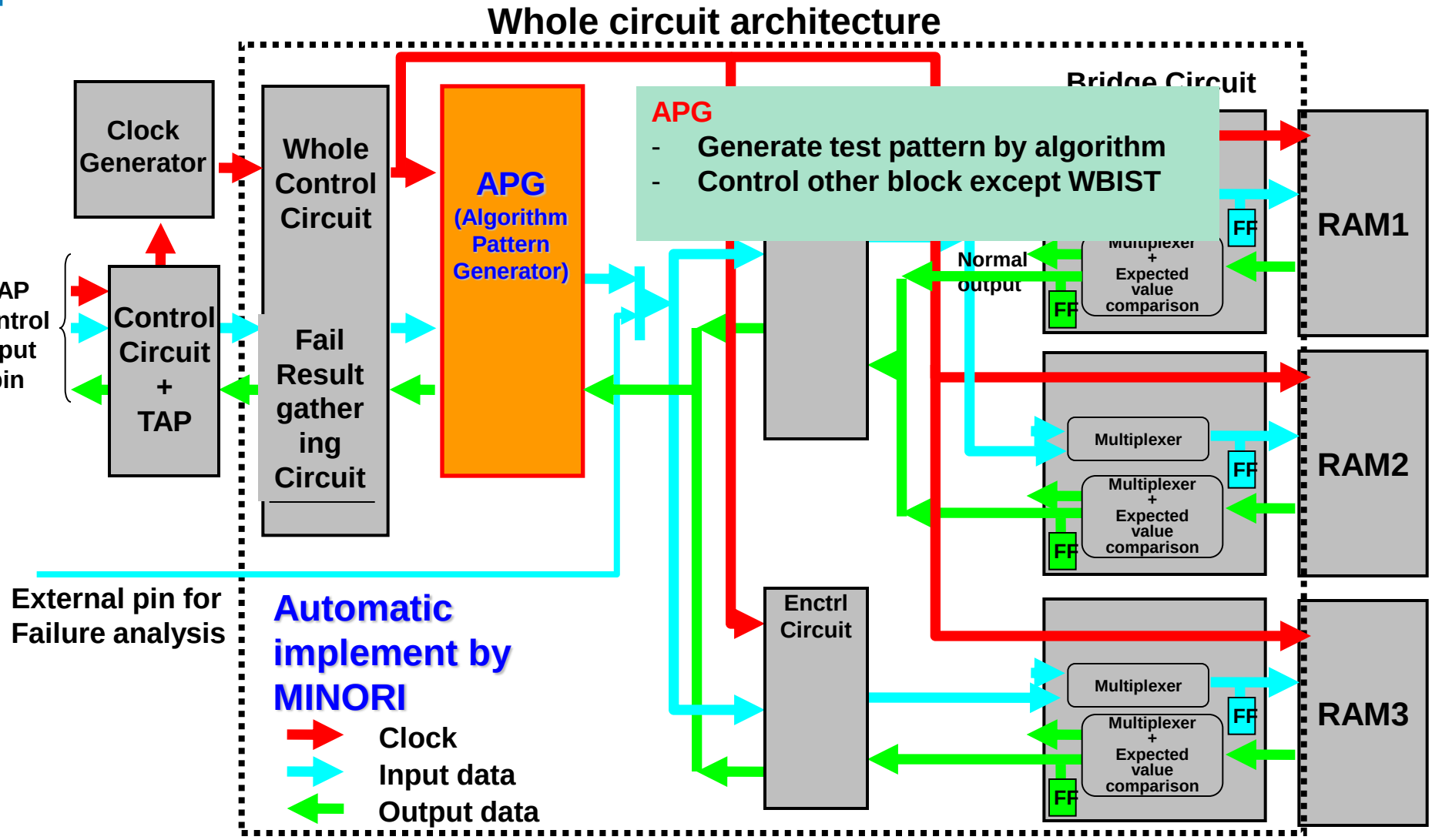
MBIST Circuit (generated by MINORI Tool)



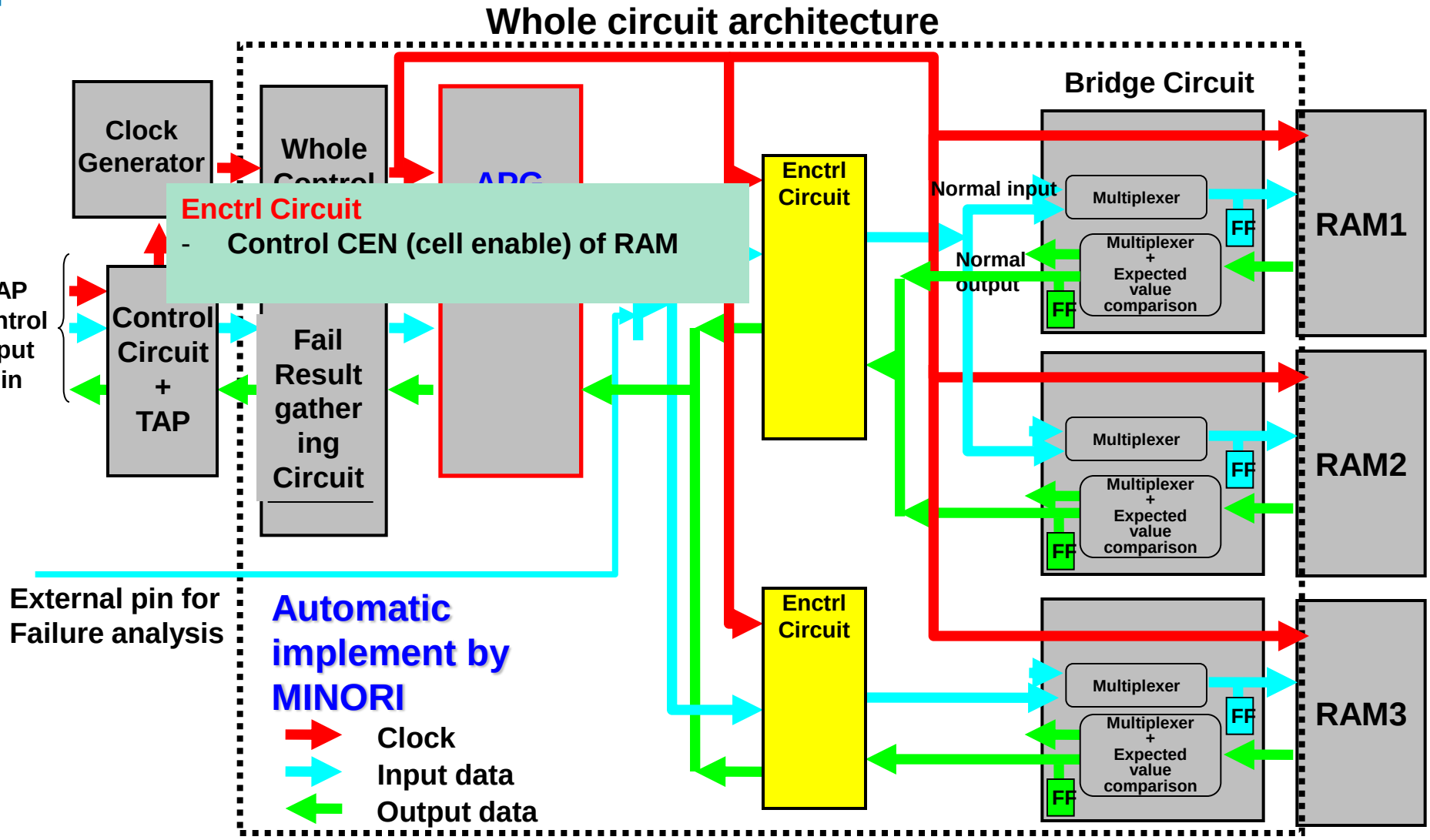
MBIST Circuit (generated by MINORI Tool)



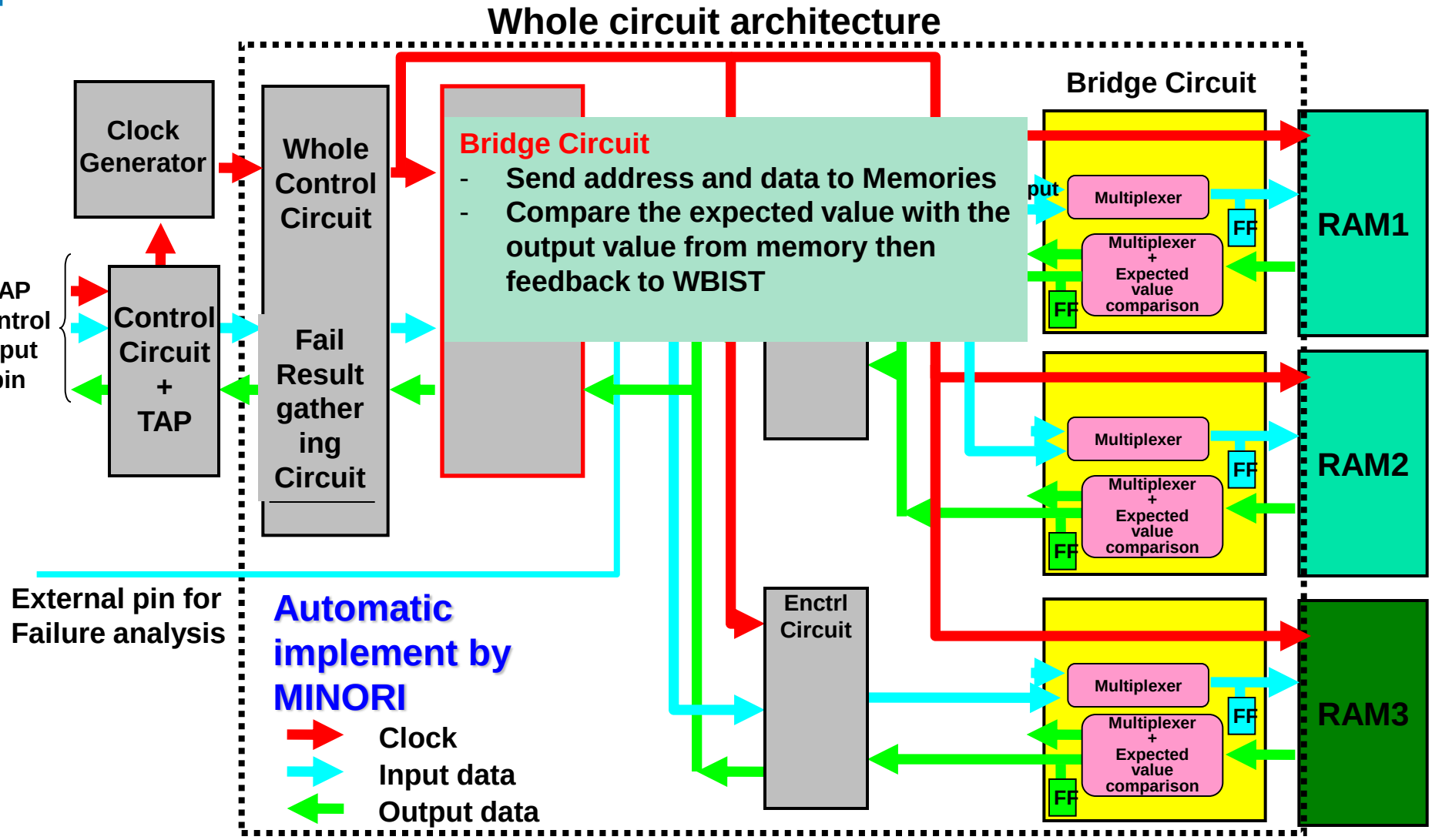
MBIST Circuit (generated by MINORI Tool)



MBIST Circuit (generated by MINORI Tool)



MBIST Circuit (generated by MINORI Tool)



Outline Design For Test (DFT)

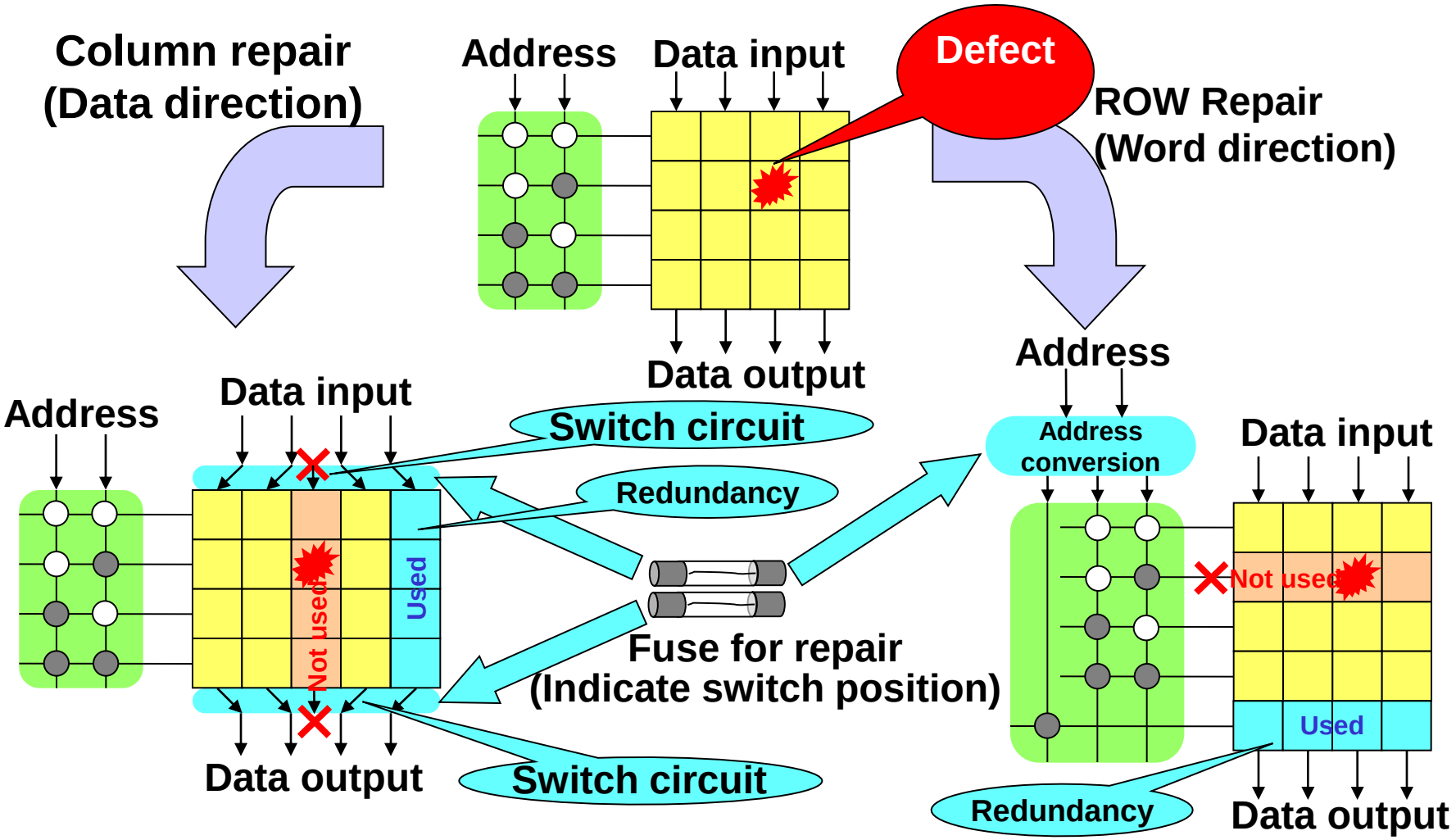
1. Importance of DFT

2. DFT for Logic Parts

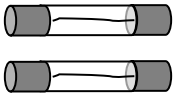
3. DFT for Embedded Memory

- (a) Failure of Memory and Fault Model
- (b) Test Algorithms
- (c) MBIST (Memory BIST)
- *(d) Defect Repair and Fuse*

Memory Defect Repair



Fuse for Repair

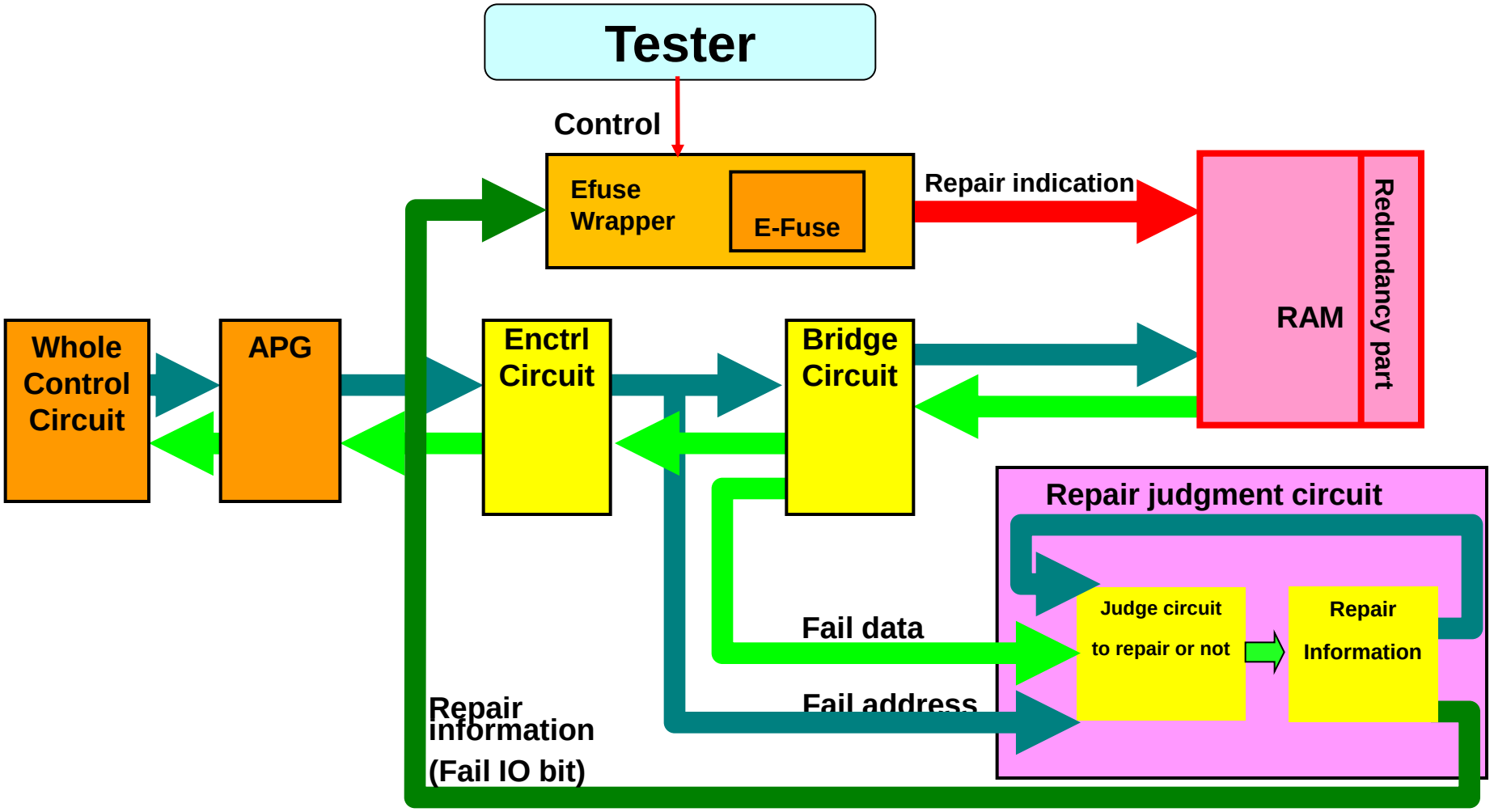
Fuse for Repair 

- **The requirements of fuse for repair**
  - High reliability (not to change, not to degrade with time and temp.)
  - Low writing cost (no extra process, no special equipment, short writing time)
  - Good area efficiency (small area with multiple fuses)

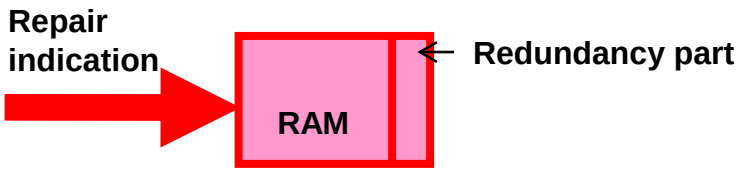
- **LT-Fuse (Laser Trimming-Fuse) (available for RC02, RC03: old technique)**
  - **Structure:** Special wiring exposed in LSI is trimmed by Laser beam.
  - **Feature:** Extra process to trim by “Laser trimmer”, many experiences
  - **Note:** Need to check reliability for packages since fuse is exposed.

- **E-Fuse (Electrical Laser-Fuse) (available for RC04, RC05, T28, T16: current technique, using in RVC)**
  - **Structure:** Blow special wiring in LSI by large current
  - **Feature:** Programmable by tester, no exposed area

Repair for MBIST Example Using E-Fuse



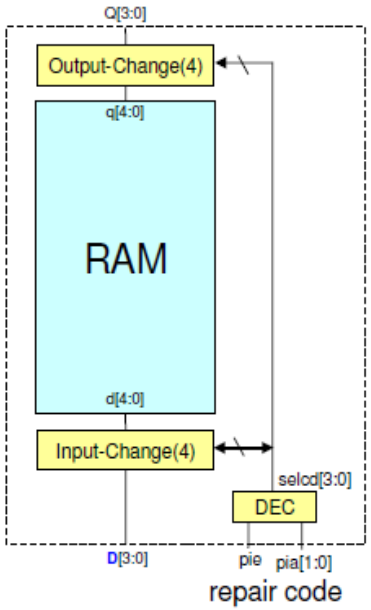
Repair for MBIST Example Using E-Fuse



Ex: A memory has 4 columns and 1 redundant repair column

Memory Wrapper for I/O-Repair

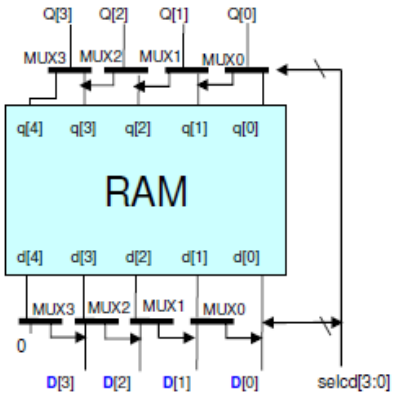
Memory wrapper
(a I/O-repair portion of Collared RAM)



repair code		Output of DEC(4)	Behavior of MUX (*1)			
pie	pia[1:0]	selcd[3:0]	MUX3	MUX2	MUX1	MUX0
0	x	0000	O	O	O	O
1	0	1111	S	S	S	S
1	1	1110	S	S	S	O
1	2	1100	S	S	O	O
1	3	1000	S	O	O	O

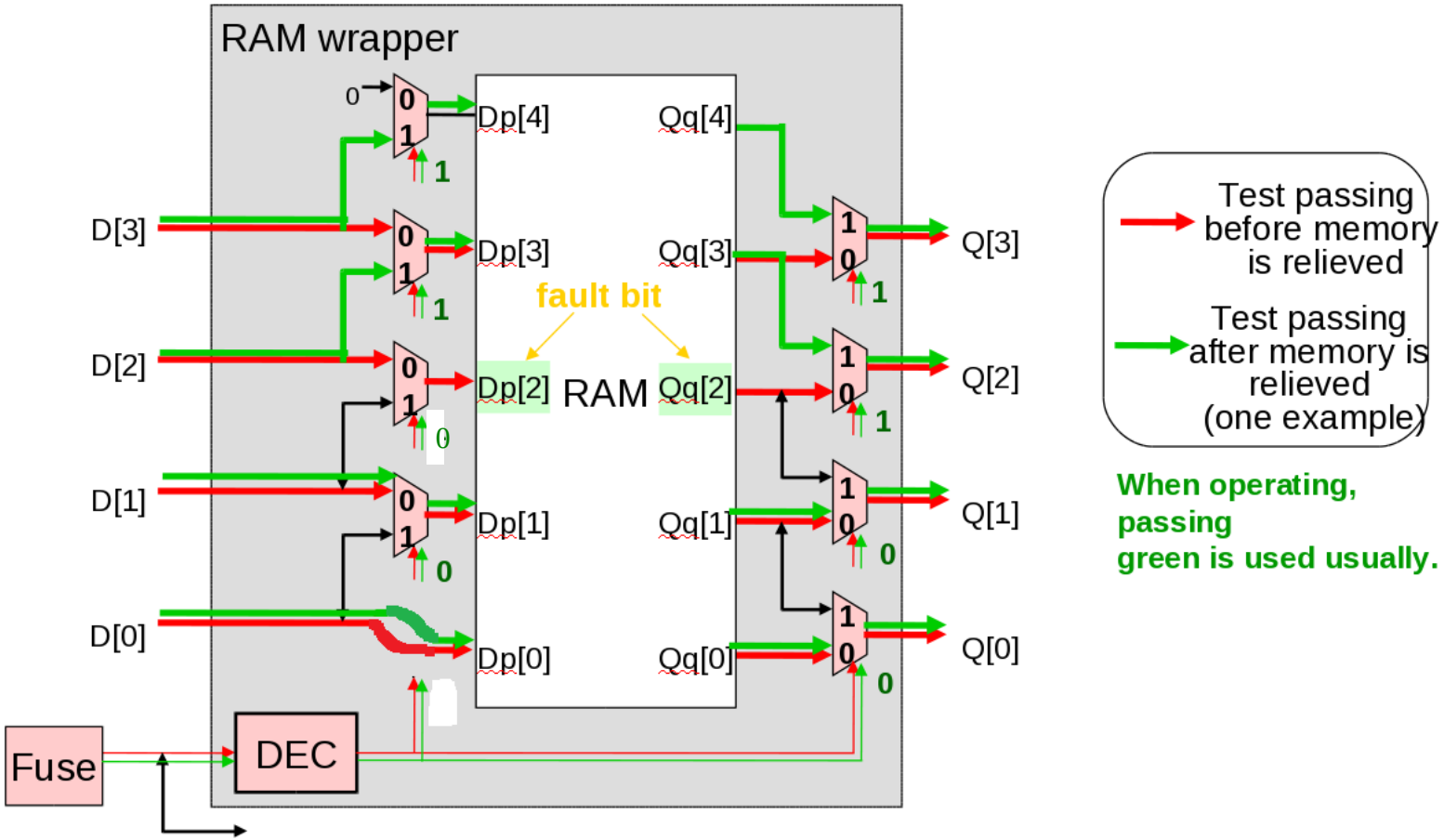
(*1) O: original path
S: Shifted path

Repair indication



Repair for MBIST Example Using E-Fuse

Ex: Fault bit belong to column 2 (Dp[2]-Qq[2]). Output of decode[3:0]=[1,1,0,0]



Summary - DFT Memory

- **Fault model of memory**
  - Stuck-at fault, Transition fault, Coupling fault, Adjacent pattern interference fault
- **Test algorithm**
  - Checker test, Marching test, Address decoder test
- **Memory**
  - Built-in Pattern generator and Expected value comparison circuit
  - Enable At-speed test
  - Arbitrary patterns are available for programmable method.
- **Memory defect repair, fuse**

Power-On Self-Test (1)

Power on

Run Automatically diagnosis

Fail ? N
Y

Can use Car

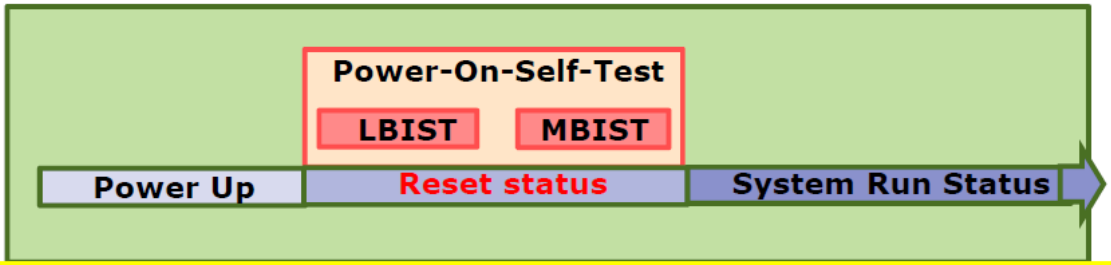
Turn on warning lamp
Can't work engine

When power is turned on, Power-On Self-Test is the diagnostic testing sequence that starting program runs to determine if LSI hardware are working correctly. If it is possible not to work correctly, System should not work or should express the warning.

This diagnostic testing sequence is realized by **LBIST and MBIST**

Power-On Self-Test (2)

LBIST & MBIST as Power-On Self-Test should be done during reset- status



Difficulties about Power-On Self test

- **Period of Power-On-Reset is fixed and short**
- **Guarantee Coverage along the standard**
- **Less Power consumption than system-run-status**

It is Three-sided contest between Time, Coverage. Power-consumption

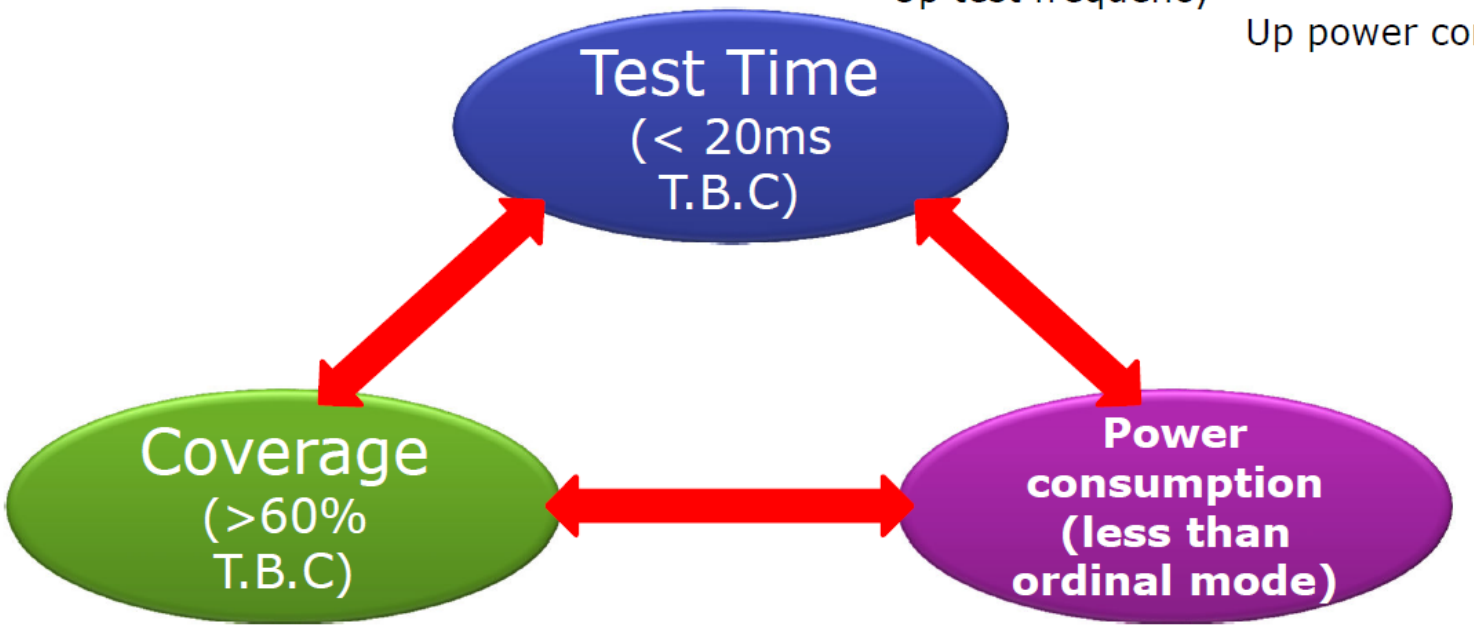
Show Appendix

Need to do Trade-Off design carefully

Power-On Self-Test (3)

It is Three-sided contest
between Time, Coverage., Power-consumption
-> Need to do Trade-Off design carefully

◆ Shorten Test-Time
Reduce seeds → down Cov.
Up test frequency →
Up power consumption



◆ Up Cov.
increase seeds → Test-Time is prolonged
Up toggle-rate → Up power consumption

◆ Down power consumption
Down toggle-rate → down Cov.
Down test frequency →
Test-Time is prolonged

At The End of The Class

- **Need Test design and strategy considering test purpose**
  - Test quality, test cost and test design resource
    - ✓ Need to determine the appropriate target
    - ✓ The relation between field reject rate, yield, area and fault coverage
  - DFT is one of method to improve quality, lower cost and resources
  - Need to manage quantitatively such as detection rate to the circuits where DFT is not applied.

- **DFT - Design for Test**
  - **Advantage:**
    - ✓ Test quality improvement
    - ✓ Can apply At-speed test
    - ✓ Test time reduction, resource reduction to generate test patterns by automatic tool
  - **Disadvantage:**
    - ✓ Need to follow design restriction
    - ✓ Need resource to implement DFT

THANK YOU