Center for Reliable Computing

TECHNICAL REPORT

ELF35 Experiment - Chip and Experiment Design

By James C.-M. Li, Jonathan T.-Y. Chang, Chao-Wen Tseng and Edward J. McCluskey

(preliminary version)

99-3
(CSL TR # 99-3)

October 1999

Center for Reliable Computing
Gates Bldg. 2A, Room #236
Stanford University
Stanford, California 94305-9020

Abstract:

A test chip has been designed and manufactured to evaluate the effectiveness of different test techniques for deep submicron technologies. The test chip uses LSI Logic G10p 0.35µm cell-based technology. It has approximately 265k LSI Logic equivalent gates. There are six types of circuits-under-test(CUTs).

Two CUTs are arithmetic processors, which perform the same function but were implemented in different ways. These two CUTs are full-scan sequential circuits. The other four are combinational circuits. Three of these four combinational CUTs are datapath circuits. The other one is a translator that maps a pseudo-random sequence into a binary sequence. The tests include stuck-at fault, delay fault, transition fault, design verification, pseudo-random, weighted-random, and I_{DDQ} tests. The I_{DDQ} test sets were generated based on various fault models, such as pseudo-stuck-at and bridging faults. A built-in-self-test (BIST) circuitry was implemented for one datapath CUT. Emulated BIST test sets for some CUTs will also be used. This report describes the chip design and the test applied. Future reports will describe the experimental results and data analysis.

Funding:

This work was supported in part by the National Science Foundation under Grant No. MIP-9107760 and by LSI Logic Corporation under Agreement No. 16517.

Imprimatur:

ELF35 Experiment - Chip and Experiment Design

By James C.-M. Li, Jonathan T.-Y. Chang, Chao-Wen Tseng and Edward J. McCluskey

CRC Technical Report No. 99-3
(CSL TR No. 99-3)
October 1999

CENTER FOR RELIABLE COMPUTING

Gates Bldg. 2A, Room #235
Computer System Laboratory
Department of Electrical Engineering
Stanford University
Stanford, California 94305-9020

ABSTRACT

A test chip has been designed and manufactured to evaluate the effectiveness of different test techniques for deep submicron technologies. The test chip uses LSI Logic G10p 0.35µm cell-based technology. It has approximately 265k LSI Logic equivalent gates. There are six types of circuits-under-test (CUTs).

Two CUTs are arithmetic processors, which perform the same function but were implemented in different ways. These two CUTs are full-scan sequential circuits. The other four are combinational circuits. Three of these four combinational CUTs are datapath circuits. The other one is a translator that maps a pseudo-random sequence into a binary sequence. The tests include stuck-at fault, delay fault, transition fault, design verification, pseudo-random, weighted-random, and I_{DDQ} tests. A built-in-self-test (BIST) circuitry was implemented for one datapath CUT. Emulated BIST patterns for some CUTs will also be used.

This report describes the chip design and the test applied. Future reports will describe the experimental results and data analysis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Many test techniques have been proposed to improve the quality level for integrated circuits (ICs). However, as the industry moves toward advanced submicron or deep submicron technologies, the test techniques that used to work in the past might not perform well nowadays. For example, there are more metal layers for deep submicron technologies, which implies that bridging faults are more likely to occur for devices manufactured by these advanced technologies. Moreover, as the cycle time reduces and the operating frequency increases, timing defects have more severe impacts for deep submicron technologies than for previous technologies. Also, the quiescent background current for deep submicron technologies may be too high to perform I_{DDQ} testing. There have been several experiments focusing on evaluating test methods [Maxwell 96] [Powell 96] [Sematech 97]. We have also performed similar experiments [Franco 94] [Franco 95] [Ma 95] [Franco 96] [Chang 98]. However, there have been few experimental results reported for deep submicron technologies. ELF35 experiment will address the challenge of testing for deep submicron technologies. The objective of this experiment is to evaluate the effectiveness of many different test techniques for advanced deep submicron technologies. To incorporate all the test techniques, we designed a test chip for this experiment. This methodology was also applied in a previous experiment [Franco 94].

We designed and manufactured a test chip using the LSI Logic G10p 0.35 μm cell-based technology. It has 265k gates and uses three metal layers. The test chip has 208 I/O pins, and 64 supply pins. It uses a 272-pin PBGA package. The die size is 10.13mm by 10.13mm. The test support circuitry design is similar to the one used in the MURPHY experiment [Franco 94]. It includes the on-chip data sources and the response analysis circuitry. To ensure that we can detect all the defects in the test support circuitry, all the data sources and response analysis circuitry were duplicated. Section 3 describes the details of the chip architecture and specifications.

The test chip has six types of circuits-under-test (CUTs). Two of the CUTs are sequential and the others are combinational circuits. The two sequential elements are 2901 arithmetic processors. One 2901 processor was generated by mapping an existing design from a previous LSI Logic library to the LSI Logic G10p library. The other 2901 processor was generated by the TOPS tool, a high-level synthesis tool developed at

Stanford CRC. Both 2901 processors are full-scan designs. One combinational CUT is a translator that maps a pseudo-random sequence to a binary sequence. One combinational CUT is a datapath logic that was generated by using the LSI Logic logic-block synthesizer. The other two combinational CUTs are the two datapath circuits used in the MURPHY experiment [Franco 94]. These two CUTs have the same gate-level structure as the ones used in the MURPHY experiment but they were mapped to the LSI Logic G10p technology. Section 2 describes the details of the CUT designs.

The tests to evaluate include stuck-at fault (SSF), delay fault, transition fault, design verification, pseudo-random, weighted-random, and I_{DDQ} tests. Multiple-detect SSF test sets are also included. The I_{DDQ} test sets were generated based on various fault models, such as pseudo-stuck-at and bridging faults. A built-in-self-test (BIST) circuitry was implemented for one datapath CUT. Emulated BIST patterns for some CUTs will also be used. The test sets were provided by ATPG vendors and university partners. A new test techniques will be evaluated: Very-Low-Voltage (VLV) testing [Hao 93] [Chang 96a] [Chang 96b]. We will use the Credence Logic Duo 100MHz Test System to perform the wafer probe and the Advantest T6682 tester t o perform the final package test.

This report is organized as follows. Section 2 describes the CUT designs. Section 3 describes the chip architecture and the test support circuitry. Section 4 describes the test plan. Section 5 summarizes the report.

2. CIRCUIT UNDER TEST

The CUTs for the ELF35 experiment consists of arithmetic processors, datapath logic and a pseudo-random-to-binary translator. Table 1 lists the basic information for the six CUTs. This section provides the detailed information for each CUT .

The criteria of the CUT selection were:

a. real world circuitry,

b. including sequential circuitry,

c. including circuitry with a low detectability profile,

d. including some CUTs used in the MURPHY experiment.

Table 1 The Basic Information for the Six CUTs

Name	Inputs	Outputs	LSI Gates*	Description
LSI2901	61**	64***	12,338	Arithmetic processor from the LSI Logic library
TOPS2901	57**	48***	18,090	Arithmetic processor generated by the TOPS high-level synthesis tool
MULT_ADDER	65	33	4,499	Multiplier followed by an adder
PB	12	12	17,468	Pseudo-random-to-binary translator
M12	24	12	1,309	12×12 multiplier
SQR	12	6	538	6×6 multiplier followed by a squarer

* one LSI gate ≈ one cell unit / 3.2 for the LSI Logic G10p technology

** include one clock pin, one scan enable pin, and one scan input pin

*** include one scan output pin

Unlike the requirements for the CUTs in the MURPHY experiment, there was no limitation on the number of inputs and outputs. The two arithmetic processors are sequential circuits with the full-scan feature. One was generated by mapping an existing gate-level design from a previous technology to the LSI Logic G10p technology. The other one was generated by using the TOPS high-level synthesis tool developed at Stanford CRC. One combinational CUT was generated by using an LSI Logic tool. Two

combinational CUTs that were also used in the MURPHY experiment were mapped to the LSI Logic G10p technology. The pseudo-random-to-binary translator has a low detectability profile. Each CUT was replicated at least twice on the chip. To ensure that the timing of different copies for each CUT type remain the same, each CUT type was laid out as a hard macro (*hardmac*).

2.1 LSI2901 and TOPS2901: Arithmetic Processors

This is a 32-bit version of the arithmetic processor AM2901 [AMD 83]. The AM2901 industry standard four-bit microprocessor slice is a cascadable ALU intended for use in CPUs, peripheral controllers, and programmable microprocessors. The 32-bit version of AM2901 consists of a 16-word by 32-bit two-port RAM, a high-speed ALU, and the associated shifting, decoding and multiplexing circuitry. The microprocessor is cascadable with full look ahead or with ripple carry. It has three-state outputs, and provides various status flag outputs from the ALU.

Two different implementations were used for the ELF35 experiment. The first one, LSI2901, was generated by transforming a gate-level implementation of a 16-bit slice from a previous LSI Logic technology into a 32-bit version arithmetic processor that uses the LSI Logic G10p technology. Without modifying the structure of the original implementation, the transformed design was synthesized by using a commercial logic synthesis tool to properly adjust the driving strength of each gate. Each bidirectional pin for AM2901 becomes three pins in LSI2901: one for input, one for output, and the other one for output enable. The three-state outputs in AM2901 become two-state output pins in LSI2901.

A 16-word 32-bit asynchronous RAM combined with a 32-bit latch array at the output of the RAM in the processor were replaced by 512 scan flip-flops. The pseudo-RAM performs the same function as the original asynchronous RAM and its output latch array do in the arithmetic processor. A scan chain connects these 512 flip-flops with the other 32 flip-flops in the processor. All the flip-flops used in this transformed arithmetic processor are rising-edge-triggered scan flip-flops.

The other arithmetic processor, TOPS2901, was generated by using the TOPS high-level synthesis tool [Avra 94] [Touba 96] [Norwood 97]. The TOPS tool is a synthesis-for-test tool which takes the function-level VHDL description of a design and

automatically generates a dependable, gate-level VHDL implementation of the design. The same pseudo-RAM used in LSI2901 was also incorporated in TOPS2901. There are 961 scan flip-flops in TOPS2901. They are connected to a single scan chain. All flip-flops are rising-edge-triggered scan flip-flops. Some flip-flops have the parallel-load enable feature. Table 2 lists the area, the delay, the utilization, and the scan chain length of the two arithmetic processors. The *utilization* is the fraction of the total area that is occupied by gates. The typical utilization for a three-metal-layer design is 50-60%.

Table 2 Specifications of LSI2901 and TOPS2901

Name	Area	Delay	Utilization	Scan chain length
LSI2901	1.073mm × 1.073mm	15.2ns	60.38%	544
TOPS2901	1.3mm × 2.0mm	18.6ns	56.43%	961

2.2 MULT_ADDER: Multipliers Followed by an Adder

This CUT consists of two 16×16 multipliers and a 32-bit adder. Figure 1 shows the structure of this CUT. The 32-bit outputs of both multipliers were connected to the inputs of the adder. The carry-in signal can be controlled from the primary inputs of the CUT.

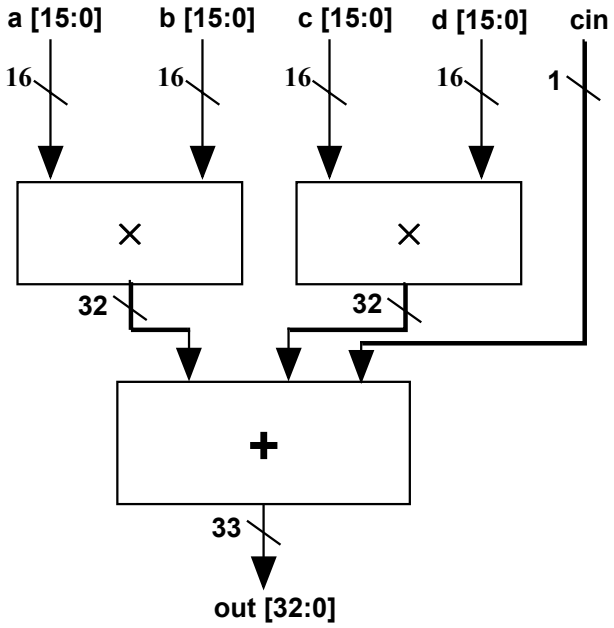

Figure 1 The Structure of MULT_ADDER

This CUT was generated by using the Logic Block Synthesizer of LSI Logic, which can automatically generate basic logic blocks such as multipliers, adders, shifters, and counters. It can also generate complex functions such as ALU and RAMBIST. The reason for including this CUT is to investigate the potential testing problem for any circuitry generated by tools similar to the Logic Block Synthesizer. Table 3 lists the area, the delay, and the utilization of MULT_ADDER.

Table 3 Specifications of MULT_ADDER

Name	Area	Delay	Utilization
MULT_ADDER	0.885mm × 0.7mm	12.0ns	56.93%

2.3 M12 and SQR: Multipliers

M12 has the same gate-level structure as MULT12O12 used in the MURPHY experiment [Franco 94]. It is a 12×12 partial multiplier made up of 6×6 multiplier building blocks. SQR has the same gate-level structure as MULT6SQ used in the MURPHY experiment [Franco 94]. The detailed description of these two CUTs can be found in [Franco 94]. Although the gate-level structure of these two CUTs remain the same as the ones used in the MURPHY experiment, the driving strength of each gate was properly adjusted by using a commercial logic synthesis tool. Since M12 has only 24 inputs, we can apply an exhaustive test for it. In addition, since SQR has only 12 inputs, we can apply a super-exhaustive test for it. Table 3 lists the specifications of these two CUTs for the ELF35 experiment.

Table 4 Specifications of M12 and SQR

Name	Area	Delay	Utilization
M12	0.238mm×0.833mm	12.5ns	52.94%
SQR	0.238mm × 0.338mm	14.0ns	55.93%

2.4 PB: Pseudo-Random-to-Binary Translator

This CUT can translate a pseudo-random sequence into a binary sequence. The reason for including this circuit is that it has many faults with low detectability. Table 5 lists the detectability profiles of all 15 4-bit PB CUTs that map pseudo-random sequences with different seeds to the same binary sequence. In Table 5, h_k is the number of faults

that are detected by k vectors, where k can be any number from zero to 16 here. For all 15 4-bit PB CUTs, there are at least 50% of the faults that can be detected by fewer than 3 vectors.

Table 5 The Detectability Profile of a 4-bit PB CUT

Seed	h_k																	total # of faults
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0001	0	18	18	14	13	2	1	4	17	1	1	0	1	0	0	0	0	90
0010	0	23	22	18	14	3	4	2	19	0	1	0	1	0	0	0	0	107
0011	0	13	23	6	13	1	1	3	19	0	3	0	3	0	0	0	0	85
0100	0	14	30	15	8	5	6	6	17	2	0	0	0	0	0	0	0	103
0101	0	16	20	17	7	4	5	1	16	0	0	0	0	0	0	0	0	86
0110	0	14	14	13	12	7	2	4	16	2	0	0	1	0	0	0	0	85
0111	0	22	20	12	11	8	2	0	16	0	1	0	0	0	0	0	0	92
1000	0	29	16	11	13	4	3	0	18	0	3	0	1	0	0	0	0	98
1001	0	14	23	12	7	6	6	4	17	0	2	0	0	0	0	0	0	91
1010	0	10	28	7	8	2	4	3	18	1	1	0	1	0	0	0	0	83
1011	0	20	21	11	8	4	9	3	17	0	1	0	1	0	0	0	0	95
1100	0	21	9	20	4	6	4	0	16	1	1	0	0	0	0	0	0	82
1101	0	23	15	10	13	2	6	2	16	0	3	0	1	0	0	0	0	91
1110	0	24	14	10	6	3	7	1	18	0	0	2	0	0	0	0	0	85
1111	0	7	19	18	6	4	3	5	16	0	1	1	0	0	0	0	0	80

We use a 12-bit version of the translator in the ELF35 chip. Because there are only 12 inputs, we can apply a super-exhaustive test set for it. PB has many nodes with high connectivity and thus, can have many potential bridging faults. However, it is difficult to lay out this circuit because of the high-connectivity nodes. In order to ease the layout problem, we built four translators that map the lower 10-bit inputs to 12-bit binary sequences with different ranges. The higher two-bit inputs are then used to select the outputs of one of the four translators and generate the final outputs. Figure 2 shows

the detailed design of the 12-bit PB. In Fig. 2, PB0, PB1, PB2, and PB3 are the four translators mentioned above.

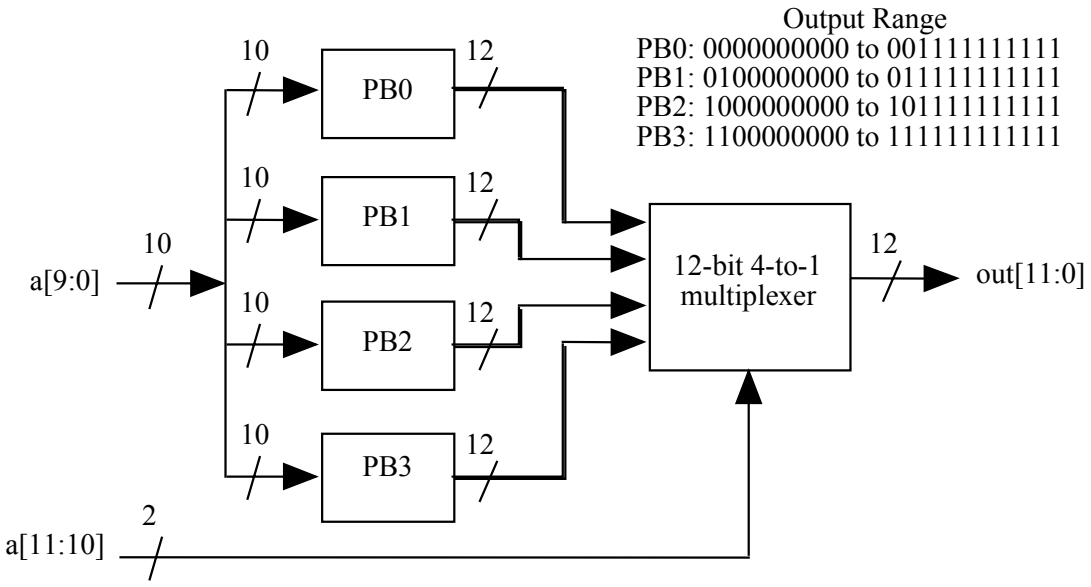

Figure 2 Block Diagram of PB

Table 6 lists the specifications of this CUT. The utilization of this circuit is very low compared to the other five CUTs.

Table 6 Specifications of PB

Name	Area	Delay	Utilization
PB	2.6mm × 3.1mm	28.37ns	16.32%

3. TEST CHIP ARCHITECTURE

Similar to the MURPHY test chip, the ELF35 test chip consists of CUTs and test support circuitry. The test support circuitry includes data sources and response analysis circuits. The chip was designed so that most of the area can be occupied by CUTs. The equivalent gate count of the chip is approximately 264k. The gate count for all CUTs is approximately 217k. CUTs contribute 82% of the gates on the chip. CUT hardmacs and the related routing area occupy 74% of the chip area.

To simply the test support circuitry, only the at-speed clocking mode is used for the ELF35 experiment. All the flip-flops are positive edge triggered. We duplicated every unit of the test support circuitry to ensure that all the defects in the test support circuitry can be detected before testing CUTs. In compliance to the LSI Logic design requirement, a processor monitor (the LSI Logic procmon cell) and an iddtn buffer were incorporated in the ELF35 design.

This section is organized as follows. Section 3.1discribess the block diagram of the test chip. The whole chip's architecture is discussed in details. Section 3.2 describes the design and functionality of the test support circuitry that supplies input data. Section 3.3 describes the design and functionality of the test support circuitry that analyzes output from CUTs. The test support circuitry for the combinational CUTs and sequential CUTs are basically the same if not mentioned otherwise.

3.1 Block Diagram

Fig. 3 shows the block diagram of the test chip. It uses a 272-pin PBGA (plastic ball grid array, coded ig56 in the LSI Logic BGA package datasheet) package. The test chip has 178 input pins, 30 output pins, and 64 V_{DD}/V_{SS} pins. To eliminate the static current, none of the I/O pins has either pull-up or pull-down resistors. We used only static CMOS logic gates for the ELF35 chip. There is no tri-state bus and no bi-directional pin. The detailed description of the pins for the ELF35 chip can be found in Appendix A.

The ELF35 chip is a full-scan design. There are 17 scan chains overall for the test support circuitry, six for the data source circuitry and 11 for the response analysis

circuitry. The detailed information of the scan chains for the ELF35 chip can be found in Appendix B.

Figure 3 ELF35 Chip Block Diagram

A "fail" signal is designed for each CUT. It signals that the associated CUT fails a test vector. Consequently, we can record the test result of each vector by observing this pin while testing a CUT.

Each CUT was laid out as a hardmac. Because the size of each CUT type is different from others, each CUT was replicated different times in order to distribute the area to each CUT type

3.2 Data Source Circuitry

3.2.1 LFSR

All primary inputs are applied to the CUTs via on-chip Parallel Data Load Linear Feedback Shift Registers (LFSRs). They are designed to generate on-chip pseudo random/exhaustive patterns as well as to latch the input patterns from the tester.

Fig. 4 shows a 24-bit LFSR (for PB, M12 and SQR). It has four modes of operations. In the scan mode, the LFSR is configured into a scan chain. This mode is used when testing the LFSR and data source circuitry. In the parallel load mode, the input patterns from the tester are latched in the LFSR. In the pseudo random mode, the LFSR generates pseudo random patterns with polynomial $H(x) = x^{24}+x^{7}+x^{2}+x+1$. In the no change mode, the LFSR keeps the previous status. All the other LFSRs are the same in structure and function. They just differ in the number of bits.

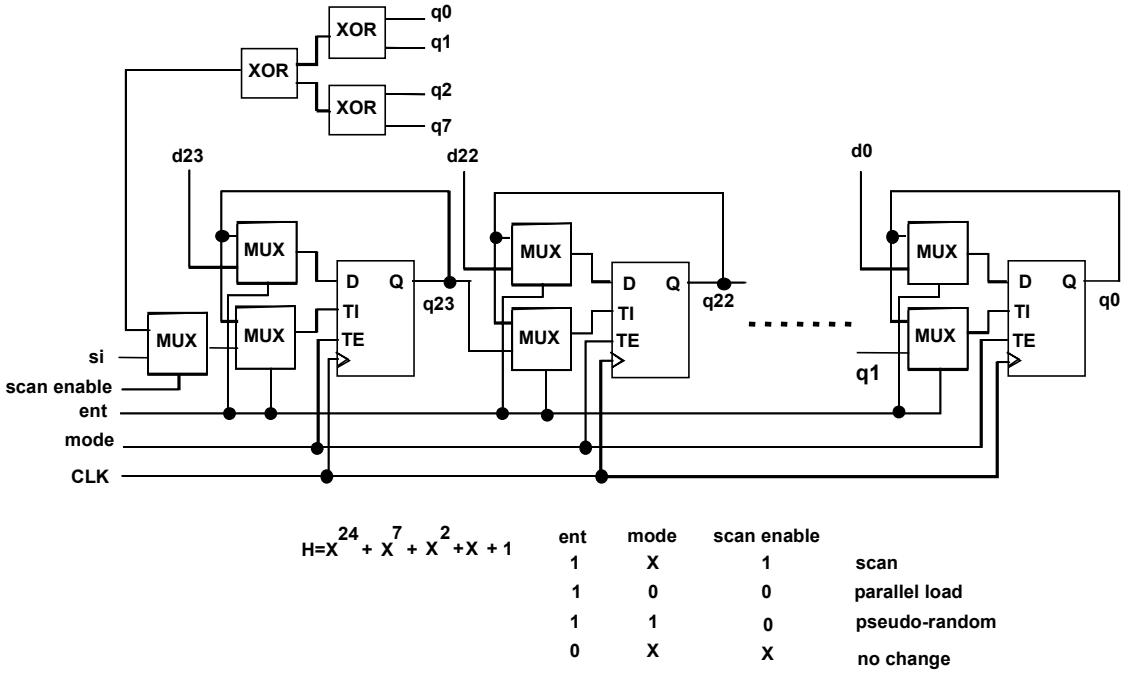

Figure 4 24-bit LFSR

The LFSRs are tested with 100% SSF test patterns as well as functional patterns. However, to double check that the LFSR is working correctly when testing the CUTs, every LFSR has an extra copy. Every bit of the output of these two copies is compared when testing the CUTs. If there is any mismatch between any two bits, the comparator will set a bit to one to indicate an error. This bit will be shifted out at the end of every CUT testing.

3.2.2 BIST for M12

On-chip hardware BIST circuitry has been designed for the M12 CUT. The outputs of two copies of 24 bits LFSRs pass through a specially designed mapping logic. This mapping logic maps pseudo random patterns into high coverage patterns so that it reduce the number of patterns from 75K to 20K to achieve 100% SSF coverage.

Fig. 5 shows the block diagram of the BIST circuitry. Outputs from two copies of LFSRs and mapping logic circuits are compared to guarantee correctness. A BIST mode signal is used to select between BIST and non-BIST patterns.

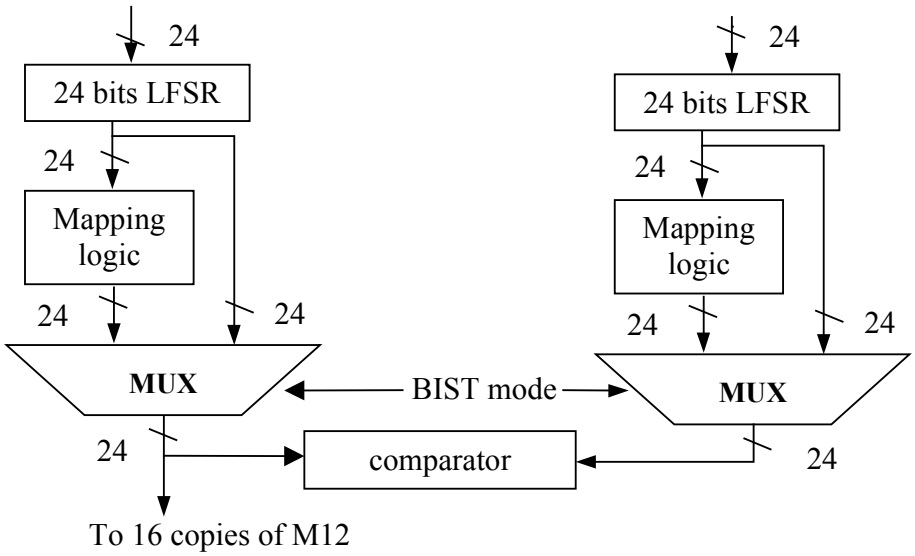

Figure 5 BIST Circuitry of M12

3.2.3 Sequential CUTs

For the sequential CUTs, all the inputs except three also pass through LFSRs. These three exceptions are “clock”, “scan-in” and “scan-enable”. The CUT clock pin is shared with the clock of the support circuitry. The “scan-in” signal passes through a MUX (selected by clockmode and imode signal) and then enters the CUT. The “scan-enable” is directly connected to the flip-flops in the CUT.

To make sure these “scan-in” and “scan-enable” signals get to the CUT without any problem, an “andtree” is designed to observe these signals. This is a tree of MUX which is selected by the “andsel” signal. This andtree is tested in the support circuitry tests. Please see Appendix A for more details.

3.3 Response Analysis Circuitry

All the outputs of CUTs are captured in the output registers. These output registers are chained so that every output can be observed by shifting. These output register scan chains are designed for further diagnosis purpose.

After captured in the output registers, the output values of multiple copies are then compared. Total numbers of failures, failure locations and the first pattern that fails will be recorded. These values will be observed at the end of every CUT test.

3.3.1 Comparators, Failure Locators and Counters

Fig. 6 shows the structure of a comparator and failure locator. The comparator compares the outputs from the first copy of CUT with the other CUTs. If there is any mismatch, the flip-flop will be set to one. These flip-flops will be shifted out at the end of every CUT test. A single 1 in these flip-flops represents at least one failure has been found in the corresponding CUT. All ones in these flip-flops means at least one failure has been found in the first copy.

A fail signal will be set to one if there is any mismatch. This fail signal will increment the on-chip total failure counter by one and freeze the first failure counter. The content of these two counters will be shifted out at the end of every CUT test. They are all binary counters. If there is no fault detected, the content of the total failure counter should be zero and the first failure counter should be the total length of the test.

The comparators, failure locators and counters are tested with 100% SSF test patterns as well as functional patterns. However, to double check that the response analysis circuitry works correctly when testing the CUTs, an extra copy of comparators, failure locators and counters is duplicated. The fail signals from these two copies of comparators are ORed and sent to the primary output of the chip. The contents of these two copies of counters are shifted out at the end of every test. This duplication guarantees the correctness of the results from the response analysis circuitry.

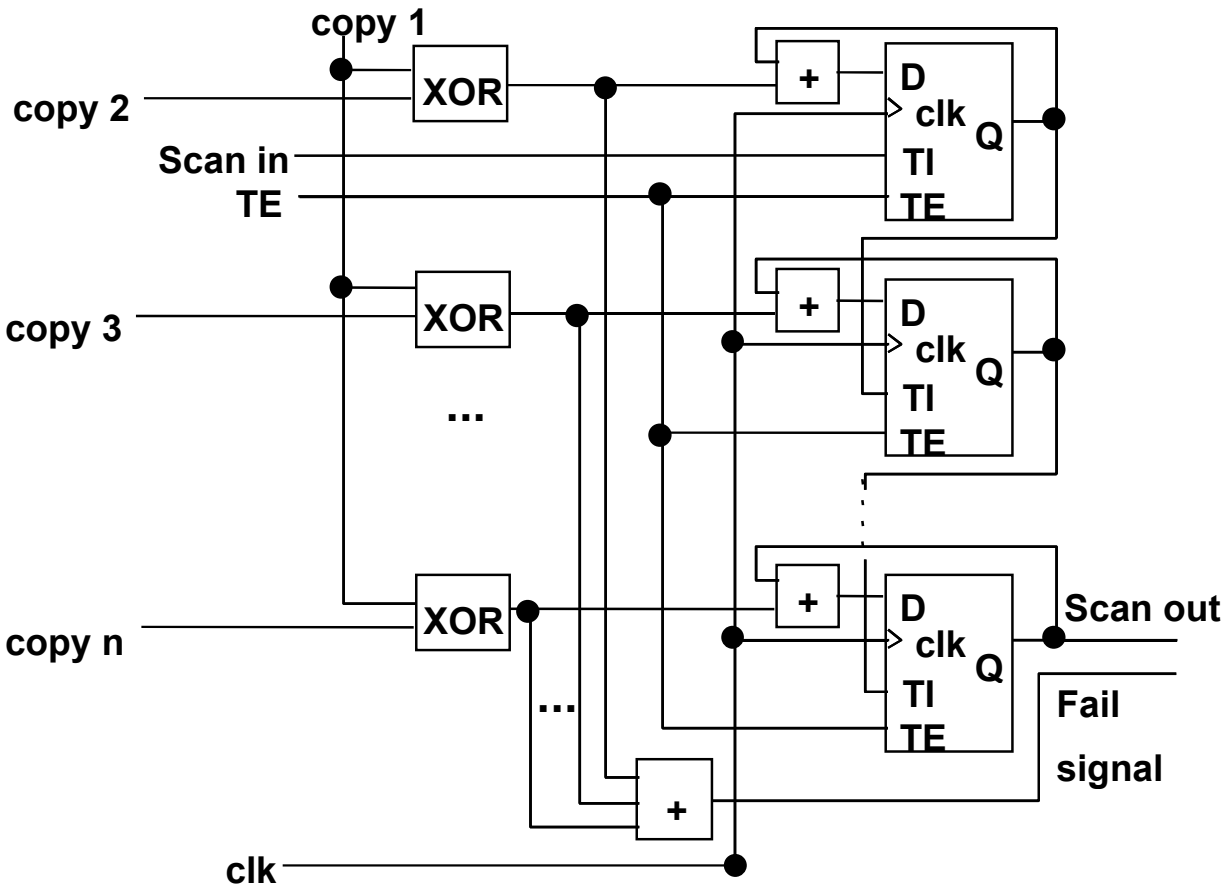

Figure 6 Failure Locator and Comparator (one bit slice)

3.3.2 Sequential CUTs

For sequential CUTs, the response analysis circuitry is basically the same except that sequential CUTs needs an additional one bit failure counter.

Fig. 7 shows the response analysis circuitry of Lsi2901. The 63 bits primary outputs can be mask individually. This is designed to mask the "don't care" output pins when testing the sequential circuit. After the mask logic, the output values go to two copies of primary outputs (PO) comparators and failure counters (FC).

The one bit scan output (SO) is also compared in the same way as described before. The comparison results from the two copies of PO comparators and two copies of SO comparators are ORed and sent to the chip's output.

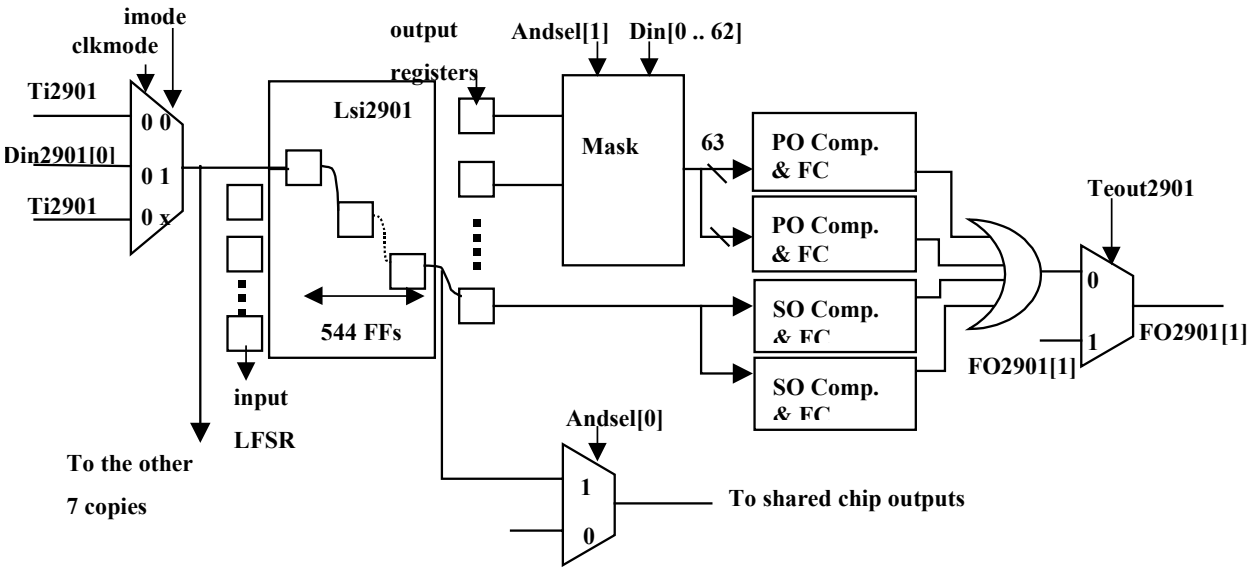

Figure 7 Lsi2901 Response Analysis Circuitry

The CUT scan chains are also directly observable from the chip's output. They are selected by the andsel[0] signal. This design is for debug and diagnosis purpose.

Tops2901 is also designed in the same way. Please see Appendix A for more details.

4. TEST PLAN

The ELF35 test chip is designed to evaluate the effectiveness of different test techniques. Therefore, many test sets are collected or generated for every CUT. A very thorough test flow is also implemented to test the chip at different conditions.

This section describes the test plan for the test chip. Section 4.1 gives the details of all the test sets. Section 4.2 gives all the details of the test flow.

4.1 Test Sets

The package parts are tested in two stages. The support circuitry and CUTs are tested separately. Different test sets for the support circuitry and CUTs are described in 4.1.1 and 4.1.2 respectively.

4.1.1 Support Circuitry Test Sets

Table 7 lists all the support circuitry test sets and their descriptions. Test sets A, C, and D are functional tests that are manually generated by the designer. Test sets E, N and O are generated by commercial ATPG with 100% SSF coverage. They test all the input LFSR, response analysis circuitry as well as the output registers. Note that these test sets are generated with all output registers kept in the scan mode. This is to prevent the ATPG from propagating the signal through the CUTs. So no CUT faults can be detected by these test sets. The m12 BIST mapping logic is also tested.

Table 7. Support Circuitry Test Sets

Name	length	Generated by	Description
A,B	10,169	Designer	Verify the functionality of the support circuitry
C	2,006	Designer	Verify the functionality of the support circuitry
D	2,006	Designer	Verify the functionality of the support circuitry
E	824,513	ATPG	Test all 2901's input LFSR, response analysis circuitry, and output registers. Total 8 chains.
N	426,436	ATPG	Test all combinational CUTs' response analysis circuitry. Total 5 chains.
O	172,540	ATPG	Test all combinational CUTs' input LFSR as well as the m12 mapping logic. Total 5 Chains.
OS	208	Designer	I/O pins Continuity
P	-	LSI Tool	VIH and VIL
X	52,213	ATPG	100 I_{DDQ} measurements

All 17 support circuitry scan chains are tested by these test sets. However, none of the 8 internal scan chains of Lsi2901 and 2 internal scan chains of Tops2901 are tested by these test sets. These internal scan chains will be tested by the CUT test sets (see next section).

Test set OS was generated manually to test the I/O pin continuity. Test set P was generated by LSI Logic's in house tool to test the NAND tree as well as the V_{IH} and V_{IL}. Test set X was generated by ATPG. It contains 100 I_{DDQ} measurements of the support circuitry.

4.1.2 CUT Test Sets

Table 8 lists all the test sets and their descriptions. Many test sets have been included in an attempt to compare their effectiveness. Both academic and commercial ATPG tools have been used to generate these patterns. They are (alphabetically) Fastscan, IBM, Power Fault, Sunrise, Syntest, Tetramax, University of Illinois, University of Iowa. The verification patterns are provided by LSI Logic and the designers. The BIST technique is developed at CRC Stanford University and University of Texas. The tools are not identified by name since some of the patterns were only provided under these conditions. They are described in details as follows.

Single Stuck-at Fault (SSF) Tests (Tests 1.x)

Many different single stuck-at fault (SSF) test sets are included in this experiment because SSF is the most common fault model in the industry.

Compressed and uncompressed version of test sets are both included (1.1 and 1.1.c, 1.4 and 1.4.c). Besides, test set 1.8 generated by an academic tool which features a very efficient pattern compaction algorithm is also included. This is to investigate if the pattern compression impact the effectiveness of the test set.

For the sequential CUTs, both scan version and sequential version test sets are included (1.3 and 1.3.s, 1.4 and 1.4.s). Besides, test set 1.5 generated by an academic tool which features a very high coverage sequential pattern generation algorithm is also included. However, these test sets can not all achieve 100% coverage because of some states can not be reached.

Table 8 List of All CUT Test Sets

			CUT						
#	tool	Description	Lsi2901	Tops2901	Mult_adder	M12	Sqr	pb	internal
1.1	1	SSF 100%	319	504	96	69	43	3170	Ftsa
1.1.c	1	SSF 100% compressed	194	325	57	30	24	2711	Ftsac
1.2	2	SSF 100%			80	63	75	2227	Ibm
1.3	3	SSF 100%	156	314	86	63	33	2901	Sr
1.3.s	3	SSF, sequential	517	2237					Srseq
1.4	4	SSF 100%	193	386	62	30	17	2768	St
1.4.c	4	SSF 100% compressed	106	129	61	27	17	2749	Stc
1.4.80	4	SSF 80%	33	72	7	13	9	912	St80
1.4.90	4	SSF 90%	62	152	14	20	12	1401	St90
1.4.95	4	SSF 95%	91	230	52	25	13	1898	St95
1.4.99	4	SSF 99%	135	320	63	29	17	2532	St99
1.4.s	4	SSF, sequential	710						stseq
1.4.g	4old	SSF 100%, gate fault				163	74		lt
1.4.p	4old	SSF 100%, pin fault				61	34		ltp
1.5	5	SSF, sequential	888	1498					strategate
1.6.1	6	SSF 1 detect			37	42	27		Iowa
1.6.2	6	SSF 2 detect			76	99	59		Iowa2
1.6.3	6	SSF 3 detect			107	140	86		Iowa3
1.6.5	6	SSF 5 detect			158	212	141		Iowa5
1.6.10	6	SSF 10 detect			288	375	265		Iowa10
1.6.15	6	SSF 15 detect			436	544	373		Iowa15
1.7.3	7	SSF 3 detect						8205	Mer3
1.7.5	7	SSF 5 detect						20433	Mer5
1.7.15	7	SSF 15 defect				973	597		Mer15
1.8	8	SSF 100%			35	34	26	2589	ilker
1.11	11	SSF			785	391	136		Doremi
2.1	1	Transition 100%	653	799	359	213	157	12421	Fttr
2.1.c	1	Transition 100%, comp.	422	546	221	107	69	9217	Fttrc
2.2	2	Transition			521	928	457	9455	Ibmtr
2.3	3	Transition 100%	377	584	554	198	154	5496	Srtr
2.3.s	3	Transition, sequential	4070	1757					Srseqtr
2.9	9	Transition 100%	231	331	210	112	62	7738	Wtr
3.2	2	Path delay			252	261	157	5238	Ibmd
3.3	3	Path delay	66	316	112	2456	36	2982	Srpd
4.1	1	Toggle100%,compressed	20	33	19	12	9	215	Fttogc
5.2	2	I_{DDQ} Pseudo Stuck-at			49	36	25	310	Ibmq
5.3	3	I_{DDQ} PSA, 100%	134	164	48	21	17	387	Srq
5.4	4	I_{DDQ} PSA, compressed	48	31	28	17	14	19	Stq
5.10.p	10	I_{DDQ} PSA, select	20			6	6	194	Pqsa
5.10.b	10	I_{DDQ} Bridging, select	50			15	11	105	Prbr
6.0	-	Pseudo random exhaustive			2^24	2^24	2^24	2^24	Pr
6.2	2	Weighted random			173	182	66	2840	Ibmw
7.0	-	Verification	3121	429		57	90	4096	Ver
8.eb.1	-	Emulated BIST,100%			7168	75616	2560		Nur
8.eb.2	-	Emulated BIST				20000			Nur
8.hb	-	Hardware BIST 100%				20000			Nur

For the combinational CUTs, multiple detect patterns are included (1.6.x, 1.7.x) because Murphy experiment showed that these test sets are very effective.

Test patterns generated with gate faults are compared with pin faults. These patterns are the same as pattern 2.2 and 2.3 in the Murphy experiment.

Transition Tests (Tests 2.x)

Transition test sets of 100% or nearly 100% coverage are included. These test sets are effective in detecting timing defects (slow-to-rise, slow-to-fall) in the CUTs. Similarly, compressed version and sequential version of test sets are also included for comparison.

Path Delay Tests (Test 3.x)

Path delay detects timing defects in the paths. However, there are too many paths even in a small circuit (for example, sqr has $7x10^{15}$ structure paths). It is not possible to generate patterns for every path. In the case of test 3.2, ATPG automatically search for the testable paths. In the case of test 3.3, path list for mult_adder and m12 were generated by ATPG exhaustive path search. For the other 4 CUTs, their path lists were generated by a static-timing analyzer. The most critical paths starting from every primary inputs or ending at every primary outputs were selected.

I_{DDQ} Tests (Test 10.x)

Three ATPG tools were used to generate IDDQ patterns using pseudo stuck-at model. Test 5.3 achieved high coverage with less pattern compression. Test 5.4 compressed the pattern but the coverage is not as high as Test 5.3.

Besides I_{DDQ} pattern generator, one I_{DDQ} pattern selection tool was also used. This tool selected I_{DDQ} patterns from the verification patterns (Test 7.x). Test 5.10.p selected patterns based on the pseudo stuck-at model. Test 5.10.b selected patterns based on the bridging fault model.

Random Tests (Tests 6.x)

For test 6.0, the on-chip parallel data load LFSR is used to generate pseudo random patterns. The LFSR cycles through all $2^{24} - 1$ non-zero states. The all-zero

pattern is supplied externally by the tester. This test is exhaustive for the 24 inputs CUT (M12). This test is also N^2 super-exhaustive for the 12 inputs CUTs (PB and SQR). Super-exhaustive refers to all possible transitions or pairs of patterns. This is implemented by taking every second stage of the LFSR. The polynomial for the 24 bits LFSR is $x^{24}+x^7+x^2+1$. The 64 bits LFSR for mult_adder is $x^{65}+x^8+1$.

For test 6.2, weighted random patterns are externally applied by the tester.

Verification Tests (Test 7.0)

These patterns were manually generated by the designers to verify the functionality of the CUTs.

BIST (Test 8.x)

Tests 8.eb.1 and 8.eb.2 are emulated BIST patterns. They are applied externally from the tester. Test 8.eb.1 were generated by an LFSR with 100% SSF coverage. Test 8.eb.2 were generated by the same LFSR but test length is only 20K. This is to compare with Test 8.hb to show the effectiveness of the mapping logic. The primitive polynomials for the mult_adder, m12 and sqr are $x^{65}+x^{18}+1$, $x^{24}+x^7+x^2+x+1$, and $x^{12}+x^6+x^4+x+1$ respectively.

Test 8.hb is generated by 24 bits on-chip LFSR and is mapped into 100% SSF coverage patterns by the mapping logic. Compared with the 100% simulated BIST patterns, the test length is much shorter.

4.2 Test Flow

Fig. 8 shows the ELF35 test flow. Section 4.2.0 describes the wafer sort tests which include gross parametric tests and gross support circuitry tests. Section 4.2.1 describes the first stage package tests which include conventional gross parametric tests and support circuitry tests. Any chip that fails the first stage tests will not go on to the next stage. Section 4.2.2 describes the second stage package tests which include all CUT tests. Every test will continue to the end even if any failure occurs in the middle of the test. All failures in the stage two tests will be logged.

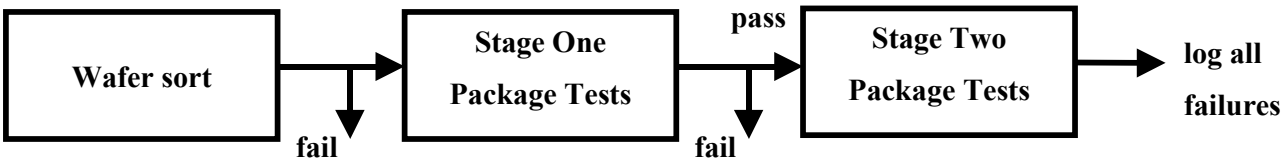

Figure 8 Test Flow

Only the data from the chips which passed the stage one package tests will be reported. Any chip that pass the first stage tests but fails some or all the second stage tests is called an "interesting die".

4.2.0 Wafer Sort Tests

Table 9 lists all the tests applied in the wafer sort testing. Tests 1-3 are gross parametric tests. Test 4 is gross support circuitry tests. Every test is described in details in this section. The exact numbers of test conditions are listed in Appendix D.

Table 9. Wafer Sort Tests

Test	Description
1	Power supply short
2	Continuity
3	VIH, VIL
4	Input Leakage
5	Support Circuitry Tests, nominal voltage, gross timing

Test 1. Power Supply Short

This test applies a very low supply voltage (0.5V) with all input pins open and measures the power supply current. If the supply current is larger than a limit (20mA) then the die is discarded because of the gross failure.

Test 2. Continuity (open and short)

This test tests the connection of every I/O pins as well as the ESD protection diode. For each I/O pin, the tester sinks a current (200 μA) and measures the voltage (with all the other pins grounded). If the measured voltage is lower than a lower limit (-1.5V), the pin is not connected to the ESD diode (open). If the measured voltage is higher than a upper limit (-0.2V), the pin is shorted to ground. Only the pin which produces a voltage between the upper and lower limit passes. Test set 2 which has a walking "z" across all I/O pins is applied (see Table 4.1.1). If any pin failed this test, the die is discarded.

Test 3. V_{IH}, V_{IL}

This test utilizes LSI logic's NAND tree to measure the V_{IH} and V_{IL} for every input pins. If any pin with a V_{IH} lower than a certain limit or a V_{IL} lower than a certain limit, this die is discarded. Test set P is applied (see Table 4.1.1).

Test 4. Input Leakage

For every input pin, force a voltage and measure the current. If the measured current exceeds a certain limit, the die is discarded.

Test 5. Support Circuitry Tests

This test applies all the test sets described in section 4.1.1 with a gross timing (very slow). If a die fails any test set, this die is discarded.

4.2.1 Stage One Package Tests

Table 10 lists all the tests applied in the stage one package testing. Every test is described in details as follows.

Table 10 Stage One Package Tests

Test	Description
1	Continuity
2	Input Leakage
3	Support circuitry Boolean tests, nominal voltage, fast timing
3.vlv	Support circuitry Boolean tests, very low voltage
4	Support circuitry I_{DDQ} tests, nominal voltage

Tests 1-2 Gross Parametric Tests

These tests are the same as what are done in the wafer sort testing (see section 4.2.0). They are repeated again here to make sure the chip is not damaged by packing.

Test 3. Support Circuitry Boolean Tests

The support circuitry is first tested at nominal voltage and fast timing. The fast timing is chosen to be the fastest speed at which the CUT can run. This is to make sure the support circuitry is defect free and fully functional at the fast timing.

Then the support circuitry is tested at a very low voltage. The timing is chosen to be the fastest speed at which the CUT can run. This is to make sure that support circuitry is fully functional at very low voltage.

Test 4. Support Circuitry I_{DDQ} Tests

The support circuitry is first testes at nominal voltage. All I_{DDQ} values are measured without any single threshold limit. This is to make sure that the support circuitry does not fail IDDQ testing.

4.2.2 Stage Two Package Tests

Table 11 lists all the tests applied in the stage two package testing.

Table 11 Stage Two Package Tests

Test	Description
1.nv.rated	CUT Boolean tests, nominal voltage, rated timing
1.nv.slow	CUT Boolean tests, nominal voltage, slow timing
1.nv.fast	CUT Boolean tests, nominal voltage, fast timing
1.vlv.rated	CUT Boolean tests, very low voltage, rated timing
1.vlv.slow	CUT Boolean tests, very low voltage, slow timing
1.vlv.fast	CUT Boolean tests, very low voltage, fast timing
2.nv	CUT I_{DDQ} tests, nominal voltage

Test 1. CUT Boolean Tests

All the Boolean test sets described in section 4.1.2 are applied in this test. All the primary outputs and scan outputs are not observed. Only the contents of the total failure counters and first failure counters are shifted out at the end of every test.

All test sets are applied at nominal voltage with three different speeds. The nominal timing is the worst case timing from the static timing analyzer. The fast timing is obtained from the shmoo plot. The slow timing is chosen to be three times slower than the nominal timing.

All test sets are then applies at a very low supply voltage. The very low voltage is chosen to be at two times as high as the transistor's threshold voltage. A shmoo is performed to determine the speed that the chip can run at very low voltage. The slow timing is chosen to be three times slower than the nominal timing.

Test 2. CUT I_{DDQ} Tests

All the I_{DDQ} test sets described in section 4.1.2 are applied in this test. The wait time before every I_{DDQ} measurement is about 7ms. All the I_{DDQ} values are logged with no single threshold limit. No failure counter content is recorded.

5. CONCLUSIONS

This report has described the ELF35 test chip experiment, including the design, architecture and functionality as well as the test sets and test flow. The experiment results will be presented in a future report.

Although the ELF35 experiment basically follows the Murphy experiment, the ELF35 experiment has the following distinct features:

- Sequential CUTs with full scan
- 0.35μm process technology
- New ATPG tools
- Hardware and emulated BIST

It is hoped that this experiment will produce many valuable results and contribute to our knowledge of the CMOS IC testing.

ACKNOWLEDGEMENTS

The authors would like to thank the following people for their precious help.

We would like to thank Mike Purtell and Advantest Inc. for their generous help in donating the tester time. This project requires enormous tester time and it would no be possible without their help.

We would also like to thank Scott Keller and LSI Logic Corp.

We would like to thank Marc Loranger, Fred Azima from Credence Incorporation, Dr. Sassan Raissi from Digital Testing Services, and Steven Liaw from ARTest. for their donation of tester time and technical support in testing the devices.

We would also like to thank the following persons for their donation of test patterns and ATPG tools:: Gary Greenstein, Ilker Hamzaoglu, Michael Hsiao, Rahit Kapur, Ray Mercer, John Waicucauski, and L.T. Wang.

We would also like to thank Nur Touba for his help in the design of BIST.

Last but not least, we would like to thank the following people in CRC for their help: Kan-Yuan Cheng, Yi-Chin Chu, Siyad Ma, Samy Makar, and Sanjay Wattal.

REFERENCES

[AMD 83] Advanced Micro Devices, Inc., "Bipolar Microprocessor Logic and Interface Data Book," pp.5-5to 5-27 1983.

[Avra 94] Avra, L., *Synthesis Techniques for Built-in Self-Testable Designs*, Ph.D. Thesis, Stanford University, Stanford, CA, Jul. 1994.

[Chang 96a]Chang, J., and E.J. McCluskey, "Quantitative Analysis of Very-Low-Voltage Testing," *VLSI Test Symposium*, pp.332-337, 1996.

[Chang 96b] Chang, J., and E.J. McCluskey, "Detecting Delay Flaws by Very-Low-Voltage Testing, " *Proc. ITC*, pp367-376, 1996.

[Chang 98a] Chang, J. and *et. al.*, "Experimental Results for I_{DDQ} and VLV Testing," VTS, pp.118-123, 1998.

[Chang 98b] Chang, J. and *et. al.,* "Analysis of Pattern-dependent and Timing-dependent Failures in an Experimental Test Chip," *Proc. ITC*, 1998.

[Franco 94] Franco, P. and *et. al.,* "An Experimental Chip to Evaluate Test Techniques, Part I: Description of Experiment," *Stanford University CRC, Technical Report,* No. 94-5, 1994.

[Franco 95] Franco, P. and *et. al.,* "An Experimental Chip to Evaluate Test Techniques Chip and Experiment Design," *Proc. ITC*, pp.653-662, 1995.

[Franco 96] Franco, P. and *et. al*., "Analysis and Detection of Timing Failures in an Experimental Test Chip," *Proc. ITC*, pp.691-700, 1996.

[Hao 93] Hao, H. and E. J. McCluskey, "Very-Low-Voltage Testing foe Weak CMOS Logic IC," *Proc. ITC*,pp.275-284, 1993.

[Ma 95] Ma, S.C., P. Franco, and E.J. McCluskey, " An Experimental Chip To evaluate Test Techniques Experimental Results," *Proc. ITC*, pp.663-672, 1995.

[Maxwell 96] Maxwell, P.C. *and et. al*., "I_{DDQ} and AC Scan: The War Against Unmodelled Defects," *Proc. ITC,* pp.250-258, 1996.

[Norwood 97] Norwood, R., *Synthesis-for-Scan: Reducing Scan Overhead with High-Level Synthesis*, Ph.D. Thesis, Stanford University, Stanford, CA, Dec. 1997.

[Sematech 97] Nigh, P. and et. al., "So What Is an Optimal Test Mix? A Discussion of the SEMATECH Methods Experiment," *Proc. ITC*, pp1037-1038, Nov.1997.

[Powell 96] Powell T.J. and *et. al*., "Correlating Defects to Functional and I_{DDQ} Tests," *Proc. ITC*, pp.501-510, 1996.

[Righter, 98] Righter and *et. al.*, "CMOS IC Reliability Indicators and Burn-In Economics, " *Proc. ITC*,1998.

[Touba 96] Touba, N., *Synthesis Techniques for Pseudo-Random Built-In Self-Test*, Ph.D. Thesis, Stanford University, Stanford, CA, June 1996.

Appendix A: Test Chip Data Sheet

Part number	**LXA0315**
Process	**LSI Logic LCBG10p**
Package	**PBGA (code ig56)**
Die code	**0pe8**
Total Gate Count	**264K (CUT 217K, support 47K)**

A.1 Test Chip Input/Output Signals

Input Pins: 178 (4 global pins, 84 2901 pins, 90 combinational cut pins)
Output Pins: 30 (17 2901 pins, 11 combinational cut pins, 2 misc. pins)
Power Supply Pins: 64 (44GND, 20 Vdd)

GLOBAL INPUT SIGNALS

clk: clock

iddtnin: iddtn buffer control signal
1: turn off all pull transistors
0: turn on all pull transistors

clkmode: 1. clocking mode selection, should be 0 for normal clock frequency should be 1 for doubled clock frequency.

2. clkmode can also be used to control the input of ti2901 (the scan in signal for 2901).If used in conjuction with andsel[0], it can be used to switch between debug mode and normal mode for 2901. It should be 0 for normal operation, 1 for debug mode operation. In the debug mode, the scan out of a 2901 can be observed from chip output.

clkmode = 1 => ti2901
clkmode = 0 & imode = 0 => ti2901
clkmode = 0 & imode = 1 => Din2901

imode: input mode for all CUTs
0: parallel load
1: input LFSR in the shift mode
Should be used in conjunction with en and telfsr

telfsr	imode	en	
1	1	1	scan chain
0	1	1	pseudo-random
0	0	1	parallel load
1	0	1	xxx (no use)
x	x	0	hold

INPUT SIGNALS FOR 2901S

Din2901[57:0]: data in for two 2901s. LSI2901 uses all 58 signals. TOPS2901 uses only Din2901[53:0].

lsi2901

Din2901[57:49]=i
Din2901[48]=aadr3
Din2901[47]=aadr2
Din2901[46]=aadr1

Din2901[45]=aadr0
Din2901[44]=badr3
Din2901[43]=badr2
Din2901[42]=badr1
Din2901[41]=badr0
Din2901[40:9]=d
Din2901[8]=rami31
Din2901[7]=rami16
Din2901[6]=rami15
Din2901[5]=rami0
Din2901[4]=qi31
Din2901[3]=qi16
Din2901[2]=qi15
Din2901[1]=qi0
Din2901[0]=cn

tops2901
Din2901[53:45]=i
Din2901[44]=aadd3
Din2901[43]=aadd2
Din2901[42]=aadd1
Din2901[41]=aadd0
Din2901[40]=badd3
Din2901[39]=badd2
Din2901[38]=badd1
Din2901[37]=badd0
Din2901[36:5]=d
Din2901[4]=c0
Din2901[3]=q31in
Din2901[2]=ram31in
Din2901[1]=q0in
Din2901[0]=ram0in

enoutfc2901[1:0]: 2901 output failure counter enable.
enoutfc2901[0]: LSI2901
enoutfc2901[1]: TOPS2901
Should be 1 for enable, 0 for disable.

teout2901: 2901 output failure counter shift enable
Both LSI2901 and TOPS2901 share the same signal.
Should be 1 for enable, 0 for disable.

tioutfc2901[1:0]: scan in for 2901 output failure counters
tioutfc2901[0]: LSI2901
tioutfc2901[1]: TOPS2901

enscanoutfc2901[1:0]:
2901 scan output failure counter enable
enscanoutfc2901[0]: LSI2901
enscanoutfc2901[1]: TOPS2901
Should be 1 for enable, 0 for disable.

tiscanoutfc2901[1:0]:
scan in for 2901 scan output failure counter
tiscanoutfc2901[0]: LSI2901
tiscanoutfc2901[1]: TOPS2901

enlfsrfc2901[1:0]: 2901 lfsr comparator enable.
enlfsrfc2901[0]: LSI2901
enlfsrfc2901[1]: TOPS2901
Should be 1 for enable, 0 for reset and disable.

en2901[1:0]: 2901 input signal enable.
en2901[0]: LSI2901
en2901[1]: TOPS2901
0: hold the previous data
1: input data are valid

telfsr2901: scan enable for 2901 input lfsr.
1: set the lfsrs into the scan mode
0: other modes

tilfsr2901[1:0]: scan in for 2901 input lfsr
tilfsr2901[0]: LSI2901
tilfsr2901[1]: TOPS2901

teoutreg2901: scan enable for 2901 output registers.
Should be 1 for enable, 0 for disable

tioutreg2901[1:0]: scan in for 2901 output registers.
tioutreg2901[0]: LSI2901
tioutreg2901[1]: TOPS2901

te2901[1:0]: 2901 scan enable
te2901[0]: LSI2901
te2901[1]: TOPS2901

ti2901[1:0]: 2901 scan in
ti2901[0]: LSI2901
ti2901[1]: TOPS2901

andsel[2:0]: and tree select
A => andsel[0]
B => andsel[1]
C => andsel[2]

A	B	C	
0	0	0	lsi2901 te2901
1	0	0	lsi2901 ti2901
X	1	0	lsi2901 teoutreg
0	0	1	tops2901 te2901
1	0	1	tops2901 ti2901
X	1	1	tops2901 teoutreg

andsel signals are also used as the control signals of other functions

andsel[0]: select signal of the muxes that were added to observe the scanout signal of each 2901. should be 0 during normal data collection. Should be 1 when we want to observe the scanout signal.

andsel[1]: can be used to invalidate the mask signals for lsi2901 outputs. Should be 1 when we want to invalidate the mask signals, 0 otherwise.

andsel[2]: can be used to invalidate the mask signals for tops2901 outputs. Should be 1 when we want to invalidate the mask signals, 0 otherwise. please refer to Din for the information of mask signals.

INPUT SIGNALS FOR OTHER CUTS

Din[64:0]: input bits for multiplier-adder, m12, 6sq, and pb.

Din[64:0]: multiplier-adder
ci=Din[64]
Din[63:48]=ina0[15:0]
Din[47:32]=inb0[15:0]
Din[31:16]=ina1[15:0]
Din[15:0]=inb1[15:0]

Din[23:0]: m12
Din[23:12]=a[11:0], Din[11:0]=b[11:0]

Din[47:24]: 6sq
Din[46]=a[5]
Din[44]=a[4]
Din[42]=a[3]
Din[40]=a[2]
Din[38]=a[1]
Din[36]=a[0]
Din[34]=b[5]
Din[32]=b[4]
Din[30]=b[3]
Din[28]=b[2]
Din[26]=b[1]
Din[24]=b[0]

Din[others]=0

Din[64:41]: pb
Din[63]=a[11]
Din[61]=a[10]
Din[59]=a[9]
Din[57]=a[8]
Din[55]=a[7]
Din[53]=a[6]
Din[51]=a[5]
Din[49]=a[4]
Din[47]=a[3]
Din[45]=a[2]
Din[43]=a[1]
Din[41]=a[0]
Din[others]=0

Din signals are also used as the output mask signals
for two 2901s.

Din[62:0] are used as the output mask signals for lsi2901.
Din[46:0] are used as the output mask signals for tops2901.

All output mask signals can be invalidated by setting proper values to andsel[2:1]. Both Din[x] and andsel[x] should be 0 to be able to mask an output bit. If one of them is one, the output signal will be passed to the response analysis circuitry

enoutfc[3:0]: output failure counter enable
enoutfc[0]: multiplier-adder
enoutfc[1]: m12
enoutfc[2]: 6sq
enoutfc[3]: pb
1 for enable, 0 for disable.

teoutfc: output failure counter shift enable
for multiplier-adder, m12, 6sq, and pb
1 for enable, 0 for disable.

tioutfc[3:0]: scan in for output failure counter
tioutfc[0]: multiplier-adder
tioutfc[1]: m12
tioutfc[2]: 6sq
tioutfc[3]: pb

enlfsrfc[3:0]: lfsr comparator enable
enlfsrfc[0]: multiplier-adder
enlfsrfc[1]: m12
enlfsrfc[2]: 6sq
enlfsrfc[3]: pb
1 for enable, 0 for disable and reset

en[3:0]: input signal enable
en[0]: multiplier-adder
en[1]: m12
en[2]: 6sq
en[3]: pb
0: hold the previous data
1: input data are valid

telfsr: scan enable for lfsr
for multiplier-adder, m12, 6sq, and pb
1 for enable, 0 for disable

tilfsr[3:0]: scan in for lfsr
tilfsr[0]: multiplier-adder
tilfsr[1]: m12
tilfsr[2]: 6sq
tilfsr[3]: pb

bistmode: bist mode enable for m12
Should be 1 for enable, 0 for disable

teoutreg: scan enable for output registers
Should be 1 for enable, 0 for disable

tioutreg: scan in for output registers
the output registers of multiplier-adder, m12, 6sq,
and pb are linked into a single scan chain.

OUTPUT SIGNALS FOR 2901S

sooutreg2901[1:0]: scan out for 2901 output registers
sooutreg2901[0]: LSI2901
sooutreg2901[1]: TOPS2901

fullout2901[3:0]: 2901 output failure counter full
fullout2901[1:0]: LSI2901
fullout2901[3:2]: TOPS2901
1: full
0: otherwise

fullscanout2901[3:0]:
2901 scan failure counter full
fullscanout2901[1:0]: LSI2901
fullscanout2901[3:2]: TOPS2901
1: full
0: otherwise

IFO2901[1:0]: 2901 lfsr and lfsr comparator shift out
IFO2901[0]: LSI2901
IFO2901[1]: TOPS2901

FO2901[1:0]: 2901 failure counter shift out
FO2901[0]: LSI2901
FO2901[1]: TOPS2901

FO2901[1] is also used as the fail indicator for LSI2901, it is connected to the output of the OR tree in the comparators. teout2901 should be used to switch between two functions.

teout2901 should be 1 when shifting out the value in the failure counter. Should be 0 otherwise.

FOscan2901[1:0]: 2901 scan failure counter shift out
FOscan2901[0]: LSI2901
FOscan2901[1]: TOPS2901

FOscan2901[1] is also used as the fail indicator for TOPS2901, it is connected to the output of the OR tree in the comparators. teout2901 should be used to switch between two functions.

teout2901 should be 1 when shifting out the value in the failure counter. Should be 0 otherwise.

andout: and tree output
to test te and ti of 2901s

SHARED OUTPUT SIGNALS FOR 2901S

Tops2901

fullscanout2901[3] -> sotops2901[1]
fullscanout2901[2] -> sotops2901[0]

Lsi2901

fullout2901[3] -> so2901[7]
fullout2901[2] -> so2901[6]
fullout2901[1] -> so2901[5]
fullout2901[0] -> so2901[4]
fullscanout2901[1] -> so2901[3]
fullscanout2901[0] -> so2901[2]
FO2901[0] -> so2901[1]
FOscan2901[0] -> so2901[0]

OUTPUT SIGNALS FOR OTHER CUTS

sooutreg: scan out for output registers

FO[3:0]: CUT failure counter outputs
- FO[0]: multiplier-adder
- FO[1]: m12
- FO[2]: 6sq
- FO[3]: pb

FO[3:0] are also used as the fail indicator for combinational CUTs. teoutfc should be used to switch between two functions.

teoutfc should be 1 when shifting out the value in the failure counter. Should be 0 otherwise.

IFO[3:0]: lfsr failure counter outputs
- IFO[0]: multiplier-adder
- IFO[1]: m12
- IFO[2]: 6sq
- IFO[3]: pb

full[1:0]: multiplier-adder failure counter full signals
- 1: full
- 0: otherwise

OTHER OUTPUT SIGNALS

procout: process monitor output
clkout: output signal for clk doubler

A.2 CUT SIZE & UTILIZATION & POWER RING WIDTH

pb_std: 2599.800um, 3099.600um, 16.32%, 20um
top_bfa0012af: 1073.000um, 1603.000um, 60.38%, 22um
top_a2901scan: 1300.000um, 2000.000um, 56.43%, 20um
top_m12: 238.000um, 833.000um, 52.94%, 2.8um
top_6sq: 238.000um, 338.800um, 55.93%, 2.8um
top_mult16_adder32: 885.000um, 700.000um, 56.93%, 8um
chip-level information:
megacells occupy 49.25% of the center region
megacells (including routing area) occupy 74.30% of the center region

cell columns (excluding megacells and placeblocks) occupy 42.06% of the center region (excluding megacells and placeblocks)
Total megacell height= 1790460
chip utilization (excluding megacells)= 6.48

chip utilization (including megacells)= 52.54

A.3 POWER ESTIMATION

250K x 0.2 x 100MHz x 1uW/MHz = 5W
vdd2/vss2 pairs: 37(core logic)
vdd/vss pairs: 16(I/O)

Appendix B Scan Chain Description

The ELF35 test chip has 17 scan chains in the support circuitry, 8 CUT scan chains in Lsi2901, and 2 CUT scan chains in Tops2901. All the scan chains are listed in Table B.

11 of the 17 scan chains in the support circuitry are for output response analysis. They are described in section B.1. The other 6 of the 17 scan chains are input LFSR. They are described in section B.2.

Table B List of All Scan Chains

Scan in	Scan enable	SCAN OUT	# of FFs	Description	Test set
tioutfc2901[0]	teout2901	FO2901[0]	112	LSI2901 PO FC	E Chian1
tioutfc2901[1]	teout2901	Fo2901[1]	98	TOPS2901 PO FC	E Chain2
Tiscanoutfc2901[0]	teout2901	FOscan2901[0]	112	LSI2901 Scan FC	E Chain3
Tiscanoutfc2901[1]	teout2901	FOscan2901[1]	98	TOPS2901 Scan FC	E Chain4
tilfsr2901[0]	telfsr2901	IFO2901[0]	117	LSI2901 input LFSR	E Chain5
tilfsr2901[1]	telfsr2901	IFO2901[1]	109	TOPS2901 input LFSR	E Chain6
tioutreg2901[0]	teoutreg2901	sooutreg2901[0]	64*8=512	LSI2901 output reg.	E Chain7
tioutreg2901[1]	teoutreg2901	sooutreg2901[1]	48*2=96	TOPS2901 output reg.	E Chain8
tioutfc[0]	Teoutfc	FO[0]	104	m&a failure counter	N Chain9
tioutfc[1]	Teoutfc	FO[1]	128	m12	N Chain10
tioutfc[2]	Teoutfc	FO[2]	128	6sq	N Chain11
tioutfc[3]	Teoutfc	FO[3]	98	Pb	N Chain12
tilfsr[0]	Telfsr	IFO[0]	131	m&a input LFSR	O Chain13
tilfsr[1]	Telfsr	IFO[1]	49	m12	O Chain14
tilfsr[2]	Telfsr	IFO[2]	49	6sq	O Chain15
tilfsr[3]	Telfsr	IFO[3]	49	Pb	O Chain16
Tioutreg	Teoutreg	sooutreg	33*4c=132 12*16c=192 6*16c=96 12*2c=24 444	m&a output register m12 6sq pb total	N Chain17 N Chain17
ti2901[0]	te2901[0]	fullout2901[3]	544	LSI 2901 internal	CUT test sets
	te2901[0]	fullout2901[2]	544		
	te2901[0]	fullout2901[1]	544		
	te2901[0]	fullout2901[0]	544		
	te2901[0]	fullscanout2901[1]	544		
	te2901[0]	fullscanout2901[0]	544		
	te2901[0]	FO2901[0]	544		
	te2901[0]	Foscan2901[0]	544		
ti2901[1]	te2901[1]	Fullscanout2901[3]	961	tops 2901 internal	CUT test sets
	te2901[1]	Fullscanout2901[2]	961		

B.1 Scan Chains of the Response Analysis Circuitry

Scan Chain ordering

Lsi2901, Mult_adder, M12, 6sq

ti -> freeze flipflop (1st copy) ->
failure location (LSB -> MSB) (1st copy) ->
total failure (MSB -> LSB) (1st copy) ->
first failure (MSB -> LSB) (1st copy) ->
freeze flipflop (2nd copy) ->
failure location (LSB -> MSB) (2nd copy) ->
total failure (MSB -> LSB) (2nd copy) ->
first failure (MSB -> LSB) (2nd copy) -> so

tops2901, pb

ti -> freeze flipflop (1st copy) ->
total failure (MSB -> LSB) (1st copy) ->
first failure (MSB -> LSB) (1st copy) ->

freeze flipflop (2nd copy) ->
total failure (MSB -> LSB) (2nd copy) ->
first failure (MSB -> LSB) (2nd copy) -> so

combinational CUTs output register (444 FFs overall)

tioutreg -> sum31..sum0,co	33bits * 4c mult_adder
-> m11..m0	12bits * 16c m12
-> m5..m0	6bits * 16c sqr
-> z11..z0	12bits * 2c pb
-> sooutreg	

Expected outputs

CUT, (the first bit is shifted out in the 0th cycle)

Lsi2901: (112 flipflops overall) FO2901_0 & FOscan2901_0
0 -> 23: x (first failure count, LSB->MSB)
24 -> 47: L (total failure count, LSB->MSB)
48 -> 54: L (failure location)
55 : x
56 -> 79: x (first failure count, LSB->MSB)
80 -> 103: L (total failure count, LSB->MSB)
104 -> 110: L (failure location)
111 : x

Tops2901: (98 flipflops overall) FO2901_1 & FOscan2901_1
0 -> 23: x (first failure count, LSB->MSB)
24 -> 47: L (total failure count, LSB->MSB)
48 : x
49 -> 72: x (first failure count, LSB->MSB)
73 -> 96: L (total failure count, LSB->MSB)
97 : x

mult_adder: (104 flipflops overall)
0 -> 23: xx (first failure count, LSB->MSB)
24 -> 47: L (total failure count, LSB->MSB)
48 -> 50: L (failure location)
51 : x
52 -> 75: xx (first failure count, LSB->MSB)
76 -> 99: L (total failure count, LSB->MSB)
100 -> 102: L (failure location)
103 : x

m12: (128 flipflops overall)
0 -> 23: xx (first failure count, LSB->MSB)
24 -> 47: L (total failure count, LSB->MSB)
48 -> 62: L (failure location)
63 : x
64 -> 87: xx (first failure count, LSB->MSB)
88 -> 111: L (total failure count, LSB->MSB)
112 -> 126: L (failure location)

127 : x

6sq: (128 flipflops overall)
0 -> 23: xx (first failure count, LSB->MSB)
24 -> 47: L (total failure count, LSB->MSB)
48 -> 62: L (failure location)
63 : x
64 -> 87: xx (first failure count, LSB->MSB)
88 -> 111: L (total failure count, LSB->MSB)
112 -> 126: L (failure location)
127 : x

pb: (98 flipflops overall)
0 -> 23: xx (first failure count, LSB->MSB)
24 -> 47: L (total failure count, LSB->MSB)
48 : x
49 -> 72: xx (first failure count, LSB->MSB)
73 -> 96: L (total failure count, LSB->MSB)
97 : x

Scan Chain ordering

tilfsr -> comparator
lfsr (1st copy) ->
lfsr (2nd copy) -> so

Expected outputs

The expected output of the input register of each CUT, (the first bit is shifted out in the 0th cycle)

Lsi2901: (117 flipflops overall)
0 -> 115: x
116 : L (L means two copies of LFSR always match)

Tops2901: (109 flipflops overall)
0 -> 107: x
108 : L

mult_adder: (131 flipflops overall)
0 -> 129: x
130 : L

m12: (49 flipflops overall)
0 -> 47: x
48 : L

6sq: (49 flipflops overall)
0 -> 47: x
48 : L

pb: (49 flipflops overall)
0 -> 47: x
48 : L

Appendix C: Pin Assignments

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Y	gnd	vdd	teout 2901	telfsr 2901	enscan outfc 2901(1)	tilfsr 2901 (1)	andsel (0)	ti 2901 (1)	fullout 2901 (3)	vdd	gnd	foscan 2901 (1)	fullout 2901 (1)	ifo 2901 (0)	foscan 2901 (0)	vdd	gnd	vdd	soout reg 2901(0)	gnd	Y
W	din 2901 (52)	din 2901 (54)	gnd	din 2901 (55)	tioutfc 2901 (1)	en 2901 (1)	te2901 (1)	andsel (2)	teout reg 2901	fullscan out 2901(2)	gnd	ifo 2901 (1)	fullscan out 2901(0)	fo 2901 (0)	tioutfc 2901 (0)	vdd	imode	tiscan outfc 2901(0)	vdd	te 2901 (0)	W
V	din 2901 (48)	din 2901 (51)	din 2901 (53)	din 2901 (56)	enoutfc 2901 (1)	tiscan outfc 2901(1)	tiout reg 2901(1)	andsel (1)	fullout 2901 (2)	fullscan out 2901(3)	clk	fullout 2901 (0)	fullscan out 2901(1)	enoutfc 2901 (0)	tilfsr 2901 (0)	andout	enscan outfc 2901(0)	tiout reg 2901(0)	enlfsrfc 2901 (0)	soout reg	V
U	din 2901 (45)	din 2901 (47)	din 2901 (50)	gnd	din 2901 (57)	vdd	enlfsrfc 2901 (1)	gnd	soout reg 2901(1)	vdd	fo 2901 (1)	gnd	gnd	gnd	vdd	gnd	gnd	fo (3)	en 2901 (0)	enoutfc (3)	U
T	din 2901 (41)	din 2901 (44)	din 2901 (46)	din 2901 (49)													ifo (3)	tioutfc (3)	enlfsrfc (3)	tilfsr (3)	T
R	din 2901 (38)	din 2901 (40)	din 2901 (43)	vdd													vdd	en (3)	enoutfc (1)	enoutfc (2)	R
P	din 2901 (36)	din 2901 (37)	din 2901 (39)	din 2901 (42)													iddtnin	teoutfc	tioutfc (1)	tioutfc (2)	P
N	din 2901 (33)	din 2901 (34)	din 2901 (35)	gnd													gnd	enlfsrfc (1)	enlfsrfc (2)	clk mode	N
M	din 2901 (29)	din 2901 (30)	din 2901 (31)	din 2901 (32)													en (1)	en (2)	tilfsr (1)	ti 2901 (0)	M
L	din 2901 (26)	din 2901 (27)	din 2901 (28)	gnd													vdd	tilfsr (2)	gnd	bist mode	L
K	gnd	din 2901 (24)	din 2901 (25)	vdd													ifo (2)	ifo (1)	fo (2)	fo (1)	K
J	din 2901 (23)	din 2901 (22)	din 2901 (21)	din 2901 (20)													din (45)	din (44)	din (43)	din (42)	J
H	din 2901 (19)	din 2901 (18)	din 2901 (17)	gnd													gnd	din (48)	din (47)	din (46)	H
G	din 2901 (16)	din 2901 (15)	din 2901 (14)	din 2901 (11)													din (55)	din (52)	din (50)	din (49)	G
F	din 2901 (13)	din 2901 (12)	din 2901 (10)	vdd													vdd	din (56)	din (53)	din (51)	F
E	din 2901 (9)	din 2901 (8)	din 2901 (7)	din 2901 (4)													din (62)	din (59)	din (57)	din (54)	E
D	din 2901 (6)	din 2901 (2)	din 2901 (3)	gnd	telfsr	vdd	din (39)	gnd	din (29)	fo (0)	vdd	din (20)	gnd	din (11)	vdd	din (4)	gnd	proc out	din (60)	din (58)	D
C	din 2901 (5)	din 2901 (1)	gnd	tilfsr (0)	tioutfc (0)	din (40)	din (36)	din (32)	full (1)	gnd	din (25)	din (21)	din (17)	din (14)	din (10)	din (7)	din (3)	din (64)	clkout	din (61)	C
B	din 2901 (0)	tiout reg	teout reg	enlfsrfc (0)	din (41)	din (37)	din (34)	din (31)	full (0)	din (28)	din (24)	din (22)	din (18)	din (15)	din (12)	din (8)	din (2)	din (1)	vdd	din (63)	B
A	gnd	vdd	en (0)	enoutfc (0)	din (38)	din (35)	din (33)	din (30)	ifo (0)	din (27)	din (26)	din (23)	din (19)	din (16)	din (13)	din (9)	din (6)	din (5)	din (0)	gnd	A
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	

	9	10	11	12	
M	gnd	gnd	gnd	gnd	M
L	gnd	gnd	gnd	gnd	L
K	gnd	gnd	gnd	gnd	K
J	gnd	gnd	gnd	gnd	J
	9	10-	11	12	

Appendix D: Test Conditions

D.1 Wafer Sort

Table D1 lists the test limits of every test in the wafer sort.

Table D1. Test limits of wafer sort

test	Description	force	limit	timing
1	Power supply short	Vdd = 0.5V	20mA	-
2	Continuity	200µA	if voltage >-0.3V, short if voltage <-1.5V, open	-
3	VIH/VIL	3.3V, 0V	0.69V (VIL) 2.41V (VIH)	-
4	Input leakage	3.3V, 0V	1µA	-
5	Support circuitry. Boolean tests	Vdd = 3.3V VIH = 3.3V VIL = 0V	-	cycle time =1µS clock rise = 250nS clock fall = 750nS strobe = 993nS

D.2 Package Test

Table D2 lists the timing of every CUT. A shmoo testing was performed on ten randomly selected packaged chips. The rated timing in Table D2 is obtained by adding 30% to the shmoo results. The fast timing is obtained by adding 20% to the shmoo results. The slow timing is three times slower than the rated timing.

The reason why the PB is slower than the design speed is that there exist a critical path which is caused by a very long wire. The support circuit speed is limited by the scan chain shifting and slew rate of some output pins.

Table D2. Timing of CUTs (nS)

CUT	Nominal voltage (3.3V)			Very low voltage (1.4V)			simulation
	fast	rated	slow	fast	rated	slow	
SQR	16.8	18.2	54.6	88	104	312	14
M12	13.2	14.3	42.9	84	91	273	12.5
MA	16.8	18.2	54.6	72	78	234	12
PB	50.4	54.6	163.8	216	234	702	28.37
LSI2901	33.6	36.4	109.2	132	143	429	15.2
TOPS2901	28.8	31.2	93.6	120	130	390	18.6
support	28.8	-	-	148.8	-	-	NA

All the shift operations are 1µS. The clock rises at 25% of the cycle and falls at 75% of the cycle time. The outputs are strobed at 95% of the cycle time.

The pseudo-random/exhaustive and BIST tests are applied only at slow speed.

The I_{DDQ} waiting time between applying a test vector and measuring current is 10mS.