

RAM-Based FPGA's: A Test Approach for the Configurable Logic

M. Renovell J.M. Portal
LIRMM-UM2 161 Rue Ada
34392 Montpellier Cedex France
renovell@lirmm.fr
Tel (33)467418523

J. Figueras
UPC Diagonal, 647
Barcelona Spain
figueras@eel.upc.es
Tel (34)34016603

Y. Zorian
Logic Vision Inc. 101 Metra Drive
San Jose CA 95110 USA
zorian@lvision.com
Tel (1)4084530146

Abstract: *This paper proposes a methodology for testing the configurable logic of RAM-based FPGAs taking into account the configurability of such flexible devices. The methodology is illustrated using the XILINX 4000 family. On this example of FPGA, we obtain only 8 basic Test Configurations to fully test the whole matrix of CLBs. In the proposed Test Configurations, all the CLBs have exactly the same configuration forming a set of one-dimensional iterative arrays. The iterative arrays present a C-testability property in such a way that the number of Test Configurations 8 is fixed and independent of the FPGA size.*

1. Introduction

Field Programmable Gate Arrays (FPGAs) combine the flexibility of mask programmable gate arrays with the convenience of field programmability [1-2]. There are many FPGA types of which the RAM based FPGA architecture is widely used and has a growing share of the FPGA market. In such a programmable circuit, a matrix of logic modules and interconnection elements can be configured in the field to implement a desired function. Considering the specificities of such reconfigurable circuits, the authors have proposed a methodology for FPGA's testing [3,4]. This methodology distinguishes two types of testing procedures: the Manufacturing Test Procedure (MTP) that must cover all possible mode of operations (configurations) and the User Test Procedure (UTP) that must cover only the user's configuration.

A User Test Procedure may be needed for example, to apply an incoming test to standart FPGAs. A Manufacturing Test Procedure is needed obviously after manufacturing but two different situations must be considered. In the first situation, the FPGA represents the whole chip and the circuit is manufactured as a standart chip for a FPGA manufacturer. In this situation, the FPGA manufacturer is in charge of defining the Manufacturing Test Procedure. More recently, a second situation appears where the FPGA is only a part of the chip and the circuit is manufactured as an ASIC designed with different cores. The availability of FPGAs as cores is a new situation that implies to perfectly dominate the FPGA test problems.

Considering the novelty of the situation, the authors have decided to focus on the problem of defining a Manufacturing Test Procedure for RAM-based FPGAs.

Testing of these chips has only recently been addressed [3-17]. In the published works, different aspects of FPGA testing have been addressed:

- Inoue and al. address the problem of testing look-up table in [6],
- Huang and al. address the problem of testing the configurable logic in [7].
- Abramovici and al. focus on BIST for FPGA in [8,9,10],
- Lombardi and al. focus on diagnosis in [11,12],

For all these authors, it seems clear that considering the FPGA as a classical digital ASIC is not a realistic test approach. As the matter of fact, testing the FPGA presents as a good recollection of all the problems encountered in testing. Consequently, due to the complexity of the whole circuit, a classical divide and conquer strategy is adopted. The FPGA is conceptually divided into 3 subparts according to the function of the different architectural elements:

- The Interconnect Structure,
- The array of Logic Modules,
- The static RAM for the configuration control bits.

Due to the strong differences of these 3 subparts, it is obvious that specific test approaches are required: an interconnection oriented approach for the Interconnect Structure, a logic oriented approach for the array of logic modules and a memory oriented approach for the static RAM. Following this approach, the authors have proposed a Manufacturing Test Procedure targetting the Interconnect Structure of RAM-based FPGAs in [3,4]. The work presently under development concerns the definition of a Manufacturing Test Procedure targetting the configurable logic of RAM-based FPGAs [16,17]. The test of the static RAM part will be considered in the future.

Taking into account the configurability of the FPGA, a Manufacturing Test Procedure consists in successively configuring the FPGA using the Configuration input then applying a test sequence using the Operation inputs. Consequently, we have to define a set of configurations only devoted to the test which are application independent. These test-oriented configurations are simply called « Test Configurations » and denoted TC^i . And so, a manufacturing test procedure MTP for FPGA is represented as a sequence of pairs of a Test Configuration TC^i and its associated Test Sequence TS^i [6]:

$$MTP = \{(TC^1, TS^1), (TC^2, TS^2), \dots, (TC^{n_{TC}}, TS^{n_{TC}})\}$$
with n_{TC} number of Test Configurations

Considering this definition, it must be clear that only two parameters can be optimized: the number of Test Configurations (n_{TC}) and the number of test vectors applied in each configuration. It is of prime importance to note that a given FPGA configuration TC^i corresponds to a fixed sequence of bits serially entered in the FPGA. Consequently, the FPGA configuration process is an excessively time-consuming process, hence the number of FPGA re-configurations (n_{TC}) must absolutely be minimized.

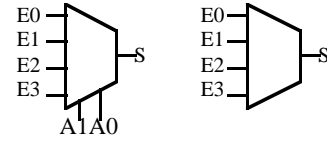
The objective of this paper is to define a Manufacturing Test Procedure targetting the Configurable Logic of a RAM-Based FPGA with the objective of minimizing the number n_{TC} of device re-configuration. In the definition of Test Configurations and Test Sequences, this paper uses a bootom-up approach starting from the elementary devices such as multiplexers and look-up tables and going till the complete matrix of Configurable Logic Modules. The XC4000 family of Xilinx is used to illustrate the proposed approach [18].

Section 2 summarizes the preliminary results proposed in [16,17] concerning the fault model and the test of the elementary devices such as multiplexers and look-up tables. Using these results, Section 3 gives the Test Configurations and Test Sequences for a single Logic Module. It is demonstrated that only 8 basic Test Configurations can be used to completely test a single CLB of the XILINX 4000 family. In section 4, the test of the whole $m*m$ matrix of logic modules is considered. It is demonstrated that the 8 basic test configurations can be cascaded preserving the property of controllability and observability. This implies that only 8 Test Configurations are required to fully test the $m*m$ matrix of logic modules. Section 5 discuss the problem of controllability and observability. Finally, section 6 gives some perspectives and concluding remarks.

2. Testing the Elementary Devices

This section is devoted to the test of the elementary devices. This problem has been discussed in detail by the authors in [16,17]. This section summarizes the main results. The Logic Modules usually consists of three types of elementary devices: D flip-flops, multiplexers and look up table units (LUT). The multiplexers and the look-up tables are typical configurable devices while the D flip-flop are not really configurables. In fact, the control signals of the flip-flop (reset, clock...) are configurables by means of multiplexers. Because we concentrate here on configurable devices, only multiplexers and look-up tables are considered in this section.

In figure 1, we have an example of classical 4-to-1 multiplexer with of 2 bit address $A0,A1$. In a typical FPGA representation, the data inputs $E0,E1,E2,E3$ are represented because they are Operation Inputs while the 2 bit address are not represented because they are Configuration Inputs not available during normal operation. The fault model associated to this device is the stuck-at of all the internal and external (I/O: $E0,E1,E2,E3,A0,A1,S$).



a) Classical b) FPGA
Figure 1: Mux Representations

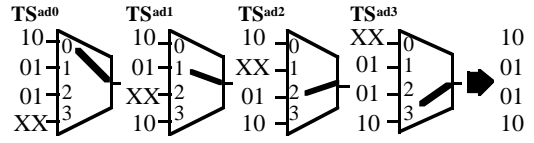


Figure 2: TC's and TS's for the Mux

It is easy to demonstrate that the assumed stuck-at fault implies that an exhaustive sequence must be applied on the address inputs. Knowing that the address inputs are Configuration Inputs, this is equivalent to define 2^n different Test Configurations for a n bit address multiplexer. The 4 Test Configurations of the 4-to-1 multiplexer of figure 1 are illustrated in figure 2 where the configurations are symbolically represented by a connection between an input and output. The 4 Test Configurations are called $TC^{ad0}, TC^{ad1}, TC^{ad2}, TC^{ad3}$. After the Test Configuration, we define the Test Sequences associated to each Test Configuration. Considering the stuck-at fault, the Test Sequences to be applied corresponds to the 2 Test Vectors illustrated in figure 2. It is interesting to note that it could be possible to define a common sequence of 2 vectors for the 4 Test Configurations. This common sequence corresponds to the exclusive-OR and complemented exclusive-OR vectors.

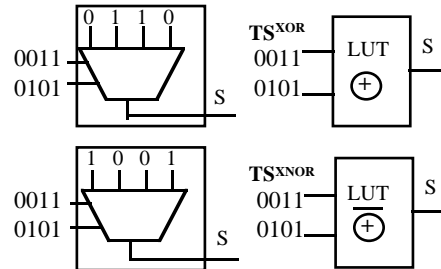


Figure 3: TC's and TS's for the Look-up Table

Concerning now the test of the LUT illustrated in figure 3, we consider a single stuck-at fault at each cell of the LUT as well as in any logical node of the address decoder. It can be easily verified that the test vectors are identical to those covering the 100% stuck-at of a multiplexer with as many inputs as the number of LUT entries. Hence, we can use the vectors previously defined for the mux. The exclusive-OR and complemented exclusive-OR vectors are applied on the LUT Configuration Inputs (Mux data inputs) and an exhaustive sequence of 2^n vectors is applied on the Operation Inputs (Mux address inputs). This is equivalent in defining 2 Test Configurations called TC^{XOR} and TC^{XNOR} and defining 2 corresponding Test Sequences called TS^{XOR} and TS^{XNOR} . The 2 Test Configurations are

symbolically represented by a XOR (\oplus) or XNOR symbol inside the LUT.

It is interesting to note that the number of Test Configurations for a LUT is two independently of the LUT size. This result favours the minimization of the number of Test Configurations as mentioned in the introduction. On the contrary the number of vectors in the Test Sequences is equal to 2^n and obviously depends on the LUT size.

3. Testing a single CLB

In this section, we define Test Configurations and Test Sequences not for a single elementary device but for the FPGA logic module obtained by interconnecting elementary devices. In order to illustrate the definition of Test Configurations for complex modules, let us consider first an example of extremely simple module composed of an interconnection of only 1 two-input LUT and 1 two-input multiplexer. The simple module shown in figure 4 has 3 Operation Inputs, 5 Configuration Inputs (4 for the LUT and 1 for the multiplexer) and 1 Operation Output. The exhaustive number of Configurations is $2^5=32$, but we want to select a minimum of them to test the simple module.

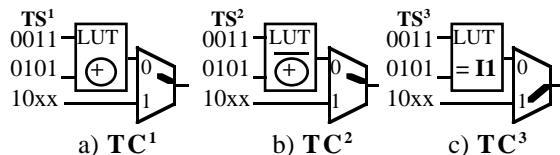


Figure 4: TC's and TS's for the Simple Module

The first optimization consists in using the results of section 2 implying to configure the LUT with the TC^{XOR} and TC^{XNOR} configurations and the multiplexer with the TC^{ad0} and TC^{ad1} configurations. These 4 Test Configurations can even more be optimized to the 3 Test Configurations of figure 4. Indeed, some Test Configurations of the elementary devices are compatible and can be simultaneously used. As an example, the LUT configurations TC^{XOR} or TC^{XNOR} are compatibles with the mux configurations TC^{ad0} . They are simultaneously used in TC^1 and TC^2 .

Note that the configuration of the LUT is indifferent in TC^3 . In this case, the LUT is not configured with a Test Configuration but with a functional configuration that makes the application of the Test Sequences TS^3 easier. In this case, the logic function implemented in the LUT makes the output of the LUT equal to the input I1 symbolically represented by '= I1' inside the LUT.

The Test Sequences TS^1 , TS^2 and TS^3 associated to each Test Configuration are obtained from the Test Sequences of each elementary devices. In fact, the Test Sequences of the elementary devices are simply justified through the other elementary devices. As an example in Figure 4.c the Test Sequences TS^3 simply corresponds to the Test Sequences of the mux justified through the equality function of the LUT.

This optimization process is applied now to a complex CLB of the XILINX 4000 family illustrated in figure 5.a. For the sake of clarity, the CLB is divided into two parts:

the combinational part in figure 5.b and the sequential part in figure 5.c [7]. The names used for the different elements correspond in general to the name of the controlled node: for example LUT G' control node G' and MUX SR control node SR... The global multiplexer (H1,DIN,SR,EC) is represented as 4 independent multiplexers: MUX H1, MUX DIN, MUX SR and MUX EC. The SR controller is represented as a demultiplexer whose test is equivalent to the test of a multiplexer. Of course, the combinatorial and sequential parts are not completely independent. It can be noted in figure 5.c that multiplexer DY and DX have G',H' and F' as inputs. The CLB presents 12 inputs C1-C4,G1-G4,F1-F4 and 4 outputs X,XQ,Y,YQ.

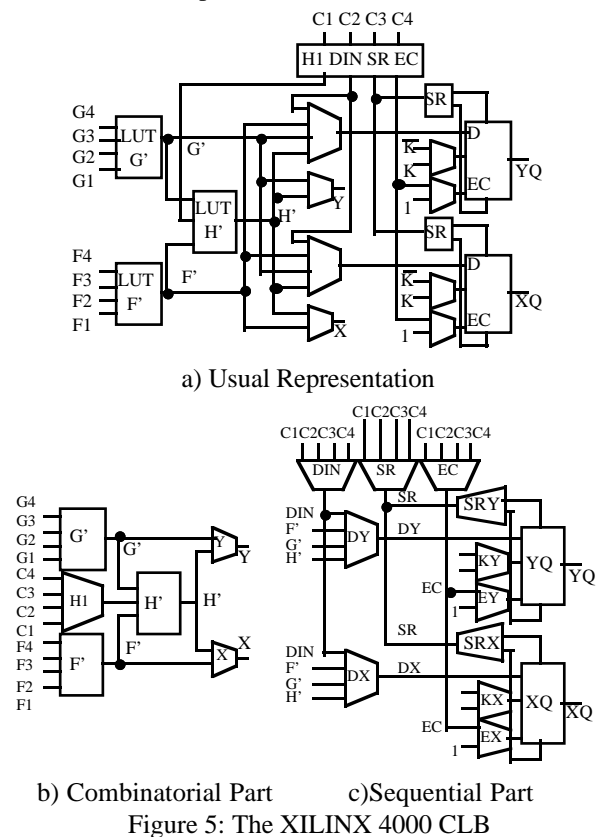


Figure 5: The XILINX 4000 CLB

The minimization of the number of Test Configuration using the elementary device Test Configurations as described in the first part of this section, leads to only 8 Test Configurations for completely testing the complex XILINX 4000 CLB. Among these basic 8 Test Configurations, 4 are mainly dedicated to the test of the combinational part of the CLB and so are called: TC^{Com1} , TC^{Com2} , TC^{Com3} and TC^{Com4} . In the same way, the 4 remaining are fully dedicated to the test of the sequential part of the CLB and so are called: TC^{Seq1} , TC^{Seq2} , TC^{Seq3} and TC^{Seq4} . Our basic 8 Test Configurations are illustrated in Figure 6 (end of the paper). In Figure 6, the configuration of each elementary device is symbolically represented as defined in the previous section. When the symbol is not present, the configuration of the device has no impact. The corresponding Test Sequences TS^{Com1} , TS^{Com2} , TS^{Com3} , TS^{Com4} , TS^{Seq1} , TS^{Seq2} , TS^{Seq3} and TC^{Seq4} are not given in the figure for reason of space

limitation but also because the definition of these Test Sequences is straightforward using a justification procedure.

Note that the basic 8 Test Configurations for the complex CLB cover the Test Configurations of any elementary device. As an example, the 3 LUT appear in the TC^{XOR} and TC^{XNOR} configurations. As another example, multiplexer DIN appears in configuration TC^{ad0} in TC^{com3} , in configuration TC^{ad1} in TC^{com1} , in configuration TC^{ad2} in TC^{com2} and in configuration TC^{ad3} in TC^{com4} .

4. Testing a Matrix of CLBs

In this section, the objective is to define Test Configurations and Test Sequences for the $m \times m$ matrix of CLBs of a RAM-based FPGA. In case of a matrix of CLBs, the problem consists in controlling and observing the whole matrix. Individual access to each CLB is not possible in practice. Indeed, a FPGA does not have enough I/O pads to control and observe each CLB in parallel from outside. As an example, the XILINX 4013 has 576 CLBs (with 12 inputs and 4 outputs each) and only 192 I/O pads

For these reason, the CLBs are interconnected in a special way forming one-dimensional arrays of cascaded CLBs. The length of the one-dimensional array is not important. The number and length of the arrays only depends on the number of available I/O pads. In practice, the most convenient solution is illustrated in Figure 7 where a $m \times m$ (3×3) matrix of CLB is distributed in m (3) one-dimensional arrays of m (3) CLBs. Using this scheme, the m one-dimensional arrays are tested in parallel.

In fact, a one dimensional array of m cascaded CLBs can be viewed as an iterative circuit. Each CLB received a local input (white or grey arrows) from the previous CLB and produces a local output (white or grey arrows) for the next CLB. The left most local inputs (grey arrows) are controllable primary inputs and the right most local outputs (grey arrows) are observable primary outputs. In addition, each CLB receives a number of controllable primary inputs (black arrows) that are common to every CLB in the FPGA.

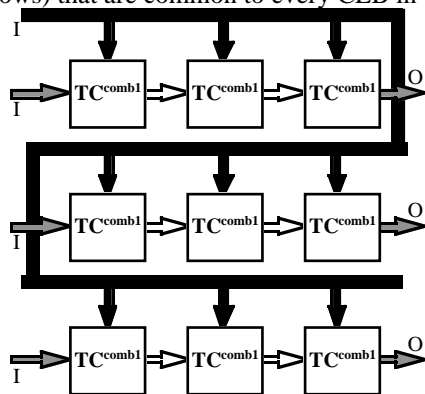


Figure 7: 3 One-Dimensional arrays of 3 CLBs

Using the concept of one-dimensional array, it is clear in Figure 7 that an ‘embedded’ CLB is controlled through CLBs located on its left side and observed through CLBs located on its right side. Consequently, the Test

Configurations and Sequences proposed for the whole matrix of CLBs must meet 3 fundamentals requirements:

- i) guarantee the complete test of each CLB,
- ii) guarantee the propagation through a given CLB of signal controlling other CLBs on its right side,
- iii) guarantee the propagation through a CLB of observing signals from other CLBs on its left side.

The complete test of each CLB i.e. the first requirement (i) can be guaranteed by using the basic 8 Test Configurations and Test Sequences proposed in the previous section. In our approach, all the m^2 CLBs in the FPGA are configured exactly in the same way. As an example in Figure 7, the Test Configuration TC^{comb1} is used for all the m^2 CLBs of the circuit. The Test Configuration TC^{comb1} means now that all the m^2 CLBs are configured with the TC^{comb1} configuration. As a result, we simply define 8 Test Configurations for the whole matrix corresponding to the 8 basic Test Configurations of the single CLB. The 8 Test Configurations with the associates Test Sequences guarantee the complete test of each CLB. The problem consists now in demonstrating that the proposed interconnecting principle (one-dimensional array) guarantees the controllability and observability of any embeded CLB i.e. the second and third requirements (ii and iii). These demonstrations are given in the following section.

As a result, the Manufacturing Test Procedure we propose for the logic modules of a RAM-based FPGA consists in using 8 basic Test Configurations with the associated Test Sequences. In each Test Configuration, the circuit is composed of m parallel and independent iterative arrays, each composed of m interconnected CLBs. At this point, it must be noted that the complete Test Procedure has been simulated using an iterative array of 4 CLBs giving 100% coverage of the assumed fault models. These simulations validate the proposed Test Configurations and Test Sequences. Note that Huang and Lombardi in [7] propose 21 Test Configurations for the same circuit using a functional fault model for LUTs and mux. However, the assumption of a functional fault model implies more vectors and some of them may be unnecessary to cover structural faults.

5. Observability and controllability of the Test Configurations

The 8 Test Configurations use the same method and so only the first one is detailed. In the first combinatorial Test Configuration, the outputs X and Y of the CLBs are connected to the inputs F4 and G4 of the next CLB as illustrated in figure 8.a. These connections form the local connections of the iterative array while the other inputs (F1,F2,F3,G1,G2,G3,C1) are the controllable primary inputs common to all the CLBs. Of course, the left most F4 and G4 inputs called G^{init} and F^{init} are also controllable primary inputs while the right most X and Y outputs are the observable primary outputs.

In such conditions, the iterative array presents a cascade of XOR function implemented in the LUT as illustrated in figure 8. The XOR configuration has been defined in section 2 to test the LUT but the corresponding XOR function of the LUT presents a very interesting property of observability and controllability. It is obvious that any faulty CLB is observable through the XOR function of the following CLBs. The XOR function guarantees the test of the LUT as well as the observability of any faulty 'embedded' CLB.

In the same way, each CLB is easily controllable by means of the primary inputs (F1,F2,F3,G1,G2,G3 and C1) and through the XOR functions of the previous CLB. Note that connections Y1 and X1 are not crossed while all the remainder Yi and Xi in the array are crossed to guarantee a periodicity. Indeed, assuming some properties of the input test sequence, the output of each CLB (local nodes) produces exactly the same vectors:

Property :

$$\text{For any vector } (C1,F1,F2,F3,Fini,G1,G2,G3,Gini) / \\ G^{ini}=F^{ini}=C1\oplus F1\oplus F2\oplus F3\oplus G1\oplus G2\oplus G3=Cnt \text{ (0 or 1)}$$

In CLB1:

$$Y1=G'=G1\oplus G2\oplus G3\oplus G^{ini} \Rightarrow Y1= G1\oplus G2\oplus G3\oplus Cte \\ F'= F1\oplus F2\oplus F3\oplus F^{ini} \\ X1=H'= G'\oplus C1\oplus F' \quad \Rightarrow X1 = Cnt$$

In CLB2:

$$Y2=G'=G1\oplus G2\oplus G3\oplus Y1 \Rightarrow Y2 = Cnt... \\ F'= F1\oplus F2\oplus F3\oplus X1 \\ X2=H'= G'\oplus C1\oplus F' \quad \Rightarrow X2 = G1\oplus G2\oplus G3\oplus Cte$$

In CLB3:

$$Y3=G'=G1\oplus G2\oplus G3\oplus X2 \Rightarrow Y3 = Cnt \\ F'= F1\oplus F2\oplus F3\oplus Y2 \\ X3=H'= G'\oplus C1\oplus F' \quad \Rightarrow X3 = G1\oplus G2\oplus G3\oplus Cte \\ \Rightarrow Y2=Y3=Y4=... \quad \text{and} \quad X2=X3=X4=....$$

The periodicity of the local output guarantees the controllability of the embedded CLBs but also a very easy sequence generation because each CLB receives the same Test Sequence. As a result a Test Sequence TS^{com1} of only 16 vectors is required for the complete iterative array of m CLBs.

Considering now the sequential part of the CLBs in the first combinatorial Test Configuration TC^{com1} . As illustrated in figure 8.b, multiplexer DIN and DY connect the input C2 to the latch YQ while multiplexer KY and EY allow to correctly clock the latch. Consequently, in the first combinatorial Test Configuration the output YQ of the CLBs are connected to the input C2 of the next CLB. These connections form the local connections of a second iterative array. Of course, the left most C2 input called C^{init} is a controllable primary input while the right most YQ output is the observable primary output. By connecting the output YQ to the next input C2 i.e. the input of latch YQ, the iterative array corresponds to a m-bit shift register.

Obviously, this shift register configuration allows a very easy Test Sequence generation because any element in the shift register is controllable and observable. As a result for a shift register of m CLBs, a Test Sequence of only 3m-1 clock cycles is required.

We finally observe that a one-dimensional array of cascaded CLBs in the first Test Configuration TC^{com1} corresponds to a cascade of XOR functions (combinatorial part in figure 8.a) in parallel with a shift register (sequential part in figure 8.b). The 4 combinatorial Test Configurations TC^{com1} , TC^{com2} , TC^{com3} , TC^{com4} use exactly the same principle.

The 4 sequential Test Configurations TC^{seq1} , TC^{seq2} , TC^{seq3} , TC^{seq4} use a similar principle since they implement two shift registers in parallel. Indeed, in the sequential Test Configurations, the outputs XQ and YQ of the CLB are connected to the inputs F4 and G4 of the following CLB in order to form two parallel shift registers. The input of the top latch YQ is controlled by G4 through LUT G' and MUX DY. In the same way, the input of the bottom latch XQ is controlled by F4 through LUT F' and MUX DX. In order to have a shift register with no data modification, the LUT are configured with the identity function (identity to input 4 as defined in section 3). Simultaneously, the input C4 controls the 'enable' functions through MUX EC, EY and EX while input C3 controls the 'set' functions through MUX SR, SRY and SRX.

6. Conclusion and Discussion

In this paper, we propose the use of 8 Test Configurations to fully test all the CLBs of the XILINX 4000 family for a single Stuck-at Fault model. In our approach, all the CLBs have exactly the same configuration forming one-dimensional iterative arrays. The 4 combinatorial Test Configurations implement a cascaded XOR or XNOR function in parallel with a shift register. The 4 sequential Test Configurations implement two parallel shift registers. A fault simulation performed with the proposed 8 Test Configurations and Test Sequences have demonstrated that 100% of fault coverage is obtained for the considered structural stuck-at fault model. These Test Configurations have been implemented on a XILINX 4000 and its viability and effectiveness verified. Figure 9 gives an example of sequential Test Configuration implemented on XILINX.

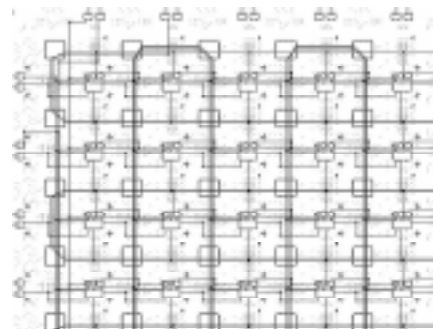


Figure 9: Implementation of Tc^{seq1} on XILINX

The concept of iterative array is used to ensure the controllability and observability of any embedded CLB. It is clear that the iterative array presents the property of C-Testability in such a way that the number of Test Configurations is fixed and independent of the size $m*m$ of the matrix.

7. References

[1] S.D. Brown, R.J. Francis, J. Rose, S.G. Vranesic, « Field-Programmable Gate Arrays », Kluwer Academic Publishers, 1992.
 [2] S.M. Trimberger (ed), « Field-Programmable Gate Array Technology », Kluwer Academic Publishers, 1994.
 [3] M. Renovell, J. Figueras and Y. Zorian, «Test of RAM-Based FPGA: Methodology and Application to the Interconnect» 15th IEEE VLSI Test Symposium, pp. 230-237, Monterey, CA, USA, May 1997.
 [4] M. Renovell, J. Figueras and Y. Zorian, «Testing the Interconnect Structure of Unconfigured FPGA», IEEE European Test Workshop, pp. 125-129, Sète (Montpellier), FRANCE, June 1996.
 [5] C. Jordan and W.P. Marnane, « Incoming Inspection of FPGAs » Proc. of IEEE European Test Conference, pp. 371-377, 1993.
 [6] T. Inoue, H. Fujiwara, H. Michinishi, T. Yokohira and T. Okamoto, « Universal Test Complexity of Field-Programmable Gate Arrays » 4th Asian Test Symposium, pp. 259-265, Bangalore, November 1995, India.
 [7] W.K. Huang and F. Lombardi, « An Approach for Testing Programmable/Configurable Field Programmable Gate Arrays » 14th IEEE VLSI Test Symposium, pp. 450-455, Princeton, NJ, USA, May 1996.
 [8] M. Abramovici and C. Stroud, «No-Overhead BIST for FPGAs », 1st IEEE International On-line Testing Workshop, pp. 90-92, Nice, FRANCE, 1995.

[9] C. Stroud, P. Chen, S. Konala, M. Abramovici, « Evaluation of FPGA Ressources for Built-In Self Test of Programmable Logic Blocks », Proc. of 4th ACM/SIGDA Int. Symposium on FPGAs, pp. 107-113, 1996.
 [10] M. Abramovici, C. Stroud, «ILA BIST for FPGAs: A Free Lunch with Gourmet Food» 2nd IEEE International On-line Testing Workshop, pp. 91-95, Biarritz, FRANCE, 1996.
 [11] F. Lombardi, D. Ashen, X.T. Chen, W.K. Huang « Diagnosing Programmable Interconnect Systems for FPGAs » FPGA '96, pp. 100-106, Monterey CA, USA, 1996.
 [12] W.K. Huang, X.T. Chen and F. Lombardi, « On the Diagnosis of Programmable Interconnect Systems: Theory and Application» 14th IEEE VLSI Test Symposium, pp. 204-209, Princeton, NJ, USA, May 1996.
 [13] M. Hermann and W. Hoffmann, « Fault modeling and test generation for FPGAs » in R.W. Hartenstein and M.Z. Servit (eds), Lecture Notes in Computer Science, Field Programmable Logic, Springer-Verlag, pp. 1-10, 1994.
 [14] R.O. Durate and M. Nicolaidis, « A test methodology applied to cellular logic programmable gate arrays, » in R.W. Hartenstein and M.Z. Servit (eds), Lecture Notes in Computer Science, Field Programmable Logic, Springer-Verlag, pp. 11-22, 1994.
 [15] T. Liu, W.K. Huang, F. Lombardi, «Testing of Uncustomized Segmented Channel FPGAs» Proc. of ACM Int. Symp. on FPGAs, pp. 125-131, 1995.
 [16] M. Renovell, J.M. Portal, J. Figueras and Y. Zorian, «Testing Unconfigured FPGA Logic modules», IEEE European Test Workshop, Cagliari (Sardigna), Italy, May 1997.
 [17] M. Renovell, J.M. Portal, J. Figueras and Y. Zorian, «Test Pattern and Test Configuration Generation Methodology for the Logic of RAM-Based FPGA», IEEE Asian Test Symposium, pp. 254-259, Akita, Japan, November, 1997.
 [18] Xilinx, «The Programmable Logic Data Book », San Jose, USA, 1994.

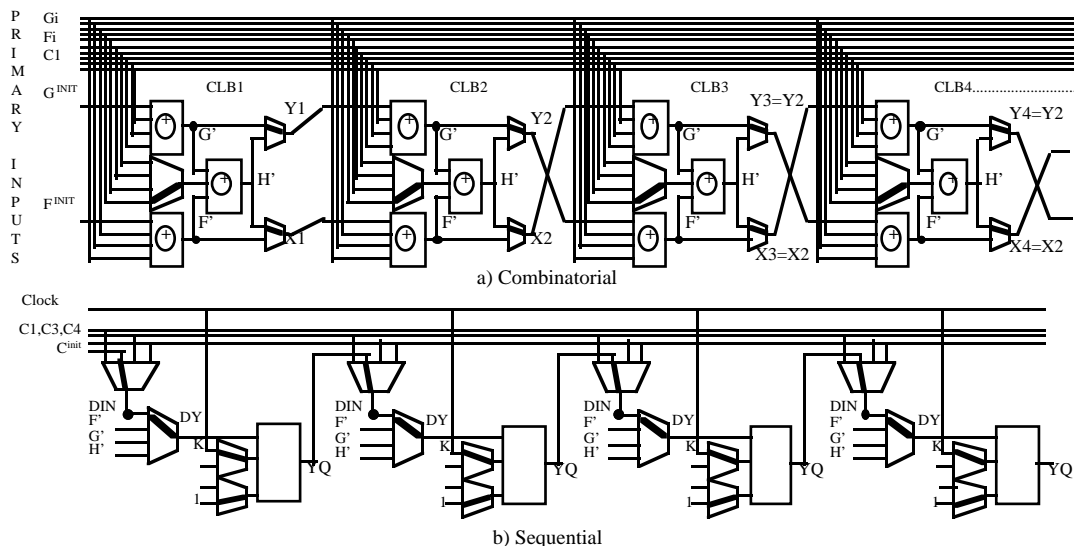


Figure 8: The Combinatorial and Sequential Arrays

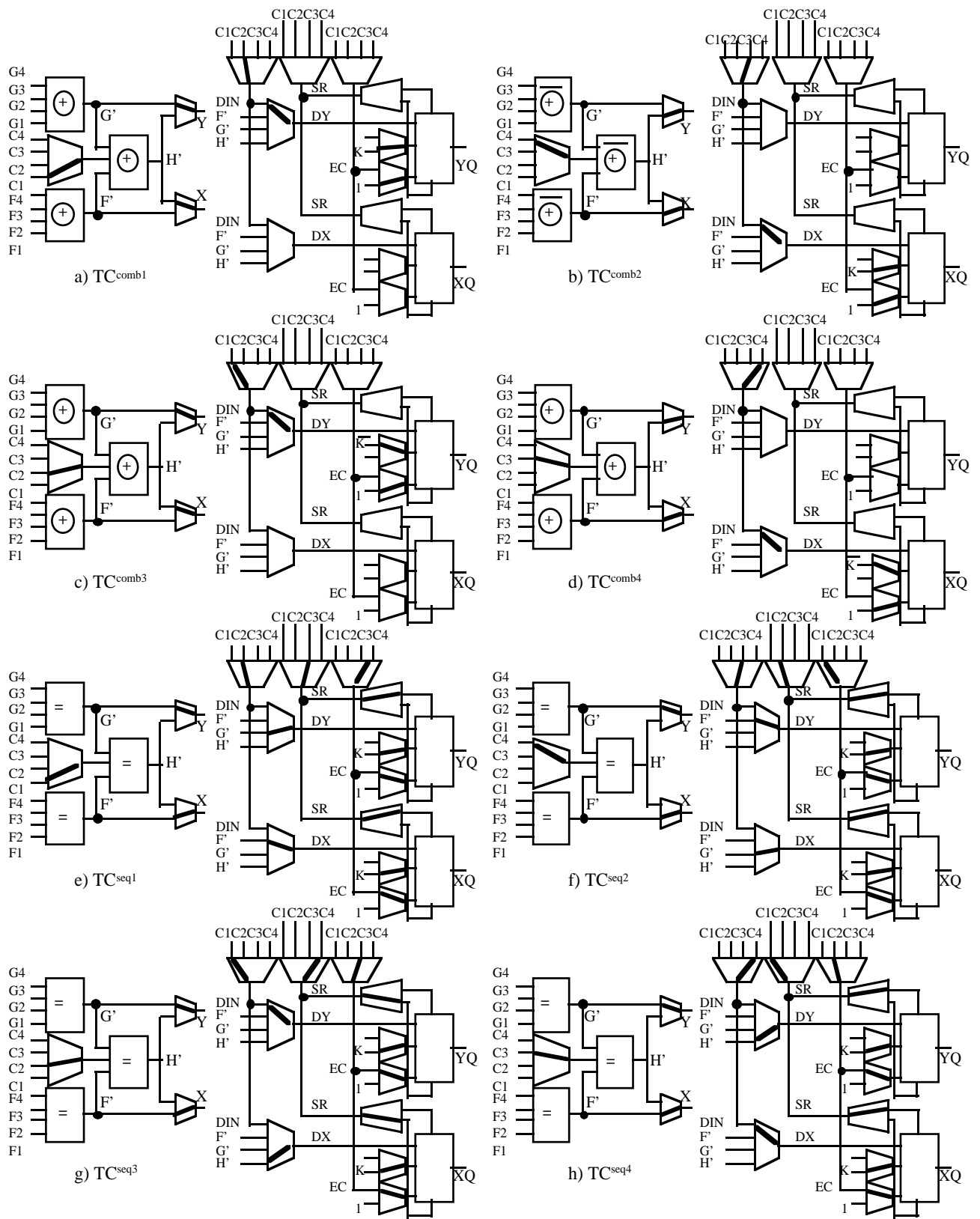


Figure 6: The basic 8 Test Configurations