RESEARCH PAPER | Engineering | Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

A Novel online Fault Reconfiguration of FPGA

KEYWORDS: BLRB, CLB, FPGA

B. Harikrishna
Research Scholar, Sathyabama University, Chennai, India

DR. S. Ravi
Professor &Head, Department of Electronics Engineering, Dr.M.G.R University, Chennai

ABSTRACT *Field Programmable Gate Arrays (FPGAs) have become predominant platform for large number of designs. To with stand faults that occur from different types of errors we need to build an online fault detection model and have fault recovery methods. In this paper we present a new technique for FPGA circuits developed by BLRB approach for online fault recovery. The methodology in this approach is that the spare is selected which ever spare is near for the faulty CLB and replaces structurally and functionally. By selecting the nearest spare the routing path is decreased.*

INTRODUCTION

Field programmable Gate Arrays (FPGAs) are pre-fabricated silicon devices that can be electrically programmed in the field to become almost any kind of the digital circuit or system. Depending upon the requirement a portion of the FPGA can be partially reconfigured while the rest of an FPGA is still running. Any future updates in the final product can be easily upgraded by simple downloading the new application bit stream. There are many different FPGA architectures available from various vendors for example- Altera [11], Xilinx [12]. Although the exact structure of these FPGAs varies from vendor to vendor, all FPGAs consist of the three fundamental components: Logic Blocks, I/O blocks, and the Programmable Routing. What comprises of a logic block, and how the programmable routing is organized defines the particular architecture. A logic block is used to implement the small portion of the circuit being implemented using an FPGA. The programmable routing is used to make all the required connections among the various logic block and the required connections to the I/O (input/output) blocks. Software that performs automatic routing has existed for many years. Routing terminology includes routing switch, track, routing channel etc. Routing in FPGAs consists of wire segments of varying lengths which can be interconnected using electrically programmable switches. Density of logic block used in an FPGA depends on the length and number of wire segments used for routing. Number of segments used for the interconnection typically is a tradeoff between density of logic blocks used and amount of area used up for routing. The routing architecture of an FPGA defines the following features:

1. The length of each routing wire segment (i.e.., how many logic blocks a routing wire spans before terminating),
2. Whether each routing switch is a pass transistor or a tri-state buffer,
3. Assign each net to a subset of the routing areas using global routing.
4. Use a detailed router to select specific wire segments and routing switches for each connection.

In this paper we present an efficient path routing method using nearest spare for reconfiguration of FPGA. In this approach first, we determine if the system can continue to work correctly in the presence of the located faults. In most of the situations this is possible and no reconfiguration is needed. If a fault does affect the system function, we determine the best alternate configurations that avoid the faulty resources. To enable automatic recovery of a device after damage, an autonomous BLRB algorithm is implemented and tested.

RELATED RESEARCH

In this section, a brief description about some techniques for tolerating faults in FPGAs is discussed. in this paper, we have limited the scope to reflect a few key efforts that provided some information for our work. For a more detailed, quantitative analysis of different online and offline FT techniques, see [2]. Several techniques employ column or row shifting [3], [4]. In [3], Hatori et al.introduced a single spare column for tolerating faults. They used specialized selector circuitry to reconfigure FPGA circuits in the presence of faults. Similar to methods used for FT in SRAMs,at least one additional column of PLB is faulty, its column is eliminated and all functions mapped to the columns between the faulty column and the closest spare are shifted toward the spare column algorithm .Kelly and Ivey use redundancy to bypass faults in applications mapped to FPGAs [7]. Their FT technique relies on a shift method to reconfigure in the presence of faults. They incorporate a reconfiguration switch to reconfigure and they use normal place and route (PAR) tools for mapping circuits to FPGAs. The switch matrix network makes their technique more flexible than the column and row techniques in [9] and [10]. For an average FPGA utilization or 80%, 20% of the resources should be available for spares. In [8], Cuddapah and Corba used Xilinx SRAM-based FPGAs to demonstrate the FT capabilities of FPGAs. In their study, they randomly picked PLBs to be faulty. They reconfigured the circuit around these faults using commercially available PAR tools. The main contributions of their work were an algorithm to determine fault coverage (the ability to reconfigure around a given number of faults) of a design and a definition of the fault recovery rate for any given design implemented in an SRAM-based FPGAs. Additionally, they demonstrated that fault recovery was feasible on FPGAs by other than modular redundant methods. Similar to Kelly and Ivey, their method requires some unused or spare resources, and the fault tolerance depends on how many spares are available. Dutt and Hanchek et al. developed a method to increase FPGA yield [9], [10]. Their method used node covering and reserved routing resources to replace the functionality of faulty PLBs. One row (column) of PLBs is reserved for spares. If a PLB in any given column (row) was faulty, the functionality of all PLBs in the column (row) from the faulty PLB to the spare PLB was shifted toward the spare PLB. Spare routing resources were used to eliminate overhead of rerouting the updated circuit placement. The main advantage of this method is that it is very fast relative to reconfiguration time. Since the spare resources have already been allocated to cover a limited number of faults, the reconfiguration time is linear with respect to the number of faults. Like many of the previous methods, the main problem with this offline technique is the limited number of faults that can be tolerated in each column (row). At most, they guarantee toleration of one fault per spare row or column. This paper is organized as follows section III BLRB approach

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

describing the overall flowchart. In section IV the implementation part is discussed with explanation at each stage. In section V results and discussion

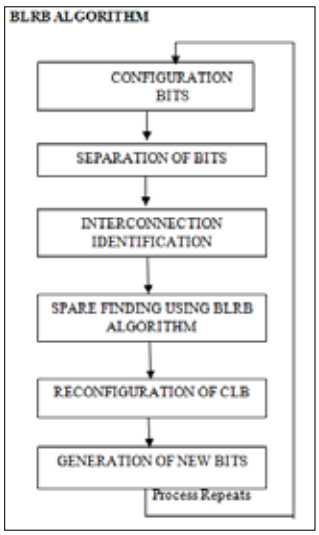

Figure 1: showing the overall implementation of approach

IMPLEMENTATION

In this BLRB (Best left right block) approach when ever any fault occurs the fault is replaced by the best near spare that is available. The algorithm is explained as follows

IV.1.SEPARATION OF BITS

We get the bit streams from the faulty resource to reconfigure the FPGA. At first the bit streams are separated by inputs, outputs and functions. A sample application corresponding to a noise filter realized on an evolved circuit is considered. The application uses 608 bits as the configuration word for a FPGA and has 64 configurable logic blocks (CLBs). The VRC decoder gives a 64 bit word as input to represent the active and spare CLBs among the 64 CLBS.

IV.2.INTER CONNECTION IDENTIFICTION

In this the CLB number to which the present CLB is connected is known and in similar way identify for all the CLBs and whole structure is known by this way. Once whole structural description is identified the spare, active CLBs are known explicitly. To extract the CLB number to which the present CLB is connected at the input side is known by using this formula

CORRECTINGFACTOR + CLB NUMBER=INPUT NUMBER. -------- Equation.1

The correcting factor depends up on the FPGA that is selected.

IV.3.SPARE SELECTION

Figure 2: Selection of spare

Finding the best spare from the spares available. Here we are finding out the best spare that is suitable for the fault .by selecting the nearest spare the interconnection path is reduced when compared to selecting any of the available spares. And latency is also reduced and reconfiguration is done in time and path delay is less by selecting the nearest spare .by reducing the path delay the time required to get the output in time may achieve. If the spare selected is not near then the path delay is more and signal from input to output is more by this there is considerable delay at the final output.

LSpare [t1] = I; RSpare [t2]=I ;

Where I indicates the first spare found near the fault. From these two LSpare and RSpare the best CLB is selected and given for reconfiguration.

IV.4.RECONFIGURTION OF CLB

Once fault location is known using well known detection and diagnosing fault the next step is reconfiguration. The reconfiguration is done to the selected spare structurally as well as functionally. The fault CLB configuration bits i.e. inputs and function bits will be copied to found spare. Now left thing is structural connection of the fault CLB connection to the spare CLB connection. The following are the steps for connection

1. Decode the CLB number to which the present CLB inputs are connected by using Equ 1.
2. Generate new input bit stream for spare using Equation 1
3. Generate reconfigured configuration bits by replacing new input bit stream and functional bits at spare location.
4. Now update the active spare bit stream indicating the faulty CLB location. By this from the input stream given by resource there will be an extra fault added and one spare less from available 64 CLB. The autonomous restructuring unit recovers the FPGA from its faults by replacing the configuration bits of faulty CLBs with the configuration bits of spare ones. For example when CLB 9 is faulty then the CLB 10 is used as a spare to repair the fault. The inputs and functions performed by CLB 9 are mapped into CLB 10. Thus the configuration bits are reconfigured autonomously.

V.RESULTS AND DISCUSSION

In this example randomly some spares and edge spare are identified and accordingly some faults are identified and by using the BLRB approach the reconfiguration is done and is shown in following figures

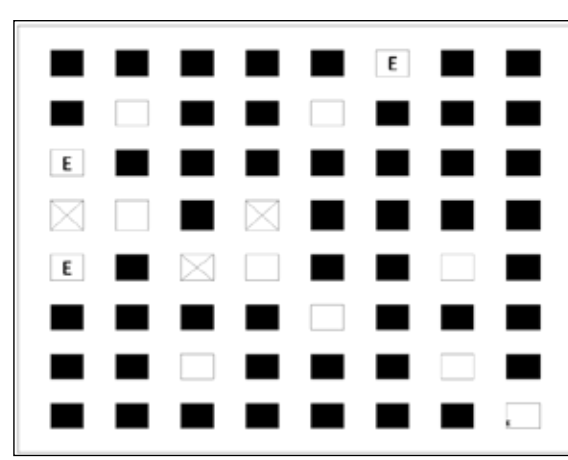

Figure3: An example of FPGA of 8x8 size. Showing the edge CLB and spare and faulty CLB.

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

faulty CLB

Spare CLB

Active CLB

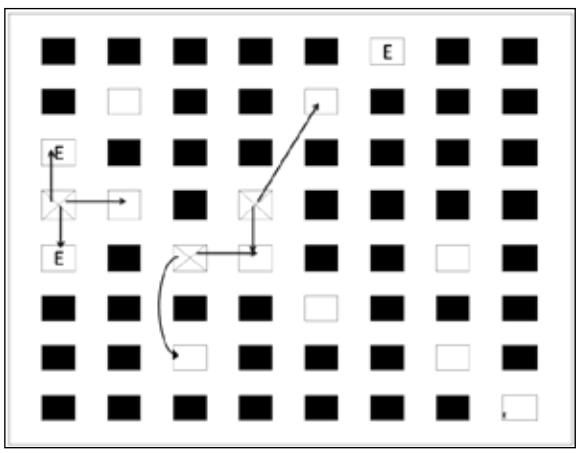

Figure 4: Showing the best possible spares for each fault.

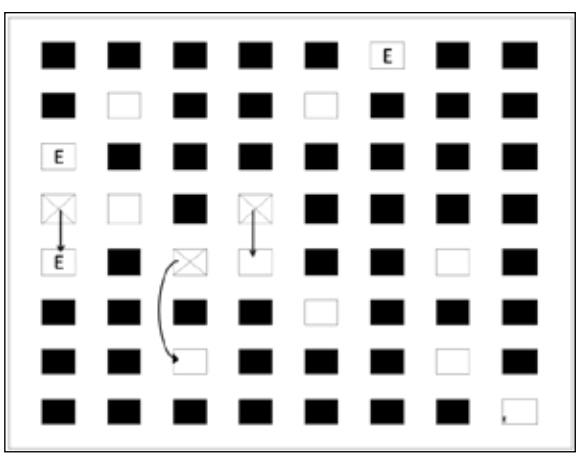

Figure 5: Selected the best spare after applying BLRB algorithm.

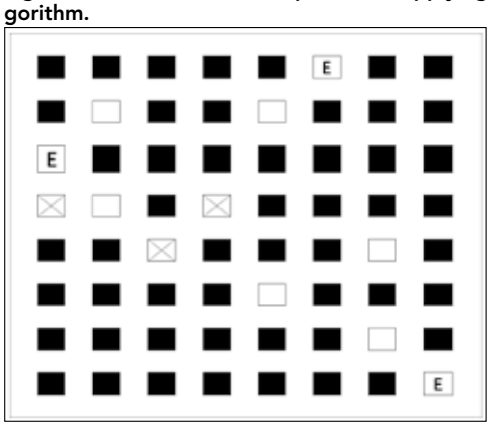

Figure 6: After reconfiguration of faulty CLB by selecting spare.

In fig 3 an example of FPGA is taken by selecting few faults randomly and spares and edge spares. After identifying the spares and faults the next process is finding the nearest spare for the fault identified .From fig 4 we can see the nearest spare and the spares identified. By using the BLRB approach the nearest spare that is found is shown in fig 5.After finding the best spare that is suitable reconfiguration is done to that faulty spare .reconfiguration is done structurally and functionally to the fault with the help of spare that is shown in fig 6.

IMPLEMENTATION RESULTS

The results obtained by implementing the algorithm presented in section IV is presented here. In this the method finds for best spare that is suitable for reconfiguration that output can be seen from fig 11.the algorithm dynamically selects the best spare and fault is reconfigured accordingly. The algorithm presented here selects the best spare dynamically

Figure 7: Input and function identification output

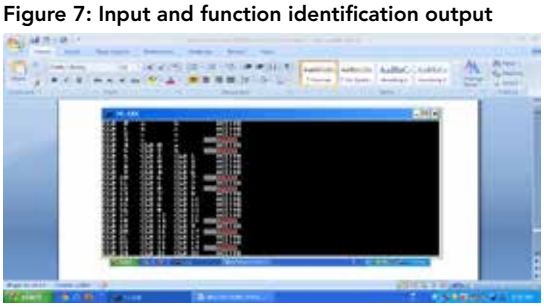

Figure 8: Identifying the active and spares output

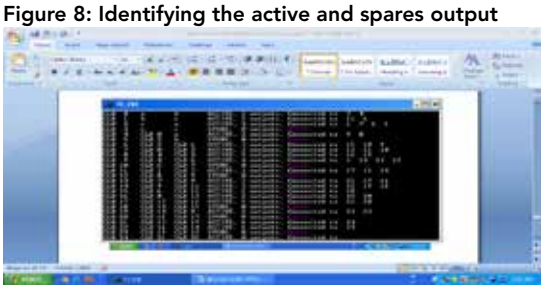

Figure 9: Structural identification

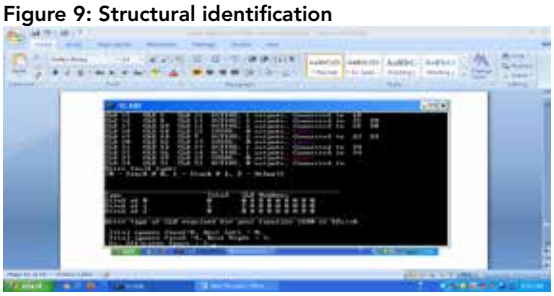

Figure 10: finding the best spare from the available spares output

Figure 11: Reconfiguration of faulty CLB with spare

Volume : 3 | Issue : 8 | Aug 2013 | ISSN - 2249-555X

CONCLUSION

In this approach a new approach to reconfigure the faulty CLB by the best Spare CLB is presented. This approach can work for multiple faults and reconfiguration is done in online. By selecting the nearest spare for reconfiguration the path between the CLB's is short even after occurrence of fault. The autonomous restructuring circuit is designed to modify the configuration word with reduced latency.

REFERENCE [1]. John M. Emmert, Charles E. Stroud, and Miron Abramovici, "Online Fault Tolerance for FPGA Logic Blocks", IEEE Transactions On Very Large Scale Integration (Vlsi) Systems, Vol. 15, No. 2, February 2007, Pp.No. 216-226. | [2] J. Cheatham, J. M. Emmert, and S. Baumgart, "A survey of fault tolerant methodologies for FPGAs," ACM Trans. Des. Autom. Electron.Syst., vol. 11, no. 2, pp. 501–533, Apr. 2006. | [3] F. Hatori et al., "Introducing redundancy in field programmable gate arrays," in Proc. IEEE Custom Integr. Circuits Conf., 1993, pp.7.1.1–7.1.4. | [4] S. Durand and C. Piguet, "FPGAs with self-repair capabilities," in Proc. ACM Int. Symp. FPGAs, 1994, pp. 1–6. | [7] J. Kelly and P. Ivey, "Defect tolerant SRAM based FPGAs," in Proc. Int. Conf. Comput. Des., 1994, pp. 479–482. | [8] R. Cuddapah and M. Corba, Reconfigurable Logic for Fault Tolerance. New York: Springer-Verlag, 1995 | [9] S. Dutt and F. Hanchek, "REMOD: a new methodology for designing fault-tolerant arithmetic circuits," IEEE Trans. Very Large Scale Integr.(VLSI) Syst., vol. 5, no. 1, pp. 34–56, Jan. 1997. | [10] F. Hanchek and S. Dutt, "Methodologies for tolerating logic and interconnect faults in FPGAs," IEEE Trans. Comput., vol. 47, no. 1, pp.15–33, Jan. 1998. | [11] Altera Inc., Data Book, 1999. | [12] Xilinx Inc., Data Book, 1999 |