Chapter 1

Introduction

Beginning with manufacture of discrete transistors in the late 1940s and early 1950s, the microelectronics industry has evolved into the world's largest manufacturing business. The development of manufacture technology has resulted in fabrication of one billion transistors on a single silicon chip. Table 1.1, from the 2004 SIA Roadmap [1], shows the overall trend of IC technology that the next generations of technology will become. The complexity of semiconductor products has increased exponentially with time. Unfortunately the complexity and difficulty of the design and test increase even faster as the number of components on a chip grows.

1.1 Reliability and Burn-in Testing

As the integration density of VLSI circuits rapidly increases, to maintain current reliability level will be a significant challenge. One of important issues which concern with product reliability of integrated circuits is the early life failure. Except in the infant mortality period, integrated circuit normally have low failure rates and long working lives. Because the failure rate can be extremely low under normal working conditions, it is necessary to take actions to expose the early life failure of the devices without affecting the long term life distribution of the main population.

Table 1.1 Technology Roadmap of Semiconductor Manufacture

Year of Production	2004	2005	2006	2007	2010	2013	2016
MPU Printed Gate Length (nm)	53	45	40	35	25	18	13
MPU Physical Gate Length (nm)	37	32	28	25	18	13	9
Cell area factor [a]	6	6	6	6	6	4	4
Cell array area at production	56.1	56.4	56.7	57.0	57.7	58.1	58.4
Generation at production	1G	2G	2G	4G	8G	32G	64G
Functions per chip (Gbits)	1.07	2.15	2.15	4.29	8.59	34.36	68.72
Chip size at production	93	147	116	183	181	239	238
Gbits/cm at production	1.15	1.46	4.85	2.35	4.75	14.35	28.85
ASIC/Low Power Printed Gate Length (nm)	75	65	53	45	32	22	16
ASIC/Low Power Physical Gate Length (nm)	53	45	37	32	22	16	11
SRAM Cell(6-transistor) Area factor	117.8	115.6	113.7	111.9	107.8	106.7	105.7
Logic Gate (4-transistor) Area factor	320.0	320.0	320.0	320.0	320.0	320.0	320.0
SRAM Cell(6-transistor) Area efficiency	0.63	0.63	0.63	0.63	0.63	0.63	0.63
Logic Gate (4-transistor) Area efficiency	0.50	0.50	0.50	0.50	0.50	0.50	0.50
SRAM Cell(6-transistor) Area w/overhead	1.5	1.2	0.93	0.73	0.22	0.17	0.13
Logic Gate (4-transistor) Area w/overhead	5.2	4.1	3.3	2.6	0.82	0.65	0.51
Transistor density SRAM	393	504	646	827	1718	3532	7208
Transistor density logic	77.2	97.2	122.5	154.3	309	617	1235
Functions per chip at introduction	386	487	614	773	1546	3092	6184
Chip size at introduction	280	280	280	280	280	280	280
Cost performance MPU	138	174	219	276	552	1104	2209
Functions per chip at production	193	243	307	386	773	1546	3092
Lithography Field Size-area	800	800	800	572	572	572	572
Lithography Field Size-length (mm)	32	32	32	26	26	26	26
Lithography Field Size-width (mm)	25	25	25	22	22	22	22
Bulk or epitaxial or SOI wafer	300	300	300	300	300	450	450
Total pads – ASIC high-performance	3600	4000	4200	4400	4800	5400	6000
Signal I/O pads-ASIC high-performance	1800	2000	2100	2200	2400	2700	3000
Power-ground pads-ASIC high-performance	1800	2000	2100	2200	2400	2700	3000
Microprocessor/controller, high-performance	1600	1760	1936	2140	2782	3616	4702
ASIC (high-performance)	2263	2489	2738	3012	4009	5335	7100
# Mask Levels - MPU	25	25	27	27	27	29	29
# Mask Levels - DRAM	24	24	24	24	26	26	26

Manufacturing burn-in testing is an effective methodology for screening out defects contributing to infant mortality. It typically combines electrical stresses with a temperature over a period of time in order to activate the temperature-dependent and voltage-dependent failure mechanisms in relatively short time for improving the quality and reliability of the manufactured chips.

Burn-in can be defined or described in many different ways. According to the scenario of applying stresses, two distinct types of burn-in testing, i.e., static burn-in And dynamic burn-in, are discussed in the AT&T Reliability Manual [2]. In static burn-in, stresses are applied to the circuit with a fixed pattern at an elevated level and the outputs are loaded for maximum power dissipation. In dynamic burn-in, stresses are exercised on the circuit to simulate the real operating environments. Since the external biases or loads do not effectively stress the internal nodes, dynamic burn-in can apply live signals to clock, address, and data lines in order to stress the internal nodes of the circuit. The dynamic burn-in is reported to be a more effective reliability testing for LSI and complex VLSI integrated circuits. Due to relatively long test time, another burn-in procedure, test during burn-in (TDBI), is used for SRAMs and DRAMs. In TDBI, the functional test patterns are usually used as test signals, and the tester monitors the outputs of the circuit under test for reducing the total testing time.

In order to accelerate these failure mechanisms like electromigration, hot-carrier degradation and oxide breakdown to surface, the input patterns of the dynamic burn-in testing need to activate and stress the potentionally defective nodes as more as possible. In order to reduce the cost of test, the test patterns of functional testing and burn-in testing can be generated by using multiple distribution of weighted random pattern generation, i.e., built-in functional testing and built-in burn-in testing are possibly achieved.

1.2 Power Consumption of CMOS Circuits

As the scale of integration improving, it leads to the steady growth of the operating frequency and processing capacity per chip, resulting in increased power dissipation. From traditional two-dimensional design space, i.e., Area/Delay space, designers have moved to a three-dimensional space, i.e., Area/Delay/Power space. During different phases of the design process, different aspects of power dissipation are needed to be considered. For portable communication applications, average power dissipation are concerned due to the limitation of battery life. For packaging and cooling, the maximum average power dissipation should be evaluated in the worst case input conditions of the IC. The maximum instantaneous power dissipation should be considered for reliability requirements.

In order to exploit the potential of reducing power by means of design techniques, an analysis of the expected power dissipation of a particular design alternative has to be made. Similar to other power related design problems like reliability, hot spot detection, electromigration and package selection, power estimation tools are also needed. A large variety of commercial and academic tools from the transistor to the system levels have been developed. For the purpose of directing a designer for low power design, rather than an absolute measure of how much power a particular circuit consumes, an accurate relative power measure between two designs will suffice.

In a digital CMOS circuit, the power dissipation comes from several different sources: dynamic capacitive switching power, P_{dyn}, short circuit power, P_{short}, and leakage power, P_{leak}. They are briefly discussed in the following:

1.2.1 Dynamic capacitive switching power

The dynamic transition current is resulted from the charge and discharge of the node capacitance. In Figure 1.1, an inverter circuit is used to explain the dynamic transition current. The power consumption of dynamic transition current on output node is often formulated as Equation (1-1), when logical value of output node is changed from 0 to 1 and vice versa. It is the major source of power consumption.

$$P_{dyn} = \sum \left(\frac{1}{2} \cdot {V_{dd}}^2 \cdot C_i \cdot f \cdot N_i \right) \qquad (1\text{-}1)$$

where C_i is the node capacitance seen by the gate i, V_{dd} is the supply voltage, f is the clock frequency, and N is the switching activity of the gate i.

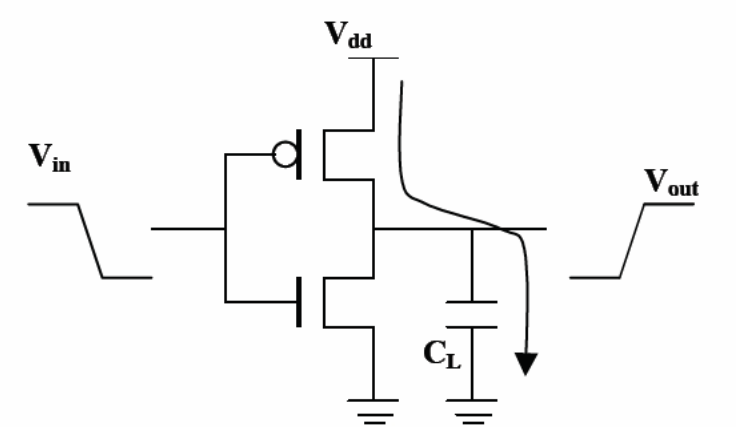


Figure 1.1 Dynamic transition current

1.2.2 Short circuit power

The short-circuit current is the current from V_{dd} to ground at the period that both PMOS and NMOS transistors turn on together during the input signal transitions. As illustrated in Figure 1.2, both PMOS and NMOS transistors are turned on together during the gray circle period of input signal V_{in}, and the short circuit current occurs at that moment. The power consumption of short circuit current can be formulated as Equation (1-2), in which I_{sc} is the mean value of short circuit current.

$$P_{short} = I_{sc} * V_{dd} \qquad (1\text{-}2)$$

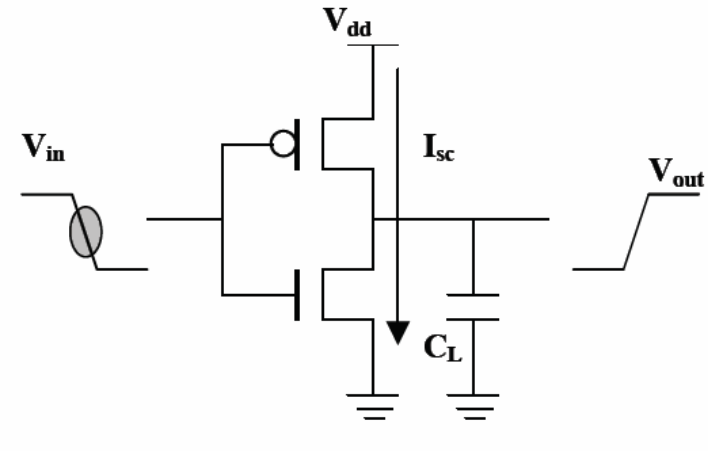


Figure 1.2 Short circuit current

1.2.3 Leakage power

The leakage current is the current from V_{dd} to ground even when the PMOS path or the NMOS path is "OFF". It is often consisted of the leakage current in the reverse biased P-N junction diode between n-well and substrate and sub-threshold current from sub- threshold conduction, as illustrated in Figure 1.3. The power consumption of leakage current can be formulated as Equation (1-3), in which $I_{leakage}$ is the mean value of leakage current.

$$P_{leakage} = I_{leakage} * V_{dd} \qquad (1\text{-}3)$$

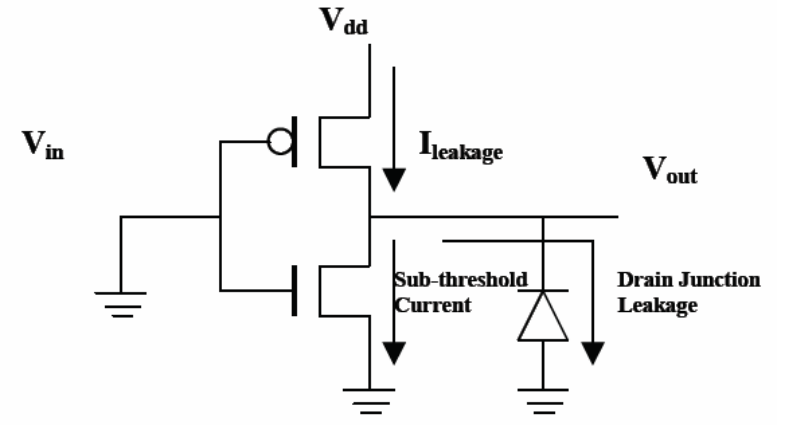


Figure 1.3 Leakage current

1.3 Proposed BIST Burn-in Test Methodology

In this thesis, we propose a burn-in test methodology for an SOC chip. The basic idea is explained with the aid of Figure 1.1, where the circuit-under-test (CUT) is fed with a set of vectors which can cause the maximal switching activities to the CUT. The maximal switching will cause large power dissipation, P_{dyn}, which will raise the temperature of the CUT to achieve the effect of burn-in. The set of vectors are pre-generated and stored on chip. In this way, the CUT is in-situ burn-in-tested without being sent into a high temperature burn-in furnace, and this saves burn-in test cost greatly.

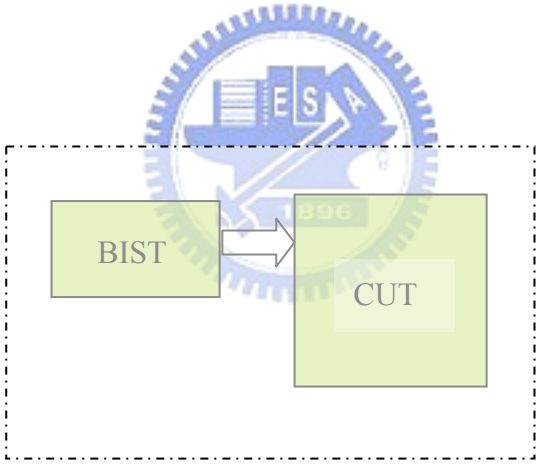


Figure 1.1 The proposed BIST bun-in test methodology

There are several issues which need to be investigated in order to make this scheme workable. Examples of these issues are: (1) how the set of input vectors which cause the maximal switching activities can be generated for a combinational circuit, a sequential circuit, a PLA type of circuit, and a memory type of circuit; (2) how the set of vectors can be stored on chip; (3) the relationship of the power

dissipated with respect to the frequency of the applied vectors. These issues will be studied in this thesis.

1.4 Review of Related Research

In the static logic, we know that the gates can glitch and make multiple transitions. The number of transitions a gate can make is a function of the arrival times of transitions on its inputs, which in turn is dependent on the propagation delays of gates on the different paths from the primary inputs [3][4]. For a CMOS circuit, the power dissipation is mainly due to switching activities charging and discharging capacitances at the internal and the output nodes of a circuit. Hence, all we have to do for accurately producing the maximum power consumption for CMOS circuits is involving exhaustively searching for two consecutive binary input vectors to induce as many switchings as possible [5][6]. That is because the transition depend not only on the present input vector, but also on the present circuit state, which is determined by the previous input vector. A simple and practical wafer burn-in technology which can effectively screens reliability failures of random access memories on a wafer, prior to diesorting[7].

1.5 Outline of Dissertation

This thesis is organized as follows: Chapter 2 presents the research on test generation for combinational circuit and the flow of the ATPG. Chapter 3 presents the research on the test generation for sequential circuit, a PLA type of circuit and a memory type of circuit. Chapter 4 presents the simulation result. Chapter 5 presents the research on how to combine the BIST circuit with CUT and the principle of BIST circuit is also shown in Chapter 5. In Chapter 6, conclusions are given.

Chapter 2

Test Generation for Combinational Circuit

In this section we will illustrate the method to obtain the set of input vectors that activate the maximum switching on the CUT. We first describe it for the combinational circuit and describe the expansion method of the modified circuit and its principle in every details. In the next section, we will describe how to obtain the set of vectors for the sequential circuit, PLA and memory types of circuits.

2.1 General Description of the Test Generation for Combinational Circuit

First, the original circuit is transformed and expanded into the modified circuit, For this modified circuit, each gate has a zero delay and the input pin is doubled since two vectors V_1 and V_2 are to be generated which will make maximal switching for each gate of the circuit. An XOR gate is connected to an added AND gate. The connection between each gate in the modified circuit depends on the time that the gate will produce a transition and the input value that the gate needs to produce a transition. The flowchart of the test generation is shown in Figure 2.1.

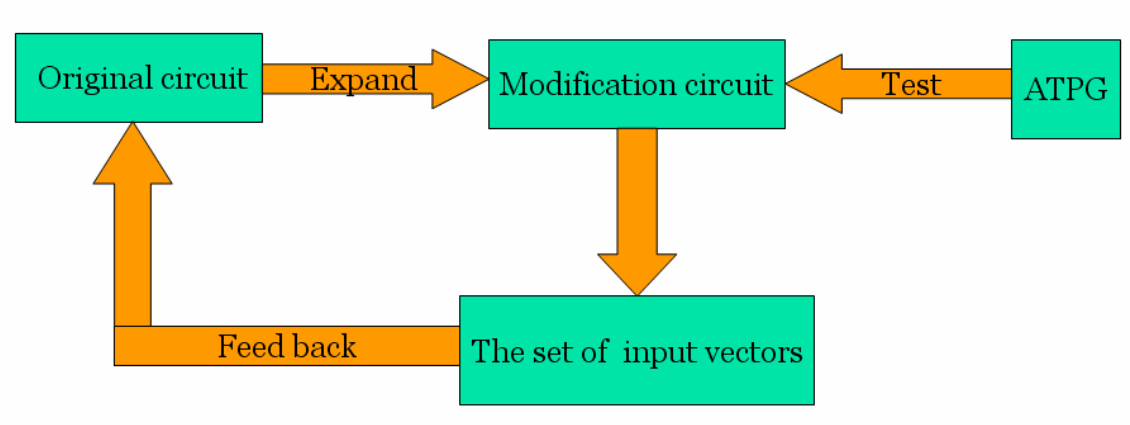


Figure 2.1 The flowchart of the test generation.

Second, after obtaining the modification circuit then we use Auto Test Pattern Generator (ATPG) tool help us to get the present input vector V_1 and next input vector V_2. We put a stuck-zero fault at the added AND gate output and use ATPG to test this fault. If this fault is detected that means the XOR gates connected to added AND gate produce a transition individually.

Thrid, the present input vector V_1 and next input vector V_2 acquired from the ATPG will produce maximum transitions in a period. We design a circuit that will generate these input vectors and then this circuit will be embeded in the original circuit.

2.2 The Method of Expansion Rely on Event-Driven

2.2.1 Event-Driven Simulation

Event-driven simulation is a very effective procedure for discrete-event simulation. It is based on the recognition that any signal change (event) must have a cause, which is also an event. Thus, an event causes new events, which in turn may cause more another events. An event-driven simulator follows the path of events. Consider a circuit at the gate-level. Suppose, all signals are in a steady-state when a new vector is applied to the primary inputs. Some inputs change, causing events. Gates whose inputs now have events are called active and are placed in an activity list. The simulation proceeds by removing a gate from the activity list and evaluating it to determine whether its output has an event. A changing output in turn makes all fanout gates active, which are then added to the activity list. The process of evaluation stops when the activity list becomes empty.

An event-driven simulator only does the necessary work. For logic circuits, which typically very few signals change at a time, this can result in significant savings of computing effort. However, the biggest advantage of this technique is in its ability to simulate any arbitrary delays. This is done by a procedure known as even scheduling.

Suppose the evaluation of an active gate generates an event at its output. If the gate has a switching delay of δ units, then the event should take effect δ time units later. For correctly considering the effects of delays, the simulator distributes the activity list in time. Event scheduling is the procedure of distributing the activity

caused by events over time according to the specified delays.

2.2.2 The Method of Expansion for Modified Modified Circuit

We start by forming a simple example using even-driven to simulate this circuit and deduce the method of expansion step by step:

In Figure 2.2, this circuit has four gates, three inputs and one output. We assume that the delays are different for each gate and are indicated in the symbol.

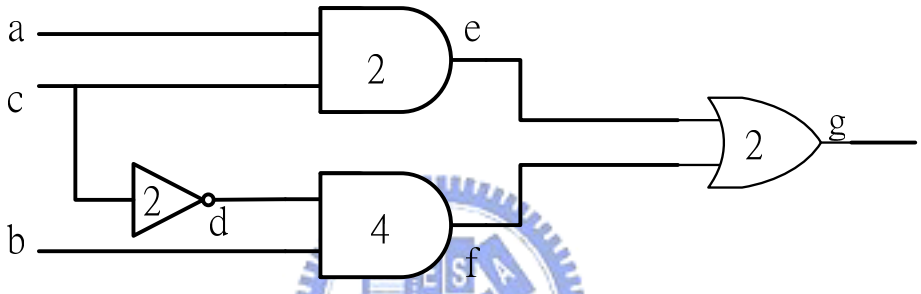


Figure 2.2 Example to illustrate the method of expansion

Assign that the input vector {a,b,c} is changed from {111} to {110}. The signal pass on the path and the circuit nodes get new values as in Figure 2.3.

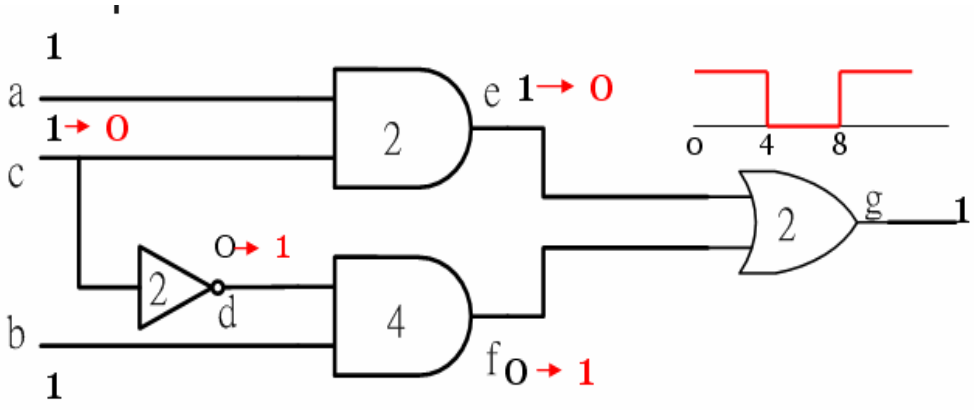


Figure 2.3 The signal passes on the path and the new value of each circuit

node is shown.

Then we use event-driven simulation to simulate this circuit and list its scheduled event and avtivity list. They are shown in Table 2.1.

	scheduled events	Activity list
t=0	*c=0*	*d,e*
t=1		
t=2	*d=1 e=0*	*f ,g*
t=3		
t=4	*g=0*	
t=5		
t=6	*f=1*	*g*
t=7		
t=8	*g=1*	

Table 2.1 The scheduled event and activity under the applied pattern

At time t =0, the input C is changed to 0 and this event will activate the “d” and “e”. Because the delay of gate “d” and “e” are two time unit, new values will be generated at t=2 and become d=1, e=0. This process on derivation on events can be done to construct the whole list. For an example, we can know the gate “e” will have a transition at t=2. The waveform of each gate is shown in Figure 2.4.

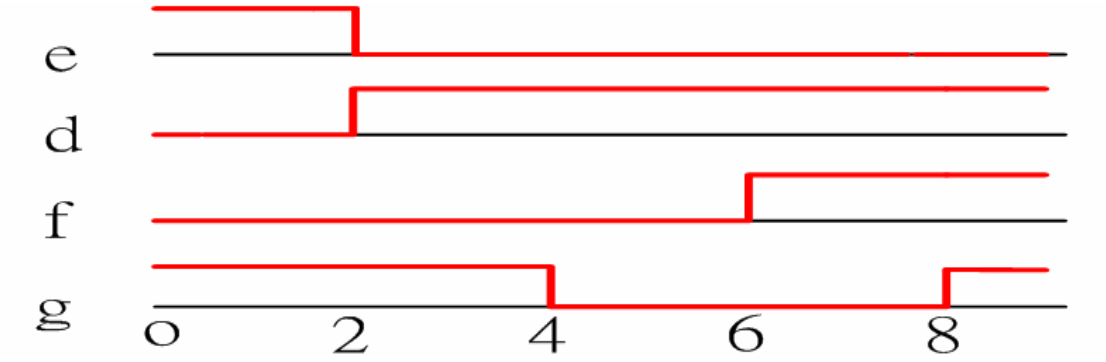


Figure 2.4 The waveform of each gate

For this example, there is another path that will produce activities on some of gates. In Figure 2.5, we show that path and all the times at which a transition may occur. The -∞t means the time that the first vector is fed to the CUT and the value 0t means that the time that the second vector is fed to the CUT.

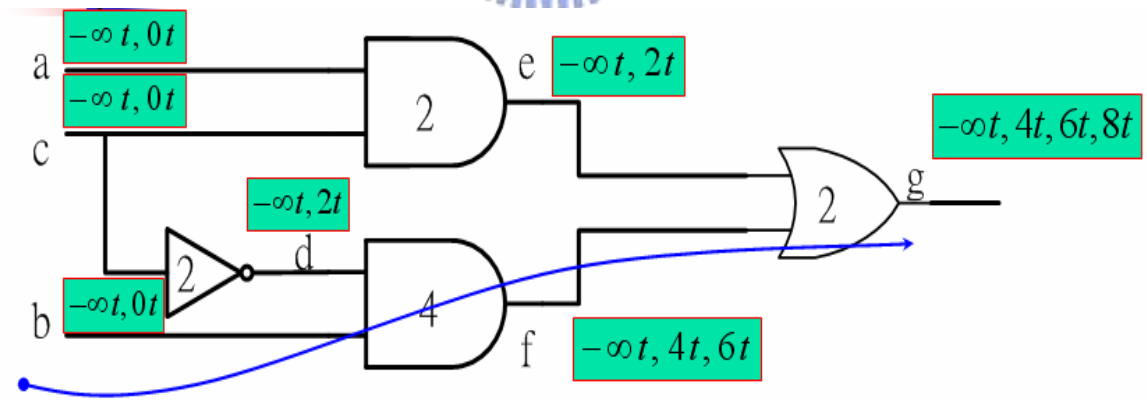


Figure 2.5 Another path and time that activities may occur

After the above information is obtained, we use a two input XOR gate to combine two vicinity of time. If the output of XOR gate is “High”, this means there is a transition in the time domain. Under this example, the whole XOR gates are shown in Figure 2.6.

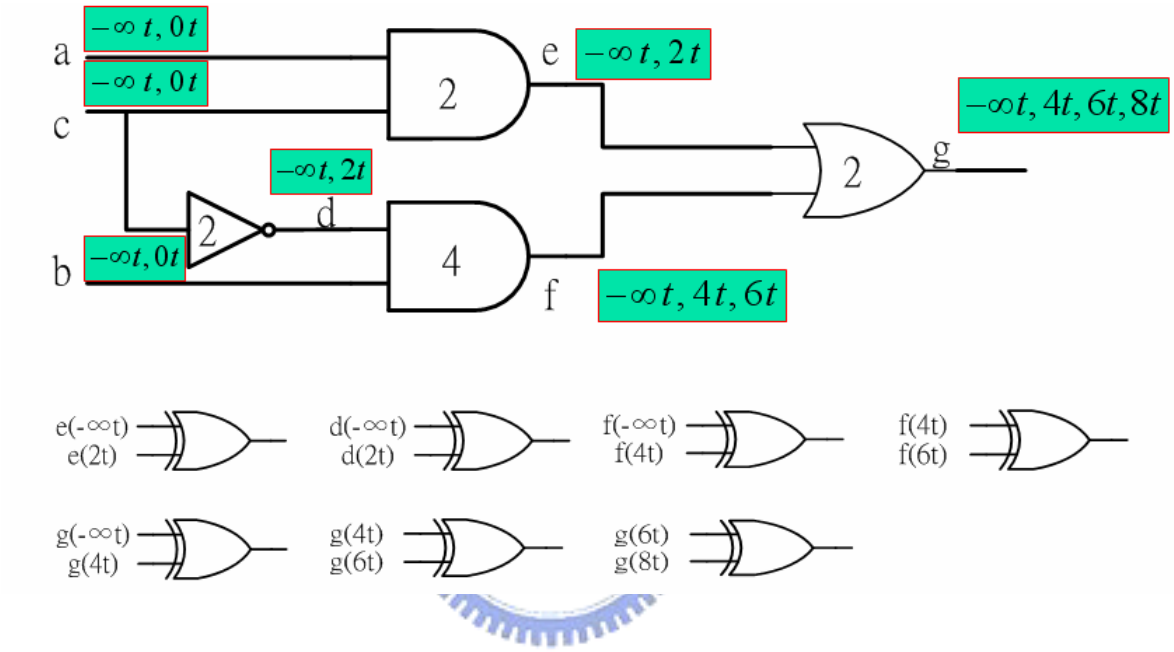


Figure 2.6 The whole XOR gates to be used in this circuit.

Therefore, all we have to do is to find two vectors which can let outputs of these XOR gates become “high” as many as possible. Then we can produce the maximal power in a period. Then, for each time slot, we list every gate that will probably generate new value as shown in Figure 2.7. In Figure 2.8, we connect the input and output of each gate which have the same value to form the semi-finished circuit. Because we already copy every gate that will probably generate new value, each gate delay in Figure 2.8 will change to zero delay. The advantage of this method is that we

can obtain the present input vector V_1 and next input vector V_2 simultaneously and need not worry about the problem of long critical path and large gate delay.

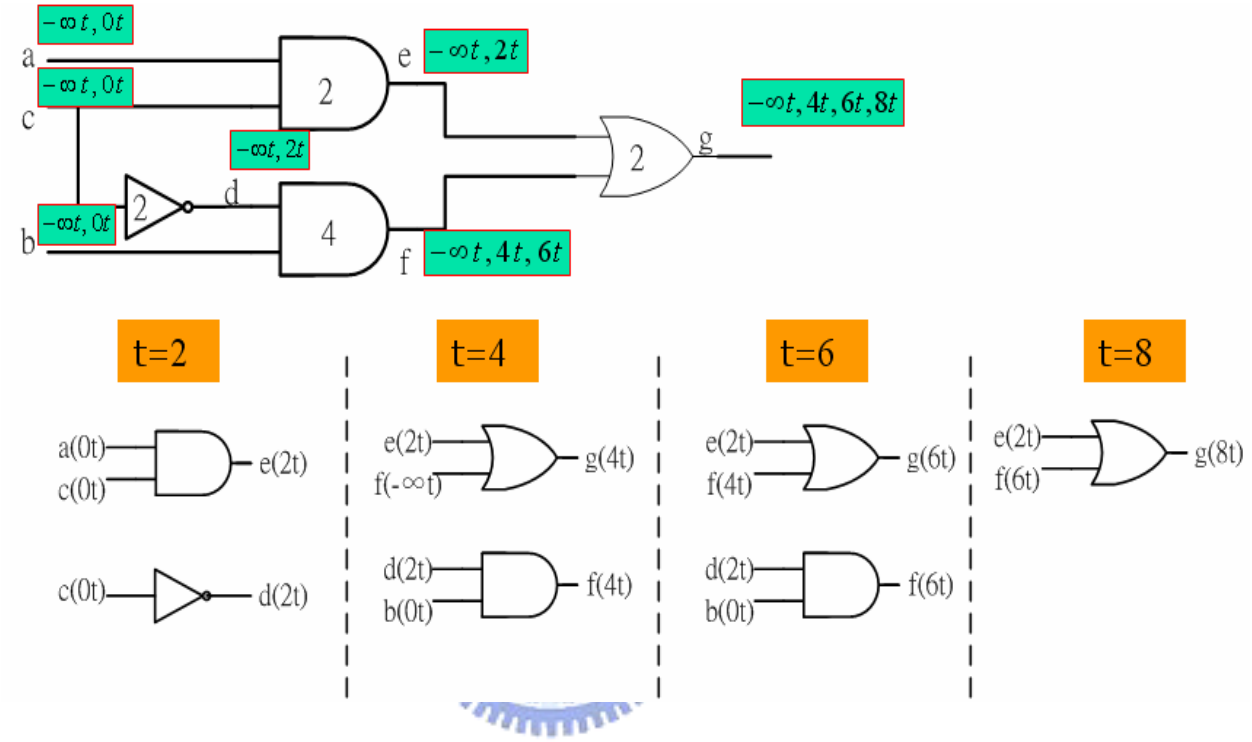


Figure 2.7 The list of every gate that will probably generate new value

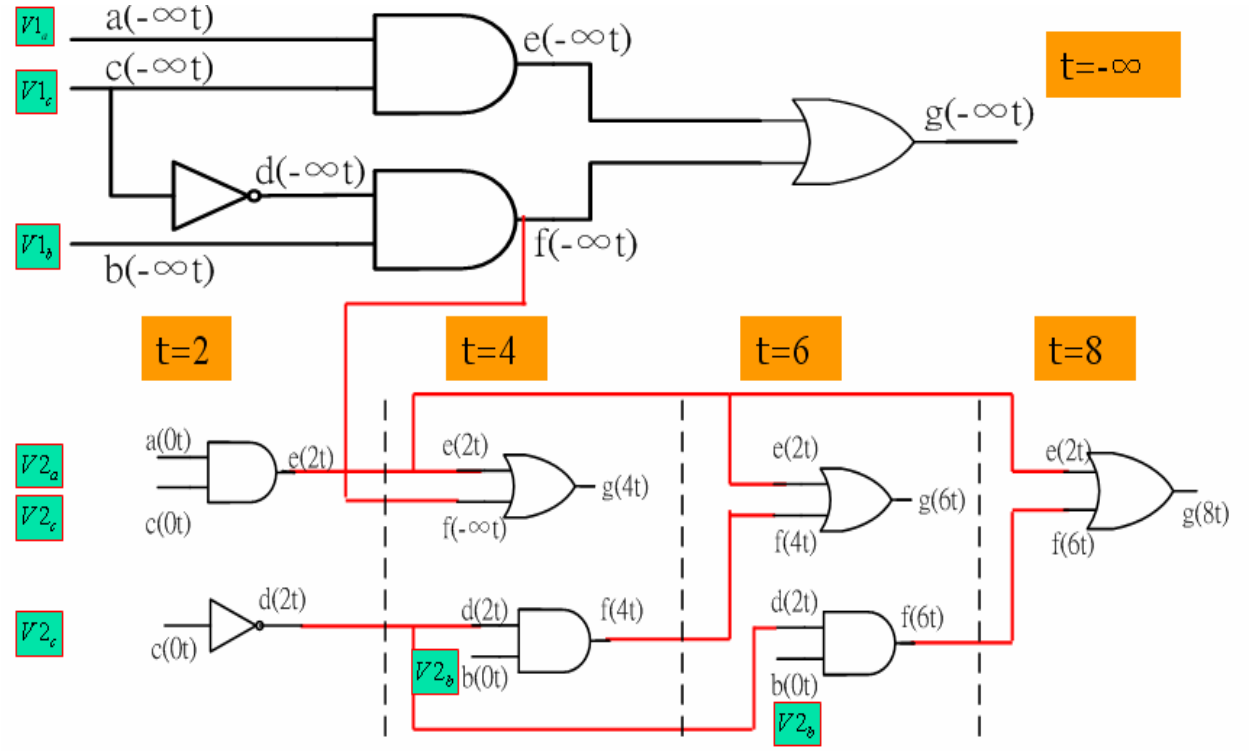


Figure 2.8 The semi-finished circuit

Combining Figure 2.8 and the XOR gates shown of Figure 2.6 will generate the complete circuit as shown in Figure 2.9. An added AND gate is used to collect all the outputs of XOR gate.

After obtaining the expanded circuit, then we use ATPG to help us find the set of input vectors which are to be fed to the CUT to cause as many switchings as possible. So we can put a stuck at zero fault at the output of the additional AND gate. If this fault is detected, that means that every XOR gate connected to the AND gate produce a transition.

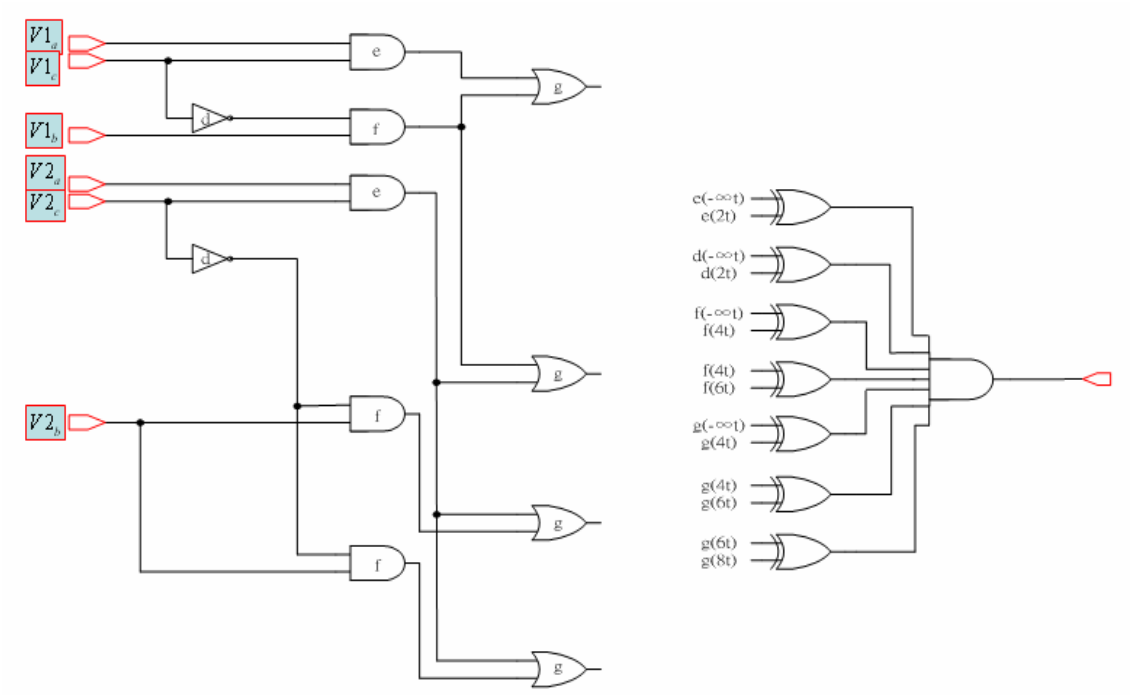


Figure 2.9 The complete expanded circuit for test generation of V_1 and V_2

Similarly, we can construct the modification circuit under the zero delay and unit delay models for each gate. Figure 2.10 and Figure 2.11 show the corresponding modified circuits for the zero delay model and Figure 2.12 and Figure 2.13 show the similar corresponding modified circuits for the unit delay model for an example circuit.

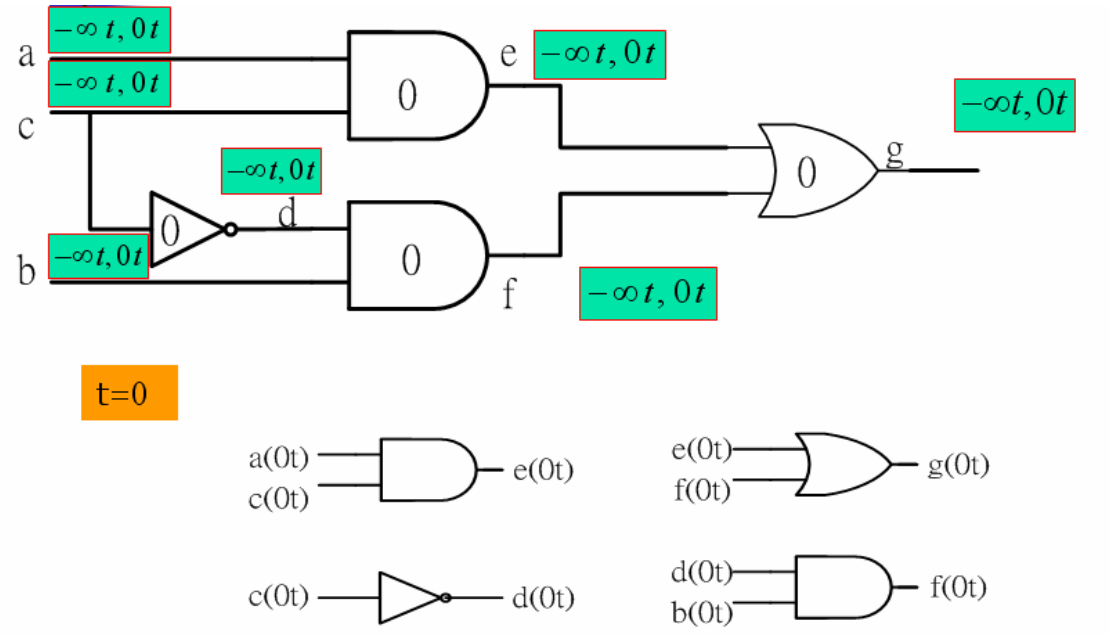


Figure 2.10 The list of every gate that will probably generate new value under zero delay model

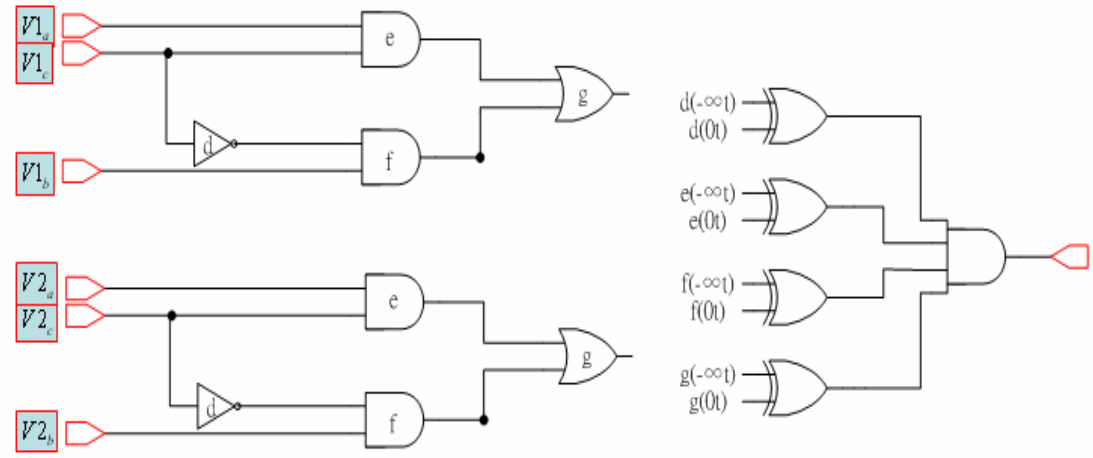


Figure 2.11 The complete circuit under zero delay model

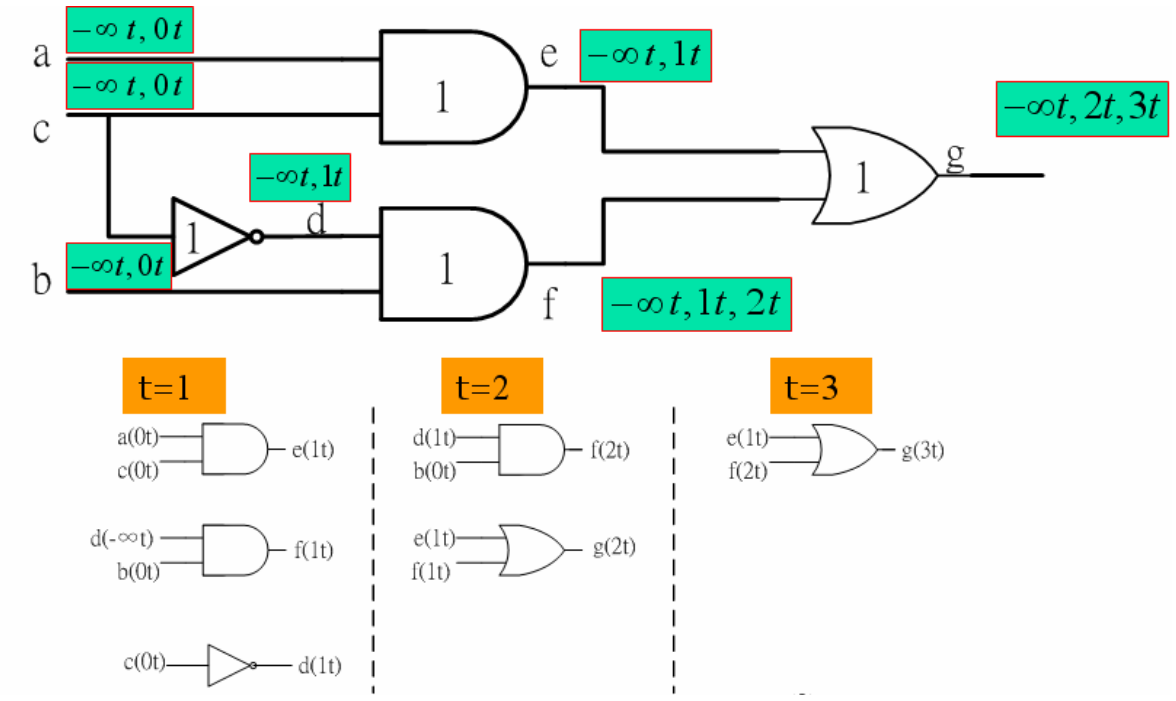


Figure 2.12 The list of every gate that will probably generate new value under unit delay model

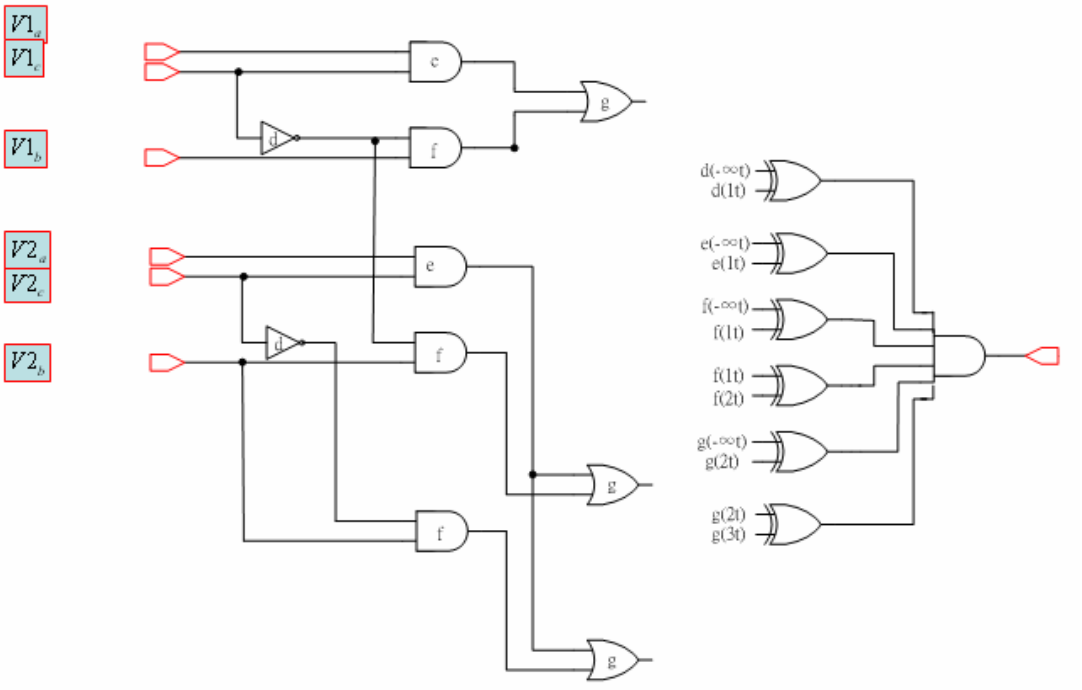


Figure 2.13 The complete circuit under unit delay

2.3 The Principle and Flow of ATPG

2.3.1 The Principle of ATPG for Maximum Transitions

After having the modification circuit, we will use ATPG tool to find vector V_1 and vector V_2 to produce maximal transitions.

Auto test pattern generator (ATPG) is a program that will generate test pattern depending on assigned stuck faults. We use a procedure to apply the ATPG program to generate the test pattern:

Procedure to apply ATPG to generate test patterns for the modified circuit:

Connect all the output of XOR gates to the AND gate and place a stuck-at-zero fault at the output of the AND gate. Set I=0.

Generate the pattern to test the stuck-at-zero fault.

If a pattern is generated, store this pattern and set I=I+1.

If I <= k (an arbitrary number set in advance), go back to Step 2.

If no pattern is generated, remove one XOR gate arbitrarily from the modified circuit and go back to Step 2.

In the above procedure, patterns are generated with all XOR gates connected to the AND gate first. Very often, more than one pattern can be generated and all these patterns are generated with this procedure. If no pattern is found, then remove one XOR gate arbitrarily from the circuit and generate the pattern again. This step is continued until a preset number (*k*) of test patterns are obtained. The vectors so obtained will produce maximal transitions for the circuit.

2.3.2 ATPG for Maximum Cyclic Average Transitions

Since, in practical, a cyclic pattern should be applied to the CUT, i.e., the first pattern should produce transitions with respect to the last pattern when the last pattern is followed by the first pattern to form a cyclic patterns. Since very often, we may not find the cyclic input patterns from the pattern set which cause the maximal transitions for the CUT, we have to resort to the pattern set which cause the secondly or thirdly maximal transitions. Then we use a procedure, for which the sketch map of Figure 2.14 is used to explain, to generate the cyclic input pattern.

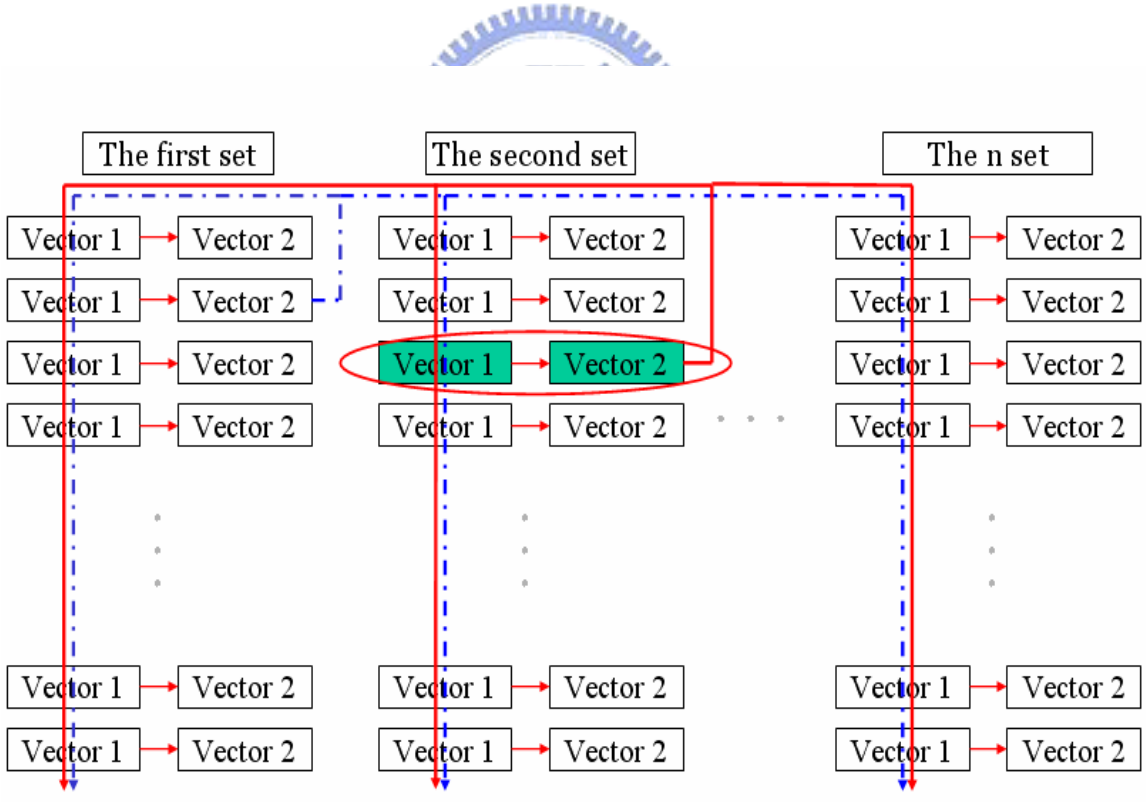


Figure 2.14 The sketch map of searching for generating the cyclic input pattern.

Suppose we start from the second pattern in the first set in Figure 2.14. Use Vector2 of the second pattern as a standard and search for the same Vector1 from the highest order. The process is indicated with the dotted line in Figure 2.14. If we find the same Vector1 in the third pattern which is circled in the second set. In the same way, use Vector2 of this pattern as a standard and search for the same Vector1 from the highest order. The process is described with the solid line in Figure 2.14. When Vector2 is equal to Vector1 of the starting pattern, the search is successful and done.

Procedure to apply ATPG to generate cyclic test patterns for the modified circuit:

Use certain Vector2 of arbitrary pattern in the first set as a standard and search for the same Vector1 from the highest order.

Use the Vector2 of the searched pattern as a standard and search for the same Vector1 from the highest order.

If the Vector2 of the searched pattern is equal to Vector1 of the starting pattern, store these cyclic test patterns.

else go back to Step 2.

Chapter 3

Test Generation for Sequential, PLA and Memory Circuit

In this section, we will describe how to obtain the set of vectors for the sequential circuit, PLA and memory types of circuits.

3.1 Test Generation for Sequential Circuit

The fundamental structure of sequential circuit is shown in Figure 3.1, which is composed of combinational circuit and flip-flops. In order to test the combinational part, we remove flip - flops from the circuit and the pins which are connected to flip-flops become primary inputs and outputs. The circuit becomes that as shown in Figure 3.2. Therefore, we apply the method derived in the previous chapter to generate test patterns for this modified circuit.

The test set up for the sequential circuit bist burn-in will be described in the later chapter 5.

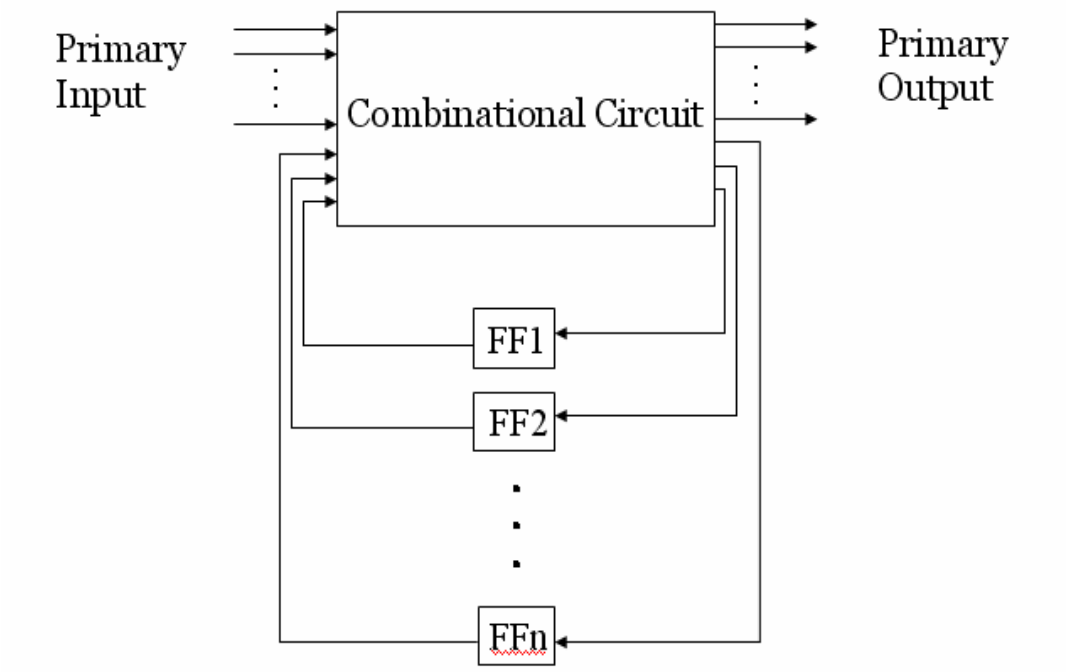


Figure 3.1 The fundamental structure of the sequential circuit

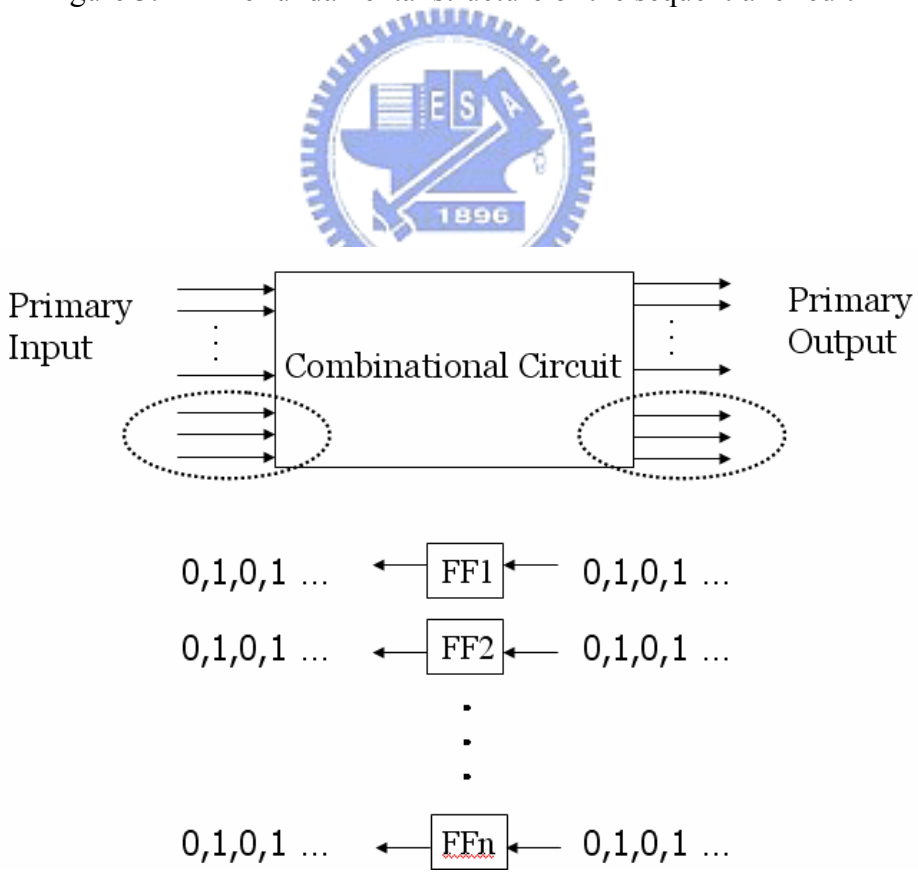


Figure 3.2 The modified combinational circuit under test.

3.2 Test Generation for PLA type of circuit

To burn-in test, a PLA circuit, we have to let the PLA circuit consume the maximal power. To do this, we want to switch as many transistors as possible. Figure 3.3 shows a PLA example. It is easily found that the best pattern to be applied to this PLA is x,y = “00,11,00,11…” which will switch all transistors on the PLA planes in a systematic way.

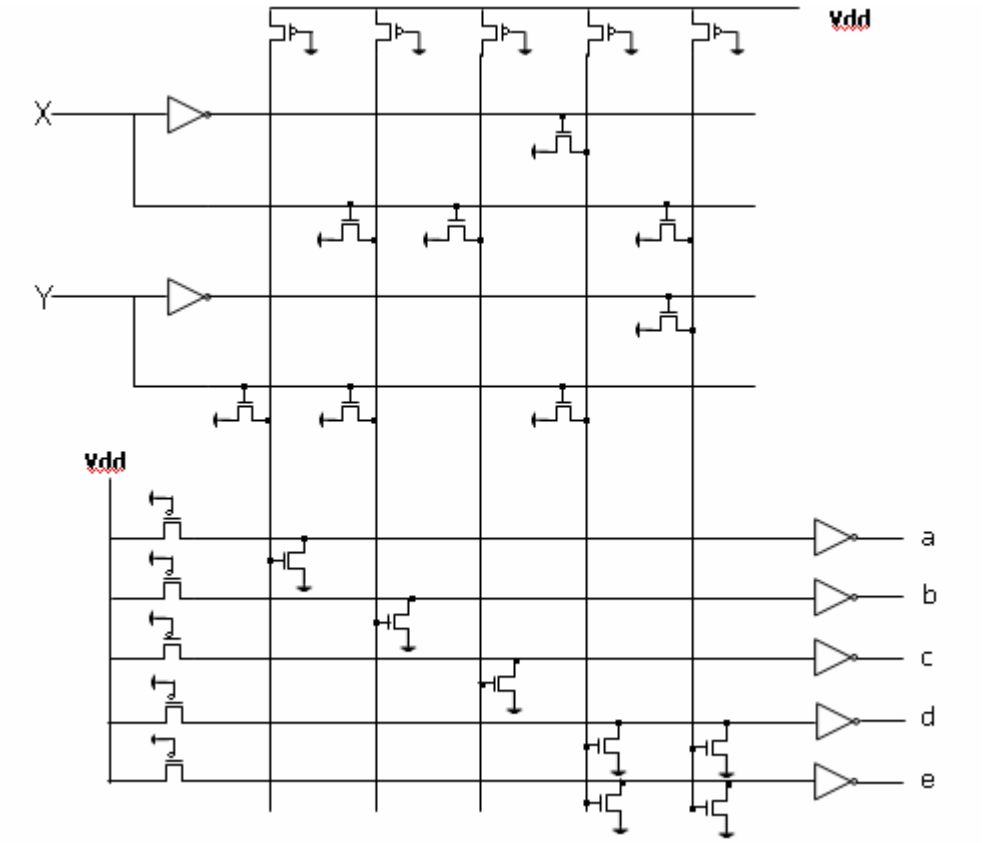


Figure 3.3 The PLA circuit which is under test

3.3 Test Generation for Memory type of circuit

In this section, we will illustrate how to let the memory burn-in and show the simulation result to compare with the normal situation. The general framework sketch of DRAM is shown in figure 3.4.

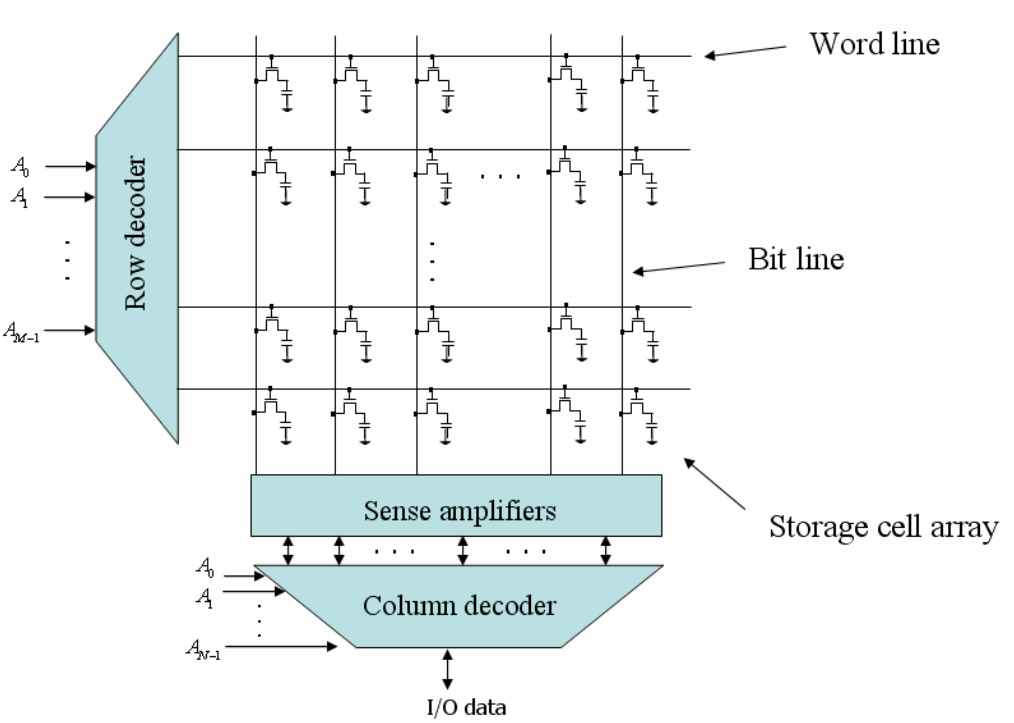


Figure 3.4 The general framework sketch of DRAM.

In order to let the DRAM consume maximum power, we modify the DRAM structure by adding an extra set of PMOS's on each word line to let all of the cells be selected during the test mode. Then we feed each bit line a repeat "0" and "1" signal. The repeated signal fed in the bit line will let the capacitor consume maximum power. The schematic of this burn-in situation is shown in figure 3.5.

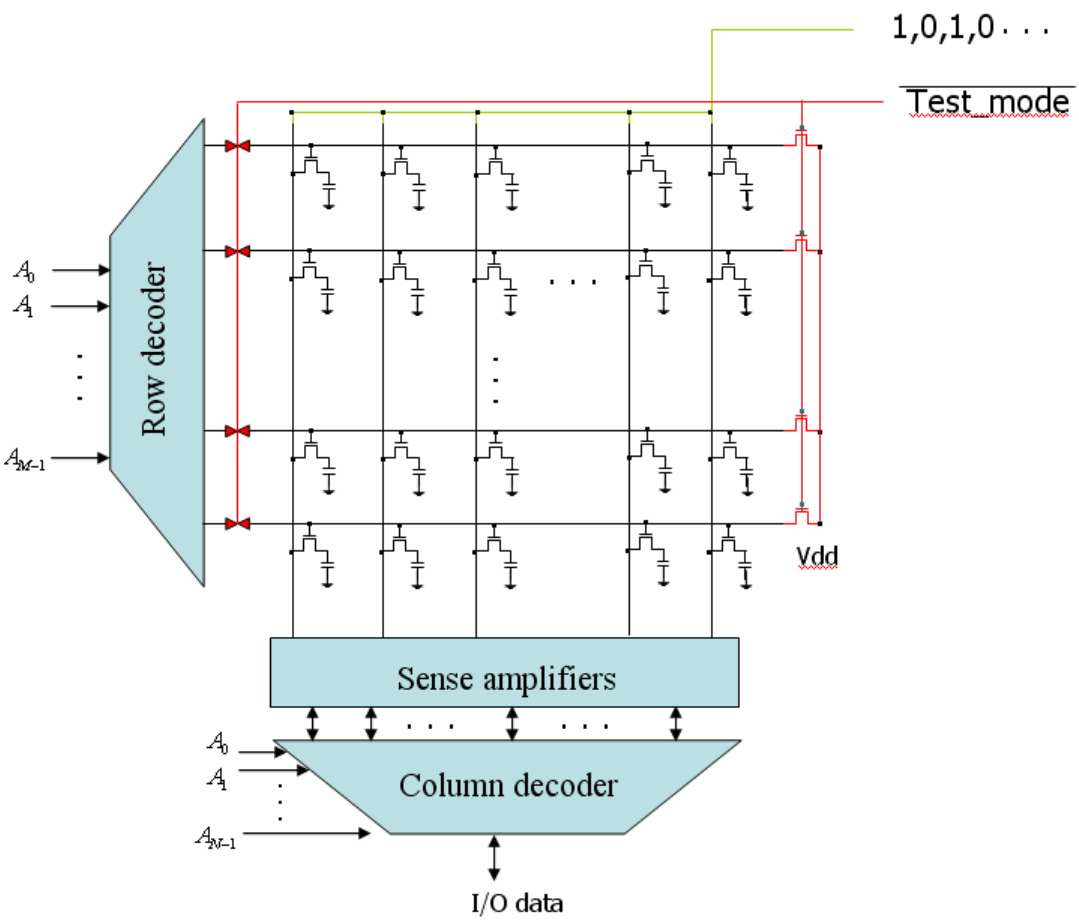


Figure 3.5 The method for testing the DRAM

The extra set of transmission gate on each output of row decoder let the signal from the decoder cut off from the word line when testing. Its aim is to ensure all memory cell can be selected.

The burn-in testing method for an SRAM is similar to that of the DRAM and the schematics of an SRAM under burn-in situation is shown in Figure 3.6.

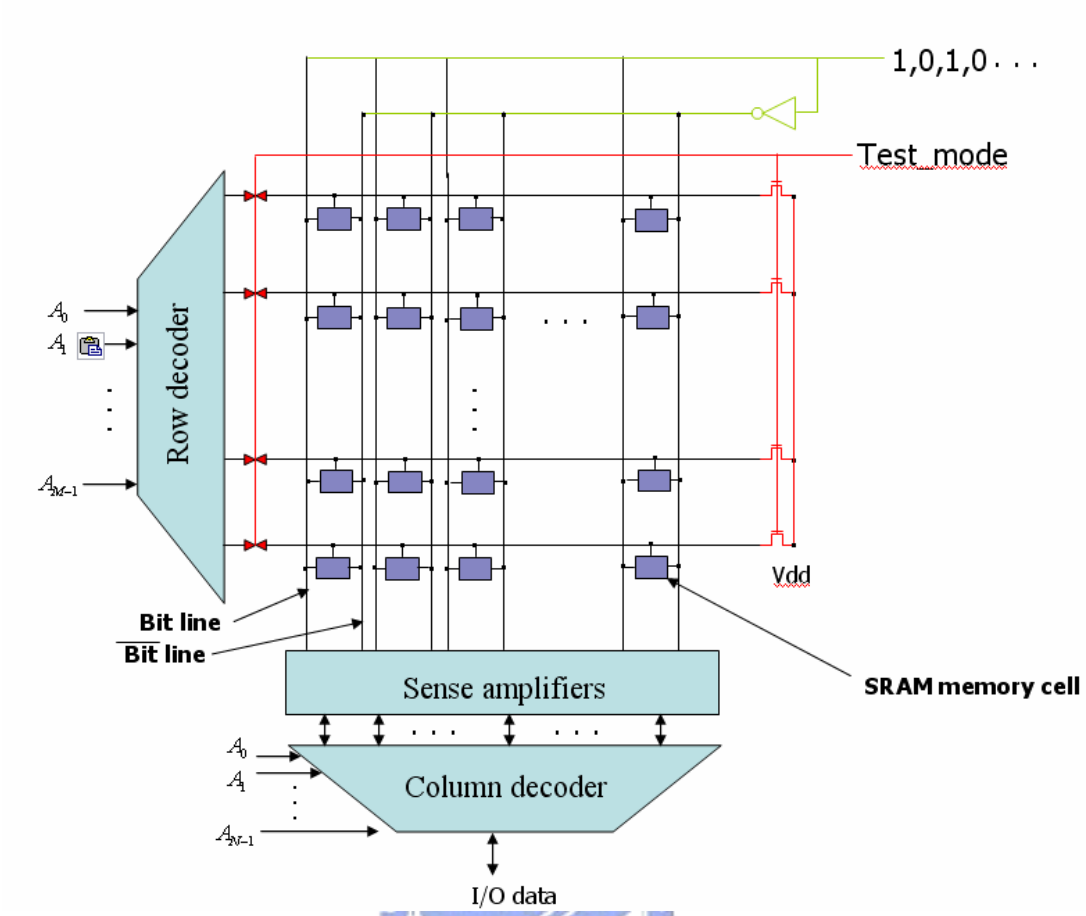


Figure 3.6 The method for testing the SRAM

To verify the above methodology, we designed a 16x16 SRAM and DRAM and simulate power consumption under burn-in situation to compare it with that of the normal situation. From the simulation results, which is shown in Table 3.1, we can see that the method is effective.

	Normal	Burn-in
SRAM	136.5 (uW)	474.9 (uW)
DRAM	157.2 (uW)	521.7 (uW)

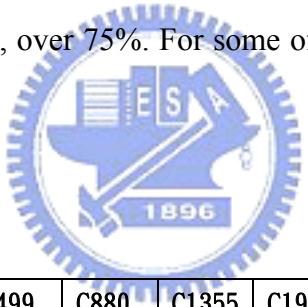
Table 3.1 The simulation results of power consumption under normal and burn-in for the DRAM and SRAM.

Chapter 4

Experimental Results

4.1 Maximum Transitions

The above methodology has been applied to derive test patterns for some of benchmark circuits [ISCAS85 & ISCAS89]. The results are shown in Table 4.1 for the maximal transitions, i.e., no cyclic pattern, for the zero delay model, unit delay model and variable delay model. The percentages on each entry mean that percentage of gates which make switching under the derived set of test patterns. It is seen that the percentages are generally high, over 75%. For some of circuits, even over 90% could be obtained.



	Circuit	C17	C432	C499	C880	C1355	C1908	C3540	S1269C	S13207C	S35932C	S38584C
Zero delay	Switching NO.	4	116	157	261	429	724	1294	469	6090	13362	15679
	Percentage	66.6%	72.5%	77.7%	68.1%	78.5%	82.2%	77.5%	82.4%	76.6%	83.17%	81.42%
Unit delay	Switching NO.	6	726	658	1935	4063	7934	11526	5137	77635	105014	120583
	Percentage	60%	83.7%	88.4%	84.2%	82.2%	90.4%	80.3%	88.2%	82.7%	81.24%	79.23%
Variable delay	Switching NO.	11	561	359	1903	1722	2836	7801	3199	57914	175180	196674
	Percentage	84.6%	88.2%	93.0%	90.1%	88.8%	85.1%	79.2%	81.5%	79.1%	84.17%	83.35%

Table 4.1 Simulation results of the switchings for the generated input patterns for benchmark circuits.

4.2 Maximum Cyclic Average Transitions

The simulation result from the maximum cyclic input pattern are shown in Table 4.2. In this table, the "average amount" means the maximum average transitions that can be obtained. The "average percentage" means the average percentage of the total gates which make switchings.

	Circuit	C17	C432	C499	C880	C1355	C1908	C3540	S1269C	S13207C	S35932C	S38584C
Zero delay	Average Switching NO.	3.5	106.5	142.3	249	388.6	662.6	1198.7	448.2	5740.5	12292	15278.4
	Average Percentage	55%	66.6%	70.5%	65%	71.1%	75.3%	71.8%	78.8%	72.2%	76.51%	79.34%
	Pattern No.	2	2	3	2	3	3	4	4	4	5	5
Unit delay	Average Switching NO.	5.6	693.3	619.6	1898.7	3821.6	451.6	10936.9	4734.6	68717.4	95961	118148.9
	Average Percentage	55%	79.9%	83.2%	82.6%	77.3%	84.9%	76.2%	81.3%	73.2%	74.24%	77.63%
	Pattern No.	3	3	4	4	3	5	6	5	6	9	13
Variable delay	Average Switching NO.	9	520.5	315.6	1748.7	1558.1	2611.5	7220.3	3113.7	55718.8	165374	189123.3
	Average Percentage	69.2%	81.8%	81.7%	82.8%	80.4%	78.4%	73.3%	79.3%	76.1%	79.49%	80.15%
	Pattern No.	3	2	3	6	6	6	8	7	9	11	17

Table 4.2 Simulation result of the maximum cyclic input pattern set

It is interesting to plot the statistic distribution of the numbers of gates in terms of number of swithchings of for the set of the burn-in patterns. Figure 4.1.shows such

a plot for the benchmark circuit C3540 where a similar plot for a set of patterns of a normal case is also included. We can see that for the normal case patterns, most of gates had 5-6 switchings while for the burn-in patterns most of gates had switchings of 7-8. The number of test vectors of the patterns is 8.

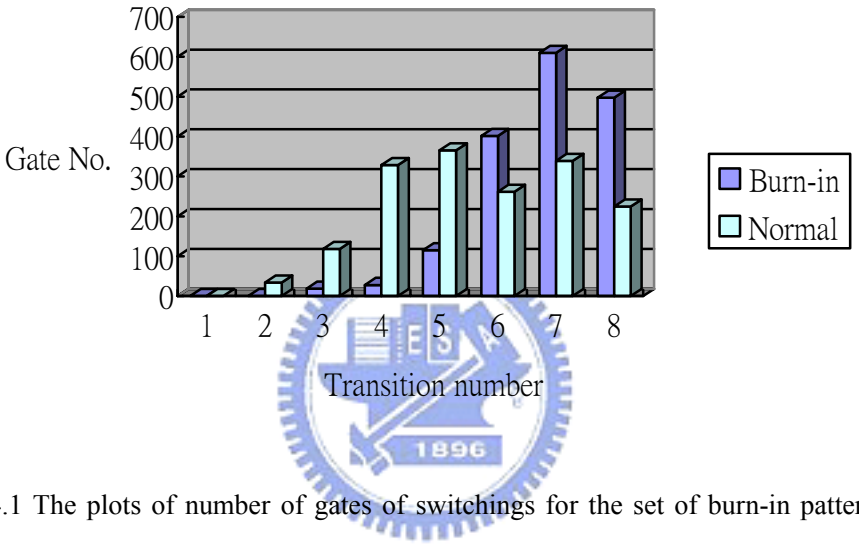


Figure 4.1 The plots of number of gates of switchings for the set of burn-in patterns and the normal case patterns for C3540.

4.3 Simulation on Power Consumption under Applied Cyclic Patterns

Furthermore, we demonstrate the burn-in power consumption for the circuit C1908 which was implemented under the UMC 0.18um technology. It is shown in Figure 4.2. 1in terms of the time. We can find that the circuit does not consume maximum power in the beginning but finally approaches a steady value of consumed power.

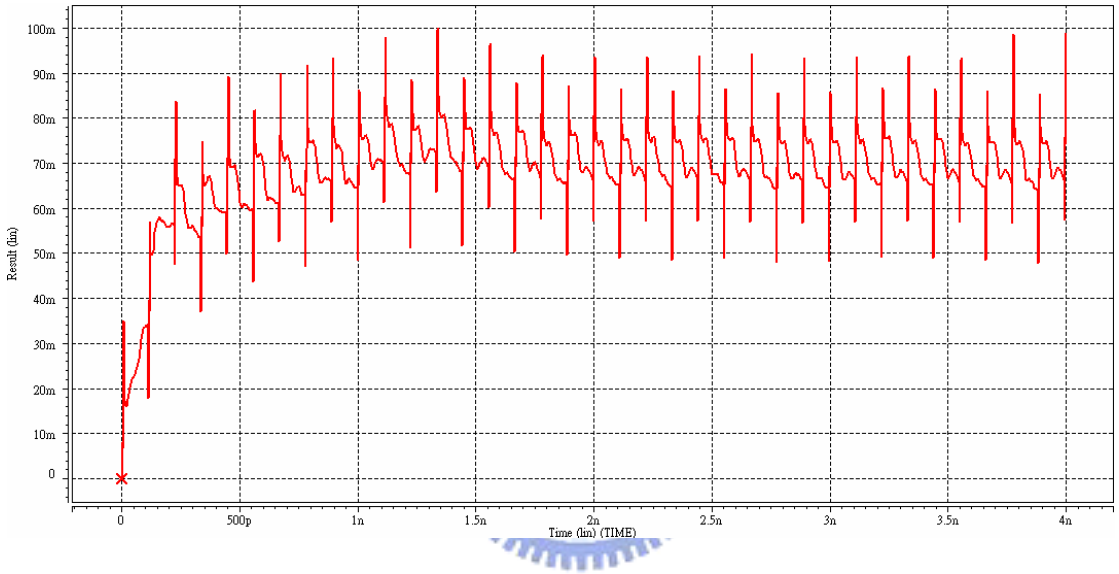


Figure 4.2 The power consumption of C1908

4.4 Power Consumption versus Frequency of Applied Pattern

Another important issue must be resolved is to seek for an optimum frequency to let the circuit produce maximum power consumption. In Figure 4.3, we depict the consumed power versus frequency for the optimal burn-in patterns and the normal case patterns respectively. In the figure, the normal pattern case was obtained by averaging the results of ten random test patterns. It can be seen that for both cases,

the consumed power increases with the pattern frequency and then decreases. This is understandable since the higher frequency of the applied signal, more wavefronts the applied signal are within the circuit, however, when the frequency reaches to a maximum, the circuit can not respond with the signal anymore, causing incomplete switchings of gates and decreasing of power.

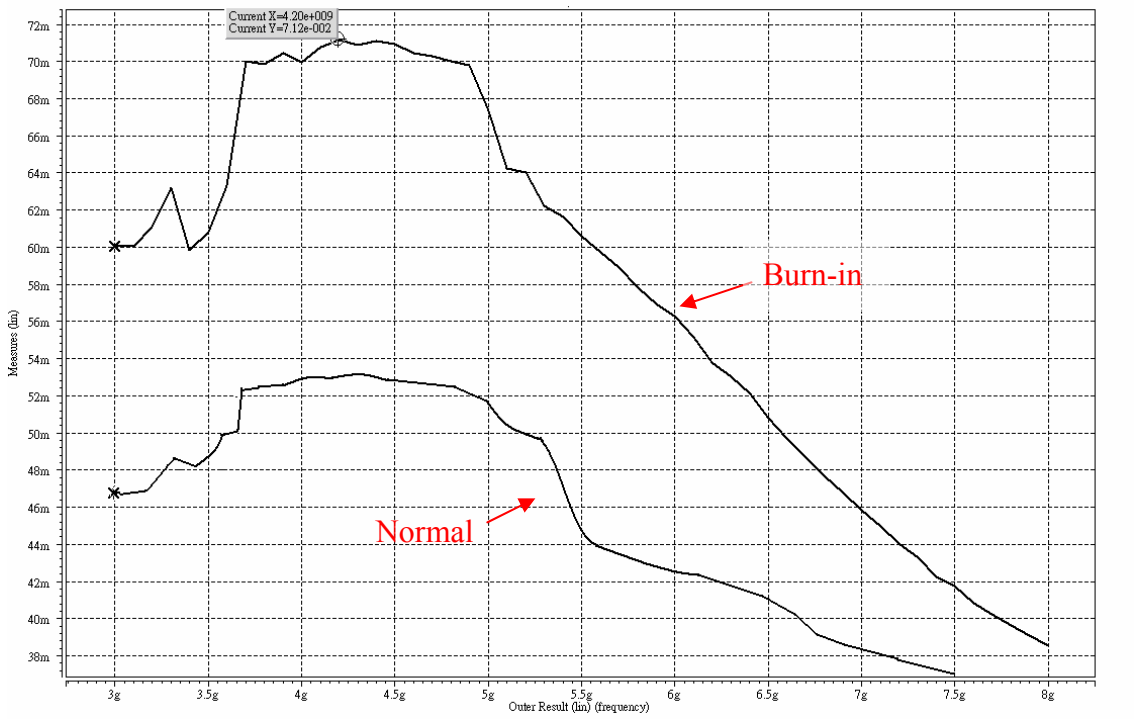


Figure 4.3 The waveform of power versus frequency for the optimal burn-in case and the normal case.

Chapter 5

Circuit Set-up for BIST Burn-in Test

For a circuit to be burn-in tested by using the scheme proposed in this thesis, the number of test vectors of the burn-in patterns in general is small. A simple ROM circuit controlled with a counter-like circuit as shown in Figure 5.1 can serve as the signal generator. For this circuit, the cyclic burn-in test patterns are stored in the ROM, whose output width will equal to the input width of the CUT. When the counter is activated in the test mode, the cyclic test patterns will be repeatedly output to the CUT.

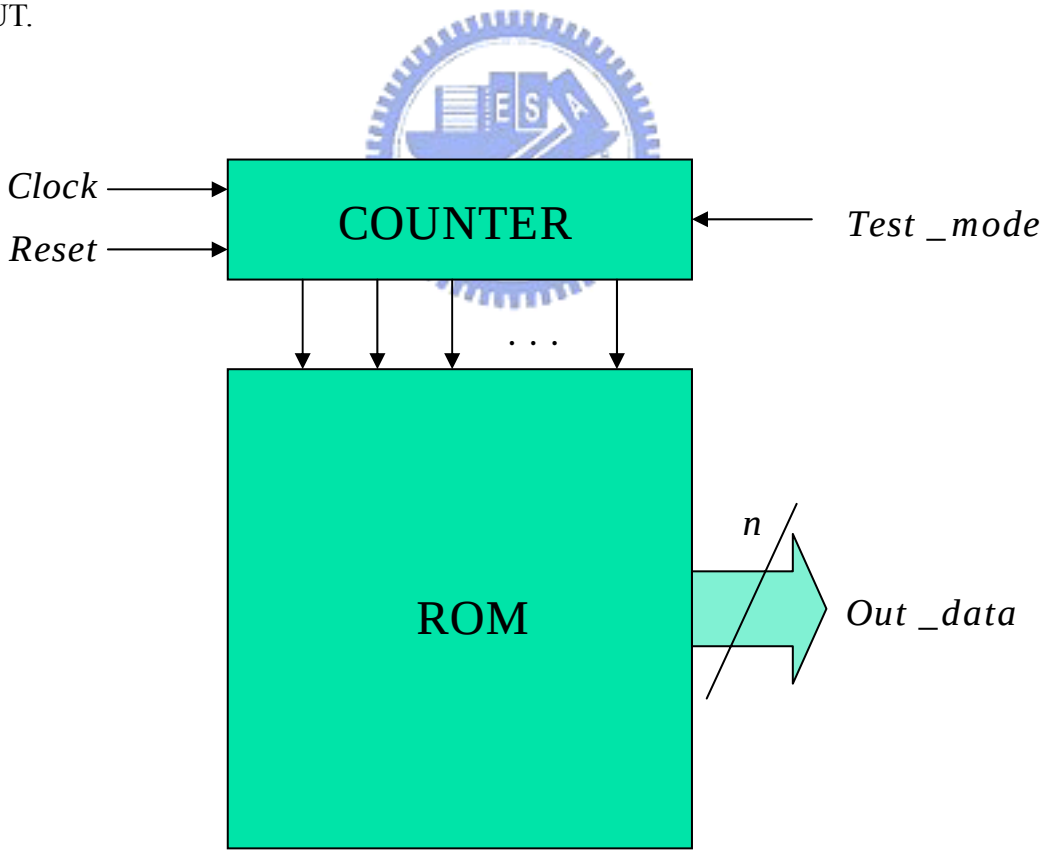


Figure 5.1 The BIST signal generator for the cyclic burn-in patterns

Figure 5.2 shows such a BIST generator to be applied to a combinational CUT burn-in testing and Figure 5.3 shows a similar application to a sequential CUT. For the sequential CUT testing the circuit is treated as a combinational circuit but with the flip-flops are also tested.

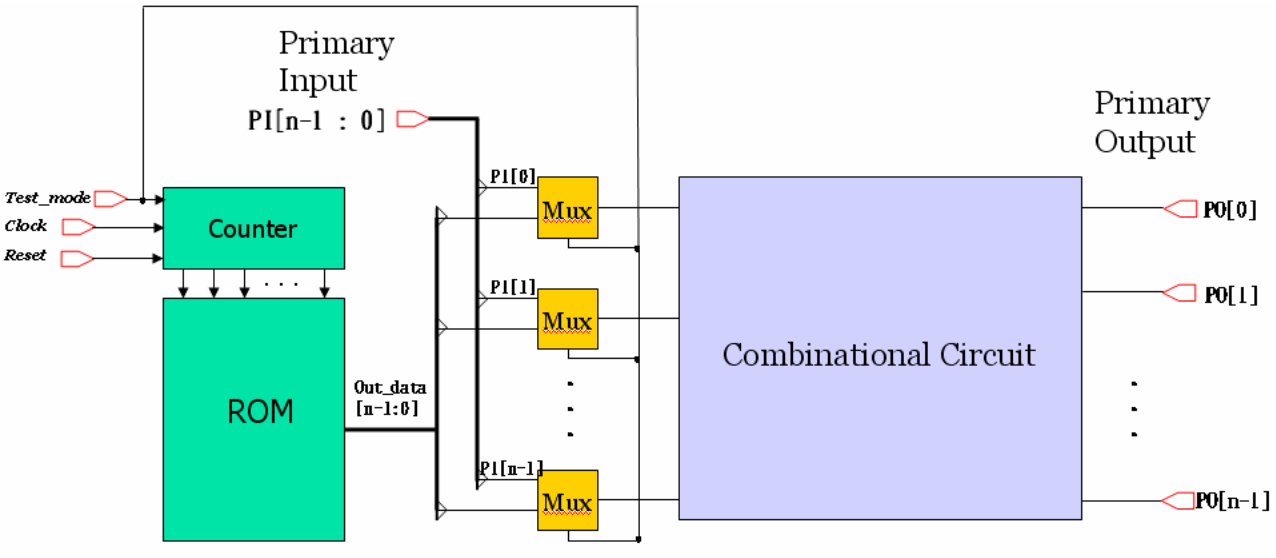


Figure 5.2 The BIST circuit applied to burn-in test a combinational circuit

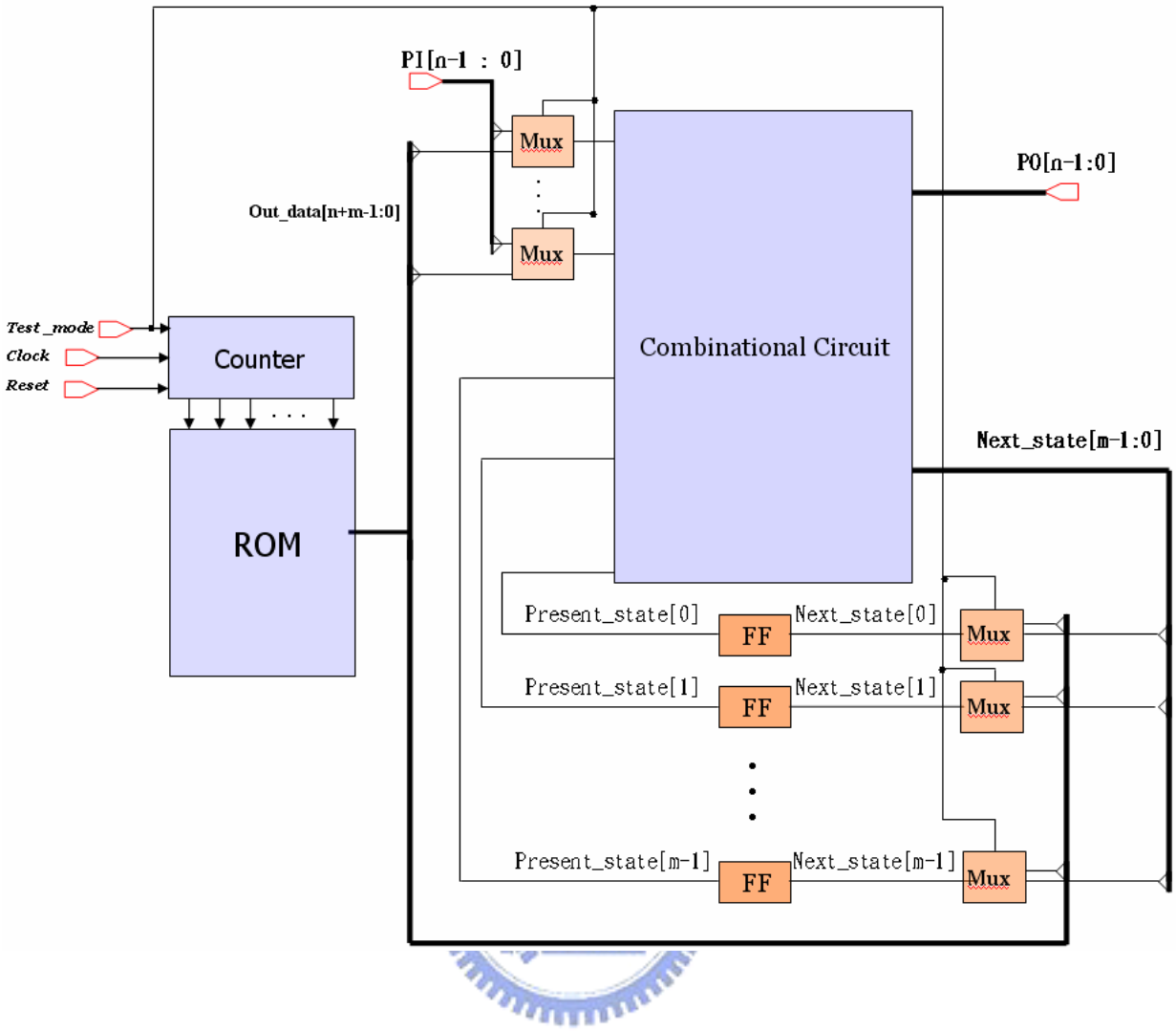


Figure 5.3 The BIST circuit applied to burn-in test a sequential circuit

Chapter 6

Conclusion

In this thesis, we have proposed and studied a BIST burn-in methodology for the SOC which is composed of combinational circuits, sequential circuits, PLA type of circuits and memory type of circuits. In the thesis, we first study the test pattern generation for the combinational circuit and then apply it to the sequential circuit. Also, we have found that optimal patterns for the PLA type of circuits and have proposed a revised design for DRAM and SRAM to let them easily applicable to BIST burn-in testing. Experimental results of applying this approach to some ISCAS benchmark circuits have shown that the patterns generated by this method significantly increase the power dissipation for the circuit to be tested. Furthermore, an architecture for applying the BIST scheme to the SOC is also proposed to make this scheme practical.