ROMANIAN JOURNAL OF INFORMATION SCIENCE AND TECHNOLOGY
Volume **14**, Number 4, 2011, 392–398

Testability Optimizations for A Time Multiplexed CPLD Implemented on Structured ASIC Technology

Traian TULBURE[1], Radu GĂVRUŞ[2]

[1]Dept. of Electronic and Computers,
Transilvania University of Braşov, Romania
E-mail: tulbure@vega.unitbv.ro

[2]eASIC Corporation Braşov, Romania

Abstract. Multi-context dynamic reconfigurable architectures can use of both spatial and temporal aspects of logic capacity. Gaining logic capacity by reusing hardware with time multiplexing techniques requires controlling logic using small distributed memories. This can create testability problems for dynamic reconfigurable architectures. In this paper we are analysing the test coverage loss and proposing a test extension for a CPLD like dynamic reconfigurable architecture. The proposed solution is implemented in structured ASIC technology and helps in regaining the test coverage with no area and timing penalty.

Keywords: reconfigurable computing, CPLD, structured ASIC, test coverage, ATPG.

1. Introduction

As the size and complexity of integrated circuits increases according to Moore law, testing of a chip becomes more and more challenging. For multi-million gate designs or systems-on-chip (SoC), functional vectors do not reveal all possible faults and complete testing is virtually impossible. Synthesis tools provide full support for design for test (DFT) insertion, allowing post manufacturing testing using automatic test pattern generation (ATPG) technique. The test coverage using this methodology should be as high as possible to minimize the chance that a chip validated using ATPG proves to be not functional due to defects in fabrication.

Structured ASIC is a technology between standard ASIC and FPGA that combines the benefits of both. It has standard cell ASIC-like unit cost, density and power consumption. It features low up-front development cost and simple, FPGA-like design flow. The power consumption can be improved further by other techniques [2]. Selected structured ASIC, eASIC, [7] has look-up table based logic cells while routing is fixed using single via metallization layer. The basic cell is designed for DFT compliance, including D flip-flop modified to support internal scan by the addition of a multiplexer (this scan style is called multiplexed flip-flop). The presence of already inserted DFT reduces design and test time considerably for projects implemented on structured ASIC.

Dynamic reconfigurable architectures can be embedded in projects implemented on structured ASIC technology, to regain the flexibility of post-production reprogramming or to allow usage in the area of reconfigurable computing. The idea of multi-context dynamic reconfigurable (also called time multiplexed) architectures is not new. Several different solutions have been proposed, such as the Dynamically Programmable Gate Array, Time Multiplexed FPGA[4] and the Time Switched FPGA [1] but only recently the idea is productized by Tabula in their 3PLD programmable devices [3].

The dynamic reconfigurable CPLD architectures for structured ASIC have been introduced in [4]. Using on-chip memory it has benefits in allowing very fast circuit reconfiguration but also has the drawback of reduced testability due to possible combinatorial loops. The same problem of combinatorial loops was described in [6] for product term based synthesizable embedded programmable logic cores. Their solution is either to used dual interconnect network or a decoupled interconnect architecture. This requires a directional network for combinatorial logic interconnect and another routing network for sequential elements. The solution works for any fabrication technology but the drawback is increased area utilization and special requirements for mapping tools.

This paper proposes a test optimized dynamic reconfigurable CPLD architecture particularized for structured ASIC. It describes changes in CPLD architecture that boost ATPG test coverage above 98% with no significant degradation in area or timing.

2. eCPLD Architecture

The proposed dynamic reconfigurable CPLD architecture, called eCPLD, is built from multiple dynamic-reconfigurable AND-OR arrays (ePLD) and one dynamic reconfigurable interconnect (eConnect) as presented in Fig. 1a.

The basic logic block, called ePLD, is built from multiple configurable product terms as presented in Fig. 1b. The programming of the desired logic function is done using memory bits that control the dynamic reconfigurable nodes, called eNodes. Assuming that we want to implement multiple functions at different time moments, when using the ePLD structure, we must control all eNodes from small size distributed memories. Those memory instances need to be placed close to the corresponding node

and need to be able to switch simultaneously to change the active ePLD context. Controlling logic with values stored in on-chip memory can cause problems during ATPG testing because invalid programming can create combinatorial loops during testing.

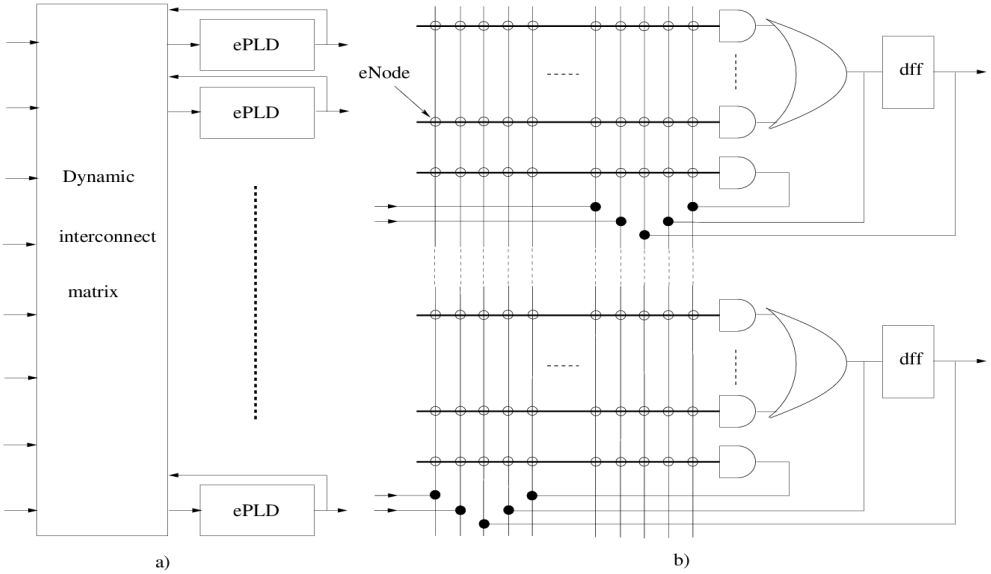

Fig. 1. a) eCPLD architecture; b) ePLD internal structure.

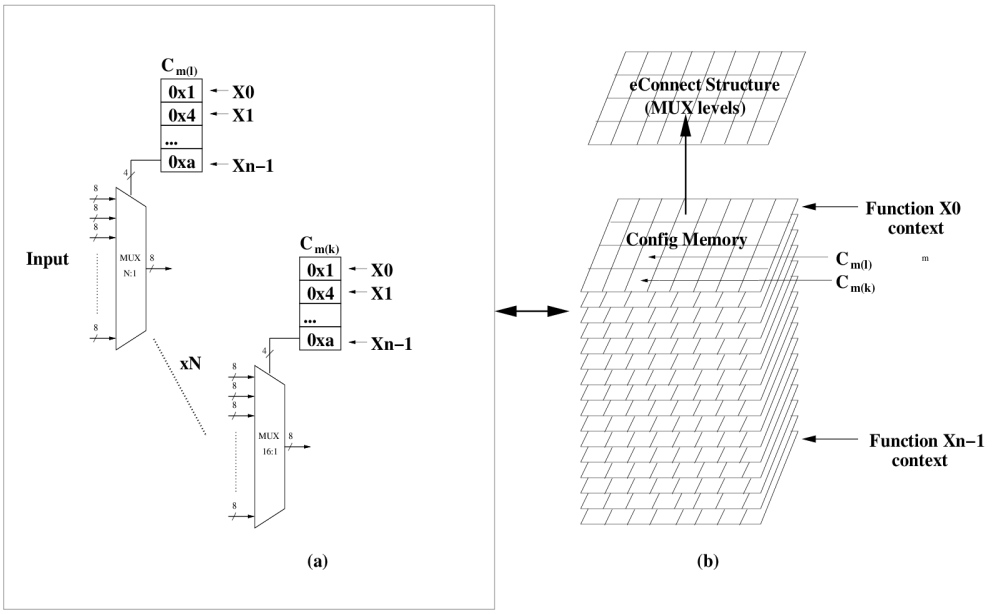

Fig. 2. eConnect: a) Multiplexer based interconnect; b) Configuration memory planes.

The dynamic reconfigurable interconnect matrix, called eConnect, must take all ePLD outputs and connect them to all ePLD inputs and circuit outputs. The interconnect matrix is realized using multiplexers that have the select lines controlled by small distributed memory blocks as presented in Fig. 2a. The configuration memory for eConnect matrix, presented in Fig. 2b, can be implemented either with distributed memory or block memory. Because a full size interconnect is quite big and difficult to layout efficiently it was chosen to implement a sparse byte-level interconnect which requires smaller configuration memory but can complicate the CAD tools that will be built to map logic functions on the architecture.

3. Test extension for the eCPLD architecture

The eCPLD architecture can be programmed to implement combinatorial or sequential logic by writing different values in the distributed context memory.

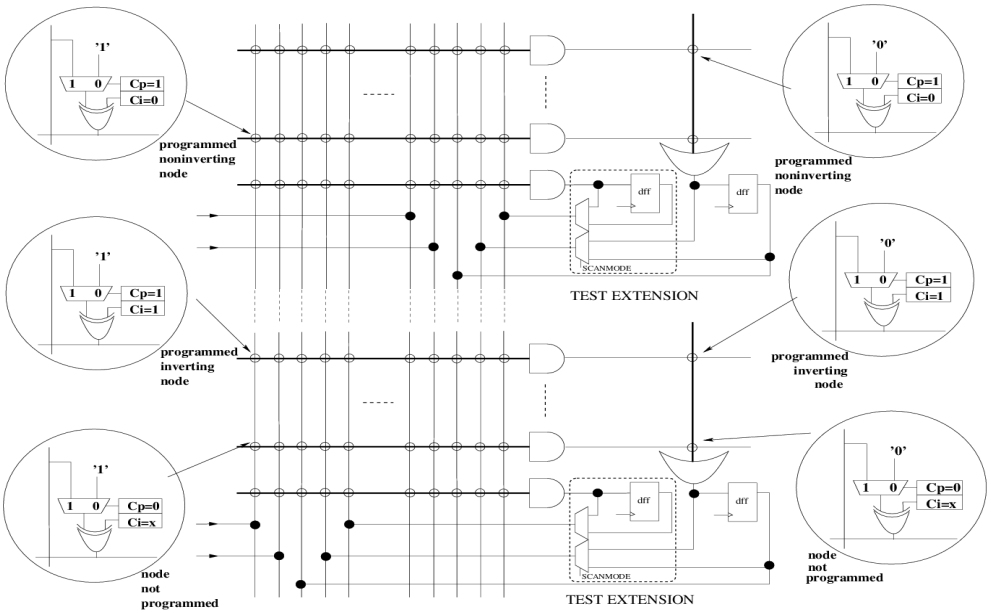

Fig. 3. ePLD with test extension.

The memory content will be generated by CAD utility software that ensures a correct programming. It can observed from Fig. 1b. that there is a possibility to create combinatorial loops when the combinatorial outputs of a product term or the expansion term are brought to the inputs of the same product term, in the case of incorrect programming of the architecture. Although a correct programming of the eCPLD architecture will not have any of these combinatorial loops active, in test mode when the logic is controlled via the ATPG scan vectors, there is the possibility to activate these loops. This will degrade the test coverage and will increase the number of required test patterns. Activating the combinatorial loops can also cause

increased power utilization during testing that can lead to device breakdown and reduced yield in production.

The solution is to break all possible combinatorial loops only in test mode but keep the same behaviour in functional mode. For each product term combinatorial output, a test extension structure is needed as presented in Fig. 3.

The test extension logic consists in two multiplexers controlled by a global test enable signal (SCANMODE) and one flip-flop used to break a possible combinatorial loop on expansion term. This can be implemented into a single structured ASIC logic cell, called eCell and each ePLD will need a number of test extension structures equal with the number of outputs. The accepted test coverage for a project implemented on structured ASIC must be greater than 98%. This test extension was implemented to remove the combinatorial loops and meet the test coverage goal for dynamic reconfigurable complex programmable logic architectures.

4. Implementation results

A medium-small sized eCPLD with 64 macro-cells grouped in 8 ePLD blocks was implemented. Two versions of dynamic reconfigurable CPLD, one that use distributed memory for storing configuration, called eCPLD64_v1, and another that use the block RAM, called eCPLD64_v2 were realized. Both versions were implemented with and without test extension to compare the test coverage and performance. The implementation was performed on structured ASIC, using eASIC NX5000 device with FC1152 package. The selected platform has 350 K cells, 175 block RAM block, 32 kbit each, and 790 available IO. The chosen synthesis strategy was the top-bottom one using the Magma Design Automation environment for structured eASIC 90 nm process. The testability was analyzed by running basic scan ATPG using Synopsys Tetramax. Table 1 presents comparative test coverage. It can be observed that initial coverage does not meet the target for structured ASIC implementation while the improved test version has much better test coverage.

Table 1. Basic scan ATPG results

Design	**Initial version**		**Improved test version**	
	Patterns. [no]	**Test coverage [%]**	**Patterns [no]**	**Test coverage [%]**
eTMCPLD_v1	260	93.27	240	98.25
eTMCPLD_v2	992	98.23	470	99.30

Table 2 and Table 3 present the overall performance results in terms of area and timing. It would be expected that the test optimized version will have worst results because of insertion of ATPG fix structure. This is not case as the logic that is added is small and the automatic placement using Magma tools can have a variation related to initial placement seed. In our case it can be observed that there are small improvements in both area and timing for the improved test version. The gain is bigger for the version that use distributed memory because the big number of small sized memories is very difficult to layout on selected structured ASIC technology.

Table 2. Comparative area results

Design	Initial version		Improved test version	
	bRAM [no]	eCell [no]	bRAM [no]	eCell [no]
eTMCPLD_v1	0	48971	0	48544
eTMCPLD_v2	8	30560	8	30431

Table 3. Comparative timing results

Design	Initial version		Improved test version	
	Intra ePLD timing [ns]	Inter ePLD timing [ns]	Intra ePLD timing [ns]	Inter ePLD timing [ns]
eTMCPLD_v1	9.117	10.303	8.099	9.569
eTMCPLD_v2	3.023	4.419	3.232	4.120

Figure 4 present the comparative layout of the two CPLD versions. The placement for the version that uses the distributed memory is more spread because of a limitation of structured ASIC technology that does not allow wide distributed memories. Thus the version that use block RAM will be the preferred version from implementation point of view but it does not provide single context change. In Fig. 4 the cell highlighted in red are used to implement the interconnect matrix, while the cells in yellow are used to implement the ePLD logic.

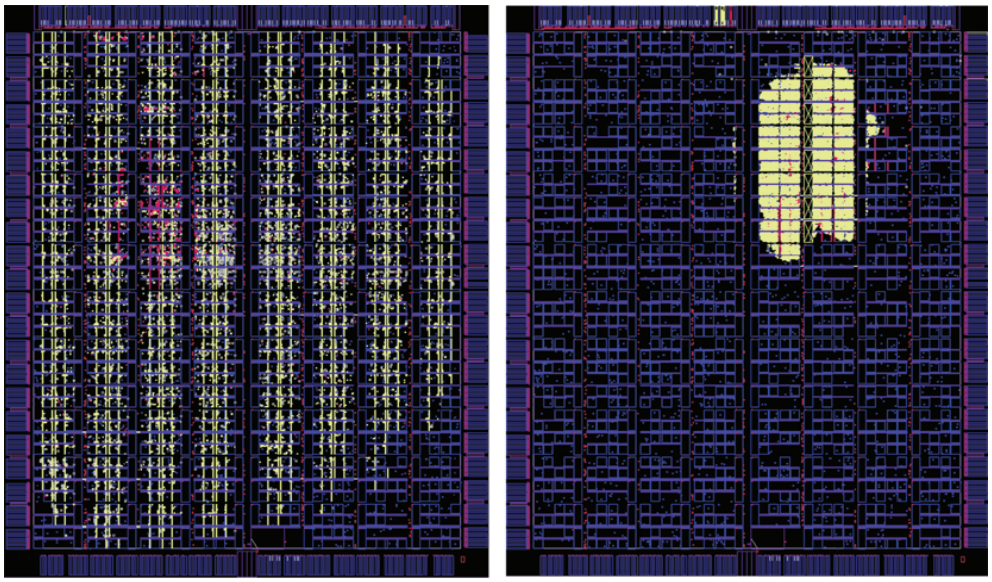

Fig. 4. Comparative layout on NX5000 device: a)eCPLD64_v1; b)eCPLD64_v2.

The solution from [6], was also implemented on structured ASIC and obtained an area of 41290 eCells and 8.027 ns inter PLD timing. The proposed architecture for structured ASIC, eCPLD64_v2, has 26% better area and with 81% better timing,

with the possibility to quickly change contexts between 16 configurations stored in on chip memory, without losing any testability due to combinatorial loops.

From area and performance results it can be observed that a dynamic reconfigurable CPLD with single clock context change is not a good option without a very wide memory that is not possible in existing structured ASIC layout. A dynamic reconfigurable CPLD that uses block RAM and has multi-cycle context change is the preferred option with good area overhead and performance.

5. Conclusions

This paper presents a new test extension for dynamic reconfigurable CPLD architectures implemented on structured ASIC technology. The optimization ensures no test coverage loss due to use of dedicated memory to store multiple configurations and to implement different functionality in time.

Experimental results proved better test coverage for proposed optimized architecture with no degradation in area and timing, and also better performance results when compared with other solutions for embedded programmable cores. The possibility to use both spatial and temporal aspects of logic circuit capacity by fast configuration change is a key feature of the proposed architecture.

References

[1] Chang D., Marek-Sadowska M., *Partitioning Sequential Circuits on Dynamically Reconfigurable FPGAs*, IEEE Transactions on Computers, **48**(6), pp. 565–569, 1999.

[2] Jipa R., *Introducing Asynchronous Structures in the Structured ASIC Platforms*, Bulletin of the Transilvania University of Braşov, **13**(48), pp. 209–212, 2006.

[3] Halfhill T., *Tabula's Time Machine, Rapidly Reconfigurable Chips Will Challenge Conventional FPGAs*, Available at: `http://www.tabula.com`

[4] Trimberger S., Carberry D., Johnson A., Wong J., *A Time Multiplexed FPGA*, IEEE Symp. FPGAs for Custom Computing Machines, pp. 34–40, 1997.

[5] Tulbure T., *A Dynamic Reconfigurable CPLD Architecture for Structured ASIC Technology*, Proceedings of 7th International Symposium on Applied Reconfigurable Computing, Belfast, pp. 296–301, 2011.

[6] Yan A., Wilton S. J. E., *Product-Term-Based Synthesizable Embedded Programmable Logic Cores*, IEEE Transactions. on VLSI Systems, **14**(5), pp. 474–488, 2006.

[7] ***, 90 nm eASIC Nextreme Overview, Available at: `http://www.easic.com`