I_{DDQ} Testing of Input/Output Resources of SRAM-based FPGAs [*]

Lan Zhao
Lucent Technologies
1247 S Cedar Crest Blvd
Allentown PA 18103

D. M. H. Walker
Computer Science
Texas A&M University
College Station TX 77843

Fabrizio Lombardi
Elec. & Computer Engineering
Northeastern University
Boston MA 02115

Abstract

This paper presents a quiescent current-based (I_{DDQ}) approach for testing input/output resources in SRAM-based FPGAs. Input/output resources include input/output blocks (IOBs) and the I/O interconnect. Test generation and application strategies are proposed by taking into account the limited controllability of the I/O resources. Configuration of these resources requires that the test stimuli must be provided by internal (logic and routing) resources. A detailed presentation for testing the I/O resources of the Xilinx XC4000 family is given.

1 Introduction

The recent growth in FPGAs (mainly logic resources) has not been matched by an equivalent increase in input/output resources (Input/Output Blocks, IOBs and I/O interconnect). Also, no work has been reported in the technical literature on I/O testing. I/O testing is important for many reasons: (1) The IOBs and the I/O interconnect provide controllability and observability for all internal resources. (2) Specific defects and faults may only affect the IOBs.

In [4] and [5], different I_{DDQ}-based testing strategies have been proposed for the detection of bridging faults in SRAM-based (reprogrammable) FPGA logic and routing resources, respectively. In this paper, we will discuss testing of I/O resources in FPGAs.

2 Previous Work

An FPGA consists of two types of resources: internal resources (logic resources, routing resources) and I/O resources. Testing of logic resources has recently been studied in depth. Many papers have dealt with testing of interconnects in FPGAs [1], [2]. For example, the *counting sequence* algorithm can be used to detect all bridging (short) faults in a wiring network with a test length of $\lceil log_2 n \rceil$, where n is the number of nets. This can be modified to exclude the all-0 and all-1 test vectors to also cover stuck-at faults. Therefore, the number of test patterns is $\lceil log_2(n+2) \rceil$. I_{DDQ} has been proposed in [4] and [5] for testing logic and routing resources of configurable FPGAs. In [4], tests and configurations of the FPGA are analyzed and generated hierarchically. [5] has considered I_{DDQ} testing of routing resources. [5] tests the programmable interconnect by using a modular approach; it has been proved that the whole interconnect can be tested in a constant number of programming phases. Test generation is based on the definition of the control and connection nets.

3 Preliminaries

The general architecture of the FPGA under test consists of a two-dimensional grid of identical configurable logic blocks (CLBs) with routing resources (including switch blocks and connection blocks) running between, and surrounded by I/O resources. The I/O resources consist of two types: the user programmable input/output blocks (IOBs) and the associated I/O interconnect. The IOBs provide the interface between the external package pins and the internal logic. Each IOB controls one package pin and can be configured for input, output, or bi-directional signals. The I/O interconnect connects the IOBs with the internal logic blocks through internal switch blocks and connection blocks. The I/O interconnect consists of a number of peripheral switch blocks (denoted as PSBs) located on each edge of the chip. The I/O interconnect basically forms a ring around the outside of the CLB array (this is referred to as the VersaRing in the Xilinx XC4000 series). A high-level diagram of the I/O resources for Xilinx-type FPGAs is given in Figure 1 [3]. The I/O resources are given by the IOBs and the PSB (shown for the left edge of the FPGA). There are two IOBs at both ends of each row (or column). The dark colored interconnect forms a PSB. The light colored interconnects are internal interconnects, which are tested while

[*] This research supported in part by grants from the Texas Advanced Research and Technology Programs.

testing the switch blocks [5]. WED denotes the wide edge decoder and is used for routing along the periphery of the chip.

There are two types of nets in the FPGA interconnect resources: *connection nets*, which are the wires running inside and between blocks, and *control nets*, which are the nets that run the control bits from the SRAM configuration memory to the switches and multiplexers. The vectors applied to the control nets are called control vectors. Based on the common assumption of a hierarchical FPGA architecture, an appropriate model has been developed for bridging faults [4]. At the block level, only bridging faults inside each block (which are referred to as *intra-block faults*) are considered. Logically adjacent blocks are assumed to be physically adjacent on the chip, and bridging faults between their external lines are considered. In the logic resources, a logic block or an I/O block consists of different types of modules (multiplexers, flip-flops, etc.). The intra-block faults include the faults inside the modules, referred to as *intra-module faults*, and faults between the modules, referred to as *inter-module faults*.

4 Basic Principles

The proposed approach considers test generation and application for two types of I/O resources: the IOBs and the I/O interconnect. As these resources have limited controllability, testing must consider the internal (logic and routing) resources; optimally, it should be possible to test each resource only once, thus reducing the number of programming phases too. The proposed approach is therefore, based on the following basic criteria: (1) The number of I/O resources (and in particular the grouping of IOBs within a PSB) limits the parallelism by which these resources can be tested simultaneously. (2) Propagation of the tests from an input to an output IOB involves internal resources. (3) The detailed structure of the I/O resources is usually vendor-specific; however, many problems (such as the I/O interface to a pad) are common across many FPGA architectures.

5 Testing the I/O Blocks

5.1 Test Generation Strategy

Basically there are three types of modules in an IOB: 2-to-1 multiplexers, D flip-flops, and logic gates, e.g. inverters. The multiplexers are controlled through the configuration software [3]. In this paper, the control and test sets are generated in a bottom-up fashion as follows:

Input: the Test Library. This library is built using a circuit level description of the modules using known implementations (e.g. 2-to-1 MUX, inverter). The circuit level diagrams are used to illustrate our testing approach. They are based on the logic-level diagrams given in the databook [3].

Step 1: Generate the Control Set. Establish the configurations of the multiplexers, by considering all bridging faults between control nets.

Step 2: Generate the Test Set. This consists of two further substeps:

Substep 2.1 Generate tests for the configurable modules. These tests are generated using the test library. Once the configurations of the control nets are established as in Step 1, the test set for the intra-module faults are generated in accordance with their control vectors. Additional control vectors may be required. The configurations and tests for the configurable multiplexers are generated first to reduce the number of programming phases (as they dominate the test time).

Substep 2.2. The remaining intra-block and inter-block faults are determined through fault simulation. The analysis is based on node equivalence classes [4]. Additional test vectors are generated for not yet detected faults, if necessary.

5.2 The Test Library

The test library has been discussed in previous papers [4], [5] to which the interested reader should refer for further details. In this section, testing of internal bridging faults in each of the modules is discussed. Special emphasis is given on reducing the number of programming phases. Two modules (i.e. the 2-to-1 multiplexer and the inverter) are covered in detail for sake of completeness.

1. *Testing a 2-to-1 Multiplexer.* In each 2-to-1 multiplexer (Figure 2), there are 5 nodes. Hence, there can be at most C(5,2) = 10 bridging faults. Using a modified counting sequence, at least $\lceil log_2(5+2)\rceil = 3$ test vectors are needed. Also, two phases ($s = 1$ and $s = 0$, where s is the select signal) are inevitable to detect the two bridging faults between the data inputs (D_0 and D_1) and the output.

2. *Testing an Inverter.* After fault collapsing, only stuck-at-0/1 faults at the input line must be considered for an inverter. These faults can be detected by two test vectors (0 and 1). These two test vectors also cover all non-redundant intra-transistor bridging faults.

5.3 An Example

As an example, the IOB of the XC4000 series [3] is shown in Figure 3. An implementation of the D flip-flop is shown in Figure 4. This is the structure used in the proposed test generation approach.

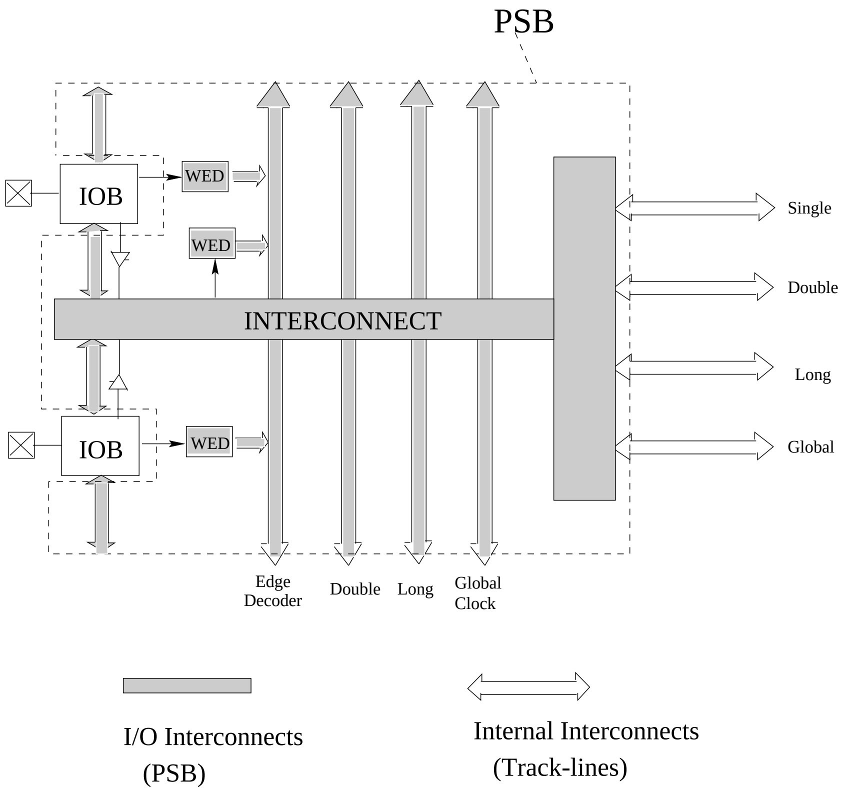
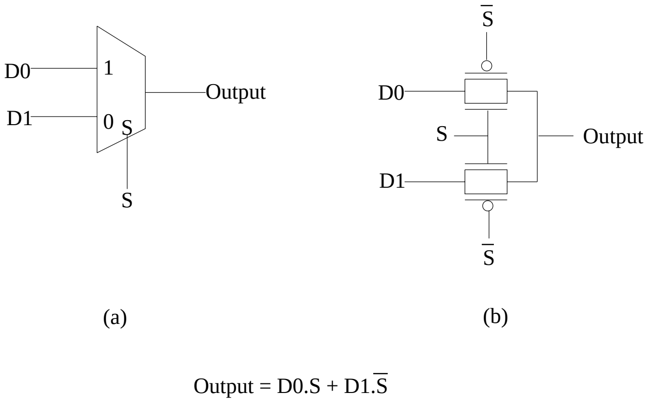


Figure 1: XC4000-Series PSB



(a)

(b)

$Output = D0.S + D1.\overline{S}$

Figure 2: 2–to–1 MUX implementation

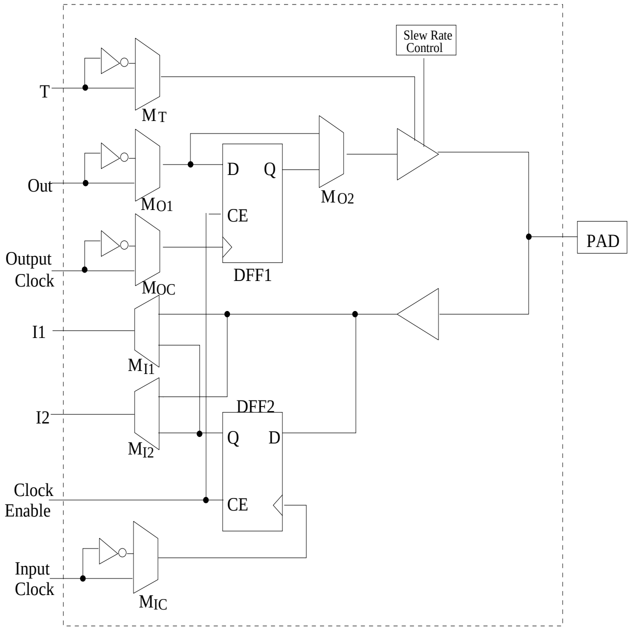


Figure 3: Simplified block diagram of the XC4000 IOB

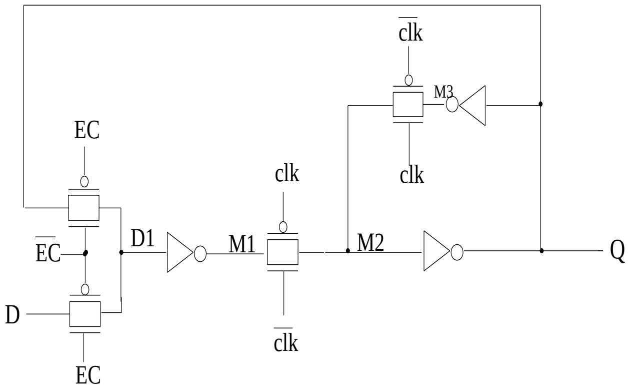


Figure 4: D flip-flop implementation

Step 1. Generate the Control Set. There are seven 2-to-1 multiplexers in each IOB, each controlled by a control net. Also, there is one extra control net for slew rate control of the output buffer. In total there are eight control nets. The control set of an IOB requires 4 phases.

Step 2. Generate the Test Set. First, the tests for intra-module faults of the configurable modules (multiplexers) are generated. The test set consists of 5 tests in 4 phases. This test set achieves 100% fault coverage for intra-module faults in the multiplexers. However, fault simulation shows that there are 4% $(\frac{13}{C_(26,2)})$ inter-module faults undetected. Then, the test set is modified and appended. The final test set is shown in Table 1 in terms of IOB inputs.

Table 1: Test Set of an IOB

Phase	Test	T	Out	PAD	CE
1	t_1	0	1	1	1
	t_2	1	0	1	1
2	t_3	1	0	1	1
3	t_4	1	1	0	1
4	t_5	0	0	0	0
	t_6	0	1	1	x

The IOBs are tested for both input and output modes by the above control and test sets. Each IOB can be defined for input, output or bi-directional signals in one phase. Multiplexer M_T controls the mode of the IOB (Figure 3). When the output of M_T is set to 0, the IOB is used as an input; when it is set to 1, the IOB is used as an output. The mode of the IOB is given in Table 2. T is the input and M_T is the control line of the multiplexer. The multiplexer implements the XOR function of T and M_T. The output of the multiplexer, IOC, controls the input/output mode. In the Mode column, I means the IOB is in input mode, O means output. In phase 2 and 3, the IOBs are configured as input and output, respectively. In phases 1 and 4, the IOBs are configured for bi-directional signals.

Table 2: Input/Output Mode of IOB

Phase	Test	T	M_T	IOC	Mode
1	t_1	0	0	0	I
	t_2	1	0	1	O
2	t_3	1	1	0	I
3	t_4	1	0	1	O
4	t_5	0	1	1	O
	t_6	1	1	0	I

6 Testing a PSB

Figure 1 shows the diagram of a PSB. First, we will discuss the test of the wide edge decoders, and then the rest of the routing resources in a PSB. The diagram of the wide edge decoders (WEDs) is given in Figure 5 [3]. $I1$ is one of the data outputs of the IOB (Figure 3). In the Xilinx 4000 series, there are two to four decoder circuits at each edge (four edge decoders – horizontal lines 1 to 4 – in Figure 5). The inputs to each decoder are any of the input signals on the edge ($I1$ from the two IOBs) plus one local interconnect per CLB row or column (C). The input signals of the edge decoders are denoted as A, $\overline{A}$, B, $\overline{B}$, C, and $\overline{C}$. Each edge decoder is assumed to behave as a wired AND.

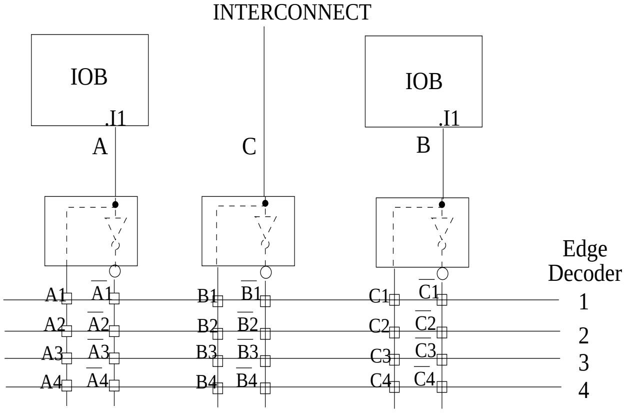

INTERCONNECT
IOB
.I1
IOB
.I1
A
C
B
Edge Decoder
A1 $\overline{A1}$ B1 $\overline{B1}$ C1 $\overline{C1}$ 1
A2 $\overline{A2}$ B2 $\overline{B2}$ C2 $\overline{C2}$ 2
A3 $\overline{A3}$ B3 $\overline{B3}$ C3 $\overline{C3}$ 3
A4 $\overline{A4}$ B4 $\overline{B4}$ C4 $\overline{C4}$ 4

Figure 5: Wide edge decoders

The control nets of the decoders are denoted as $A1$,

$\overline{A1}, A2, \overline{A2}, ..., C4, \overline{C4}$. The control nets can be tested with a modified counting sequence. However, an input signal and its complement cannot be connected to the same decoder at the same time. This means $Xi\overline{Xi}$ ($X = A, B$ or C, $i = 1, ..., 4$) can not have the control vector to be 11. For generating the control set, this is guaranteed by assigning opposite values to the control net of an input signal and its complement. Testing the connection nets is accomplished by using a modified counting sequence. Since the values of the wires are controlled by the signals A, B and C, it is possible to use the inputs of the IOBs for controlling the value of A and B. The value of C can be controlled by the content of the look-up-table in the neighboring CLBs which are connected to C. For the remaining control nets in a PSB, we assume no constraint on the control vectors. Usually, this is not allowed in normal operation. Otherwise, two driving nets with opposite values can be connected to the same net, and this will be equivalent to a short fault. However, in the testing mode, this can be avoided by connecting all connection nets to VDD or GND. Let the number of control nets in a PSB be n_c (excluding the control nets in WEDs). The number of phases required is therefore $\lceil log_2(n_c + 2)\rceil$.

The connection nets in a PSB can be classified into two types: *input nets*, which route the signals from the IOB ($I1, I2$) to the logic array (logic blocks and switch blocks); and *output nets*, which route the signal from the logic array to the IOB. Some nets are designed to serve both purposes. The stimuli to the *input nets* come from the pins, while the stimuli to the *output nets* must come from other IOBs. Since all the connection nets are controlled by I/O pins, faults between these nets can be tested in one phase. The interface to the internal interconnect is tested while testing the internal interconnect. Bridging faults in the vertical interconnect in a PSB, and between nets in the vertical and horizontal interconnects are considered here. Let the number of nets in the vertical interconnect be n_v. Faults in the vertical nets can be tested in one phase, with $\lceil log_2(n_v+2)\rceil$ vectors. Faults between vertical and horizontal nets are tested in an additional phase, i.e. by assigning opposite values to these two types of nets. Then, $\lceil log_2(n_c+2)\rceil+7$ phases and $\lceil log_2(n_v + 2)\rceil + 1$ vectors are required for testing a PSB.

7 Parallel Testing of I/O Resources

I/O resources are placed on the periphery of the chip. There are one PSB and two IOBs for each row or column. On a chip with a CLB array of size $N \times N$, there are $4N$ PSBs and $8N$ IOBs. Ideally, all these blocks can be tested simultaneously with the same configurations and test vectors. However, this is not always possible, because the I/O resources that are used for providing the stimuli, are not yet fully tested at the time. To alleviate this problem, the CLBs are used as test vector generators to provide the stimuli for the IOBs (Figure 6).

7.1 Test Strategy

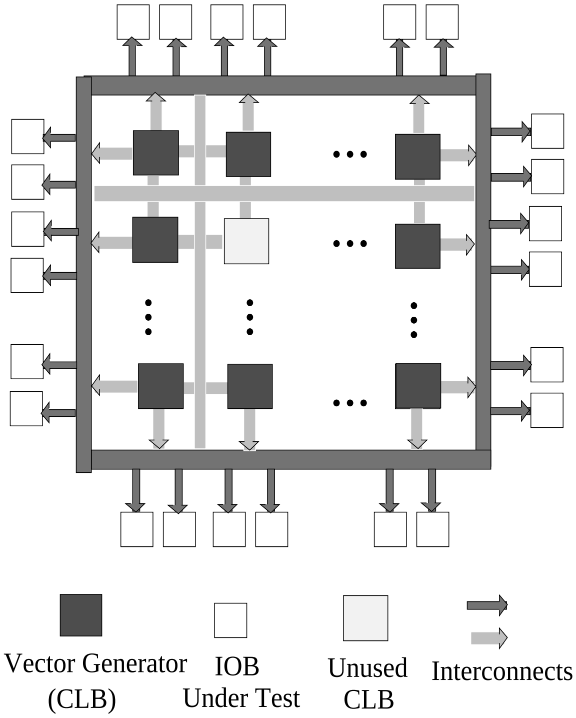


Figure 6: Configure CLBs as test generator for IOBs

The look-up tables (LUTs) in the CLBs can be configured as memory cells, and the values stored in the LUTs are used as a source of stimuli. However, since the configuration of LUTs can not be changed in one phase, they can only provide one test vector in each phase. So they can be used in those phases where only one test is needed, such as Phases 2 and 3 in Table 1. The use of the LUTs is not sufficient in phases with more than one test (e.g., Phase 1 or Phase 4 in Table 1). If there are two tests in a phase, and a sequential test vector (STV) consists of constant values ('00' or '11'), then by programming the LUTs it will suffice to provide the stimuli. For an STV given by '01' or '10', we configure the CLBs as a T flip-flop, and generate the required pattern. In this case, an additional test vector is required to set the CLBs to the proper initial state. When using the IOBs for providing the stimuli, the I/O resources are divided into two parts

(Figure 7): while part 1 is under test, part 2 is used to provide the stimuli. Then, their roles are reversed, and part 2 is tested. While testing the IOBs in part 1, the input values for the IOBs are directly controlled by the primary inputs from the pins: the stimuli for PAD come from the pin to the IOB under test, and T, Out and CE are controlled by the IOBs in part 2. The number of control and test vectors for testing all IOBs are twice those for a single IOB. For testing the interconnect, the two parts (stimuli and under-test) cannot be tested in the same phases, because the interconnect under test and the interconnect that routes the stimuli, have different configurations. They must be reconfigured when their roles are reversed.

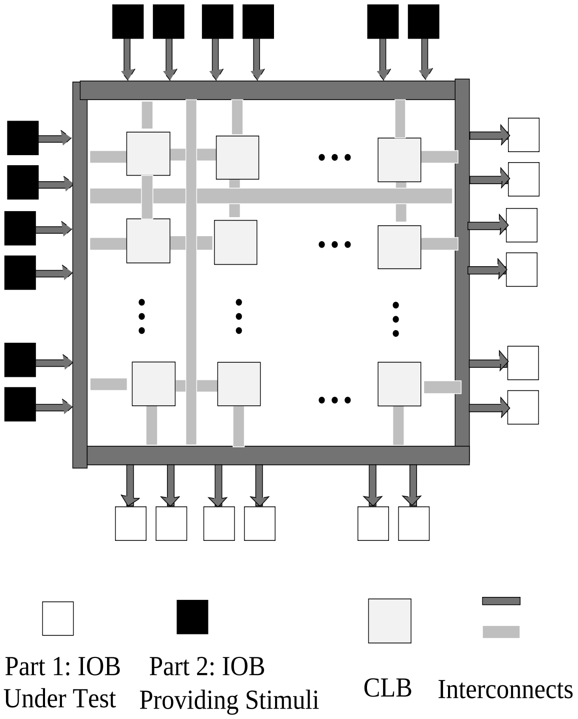


Figure 7: Partition of IOBs for parallel testing

7.2 Example: Parallel Testing of the IOBs in the XC4000

Each IOB has four inputs (T, Out, Clock Enable and PAD) excluding two clock inputs (as shown in Figure 3). The stimuli for T, Out, and Clock Enable must come from inside the chip (CLBs or the other IOBs). The stimuli for PAD comes directly from the pin. In phases 2 and 3, only one test is needed. So, the LUTs of the neighboring logic blocks are configured as memory cells to provide stimuli for the three inputs. In phases 1 or 4, there are two tests. Some of the neighboring CLBs are configured as T flip-flops to generate the changing stimuli (e.g., STVs of T, Out in phase 1, and the STVs of Out in phase 4). The STVs for the remaining signals (e.g. CE in phase 1, T and CE in phase 4) have constant values (00 or 11), and are provided by the CLBs used as memory cells. Table 3 shows the stimuli providers for each input of an IOB in each phase. LUT means that the stimuli are provided by the stored values in the LUT; TFF means that the CLBs are configured as T flip-flops to provide the stimuli.

Table 3: Test Generator (Provider) for Parallel Testing of IOBs

Phase	T	Out	PAD	CE
1	TFF	TFF	Pin	LUT
2	LUT	LUT	Pin	LUT
3	LUT	LUT	Pin	LUT
4	LUT	TFF	Pin	LUT

Testing of the I/O resources of the XC4000 series FPGA requires a total of $2\lceil log_2(n_c+2)\rceil + 18$ phases: 4 phases for testing the IOBs, and $2\lceil log_2(n_c+2)\rceil + 14$ for the PSBs. The number of vectors required is $2\lceil log_2(n_v+2)\rceil + 8$: 6 for the IOBs, and $2\lceil log_2(n_v+2)\rceil + 2$ for the PSBs (where n_v is the number of nets in the vertical interconnect and n_c is the number of control nets in a PSB excluding the WEDs).

References

[1] Lombardi, F., D. Ashen, X.-T. Chen, and W. K. Huang, "Diagnosing Programmable Interconnect Systems for FPGAs," *ACM Symposium on FPGAs,* pp. 331–339, 1996.

[2] Michinishi, H., et al., "A Test Methodology for Interconnect Structure of LUT Based FPGAs," *IEEE Asian Test Symposium,* pp. 68–74, 1996.

[3] Xilinx, Inc., *The Programmable Gate Array Data Book,* San Jose CA, 1996.

[4] Zhao, L., D. M. Walker, and F. Lombardi, "I_{DDQ} Testing of Bridging Faults in Reconfigurable Field Programmable Gate Arrays," *IEEE Transaction on Computers,* Vol. 47, No. 10, pp. 1136–1152, 1998.

[5] Zhao, L., D. M. Walker, and F. Lombardi, "Bridging Fault Detection in FPGA Interconnects Using I_{DDQ}," *ACM Symposium on FPGAs,* pp. 95–104, 1998.