Indian Journal of Electronics and Electrical Engineering (IJEEE)
Vol.2.No.1 2014pp 46-48.
available at: www.goniv.com
Paper Received :05-03-2014
Paper Published:28-03-2014
Paper Reviewed by: 1. John Arhter 2. Hendry Goyal
Editor : Prof. P.Muthukumar

STANDARD CELL BASED ANTIFUSE FPGA

Gokulakrishnan Shanmugam, Prof.S.Ramaratnam
KCG college of technology, Chennai
js.gokul@gmail.com, rcece@kcgcollege.com

ABSTRACT

Field programmable gate array (FPGA) is a programmable logic device that supports implementation of relatively large logic circuits. The standard logic elements are available for the designer by interconnect these elements to achieve the desired functional performance. In this paper we consider the most interesting and popular classes of Antifuse FPGA's. Our purpose is to provide basic concept & working of Antifuse FPGA which might help beginner designers in choosing the most for appropriate FPGA architecture for the implementation of the circuits.

Index terms-- Antifuse, Logic cells, Bitfile, FPGA

1. INTRODUCTION

The value of programmable logic has always been its ability to shorten development cycles for electronic equipment manufacturers and help them get their product to market faster. Field programmable gate arrays (FPGAs) can provide many advantages over standard cells, in terms of satisfying market demand while assuring configurability. Fast time-to- market satisfies designers to keep up with newly created standards, and configurability provides flexible hardware on demand of both new standards without fabricating a new chip.

In addition, antifuse-based FPGAs have emerged as a promising technology for limited space, high speed, and low power. The architecture consists of interconnects, antifuse switches, and programmable logic cells as shown in Figure 1. which has 26 inputs and four outputs. The function of the logic cell is determined by the logic levels applied to the inputs of the AND gates and OR gates. Antifuse FPGAs are used in a wide variety of applications ranging from GPS, portable media players, Point of sale, scanners, portable printers, Flight computers, mission computers, weapon systems.

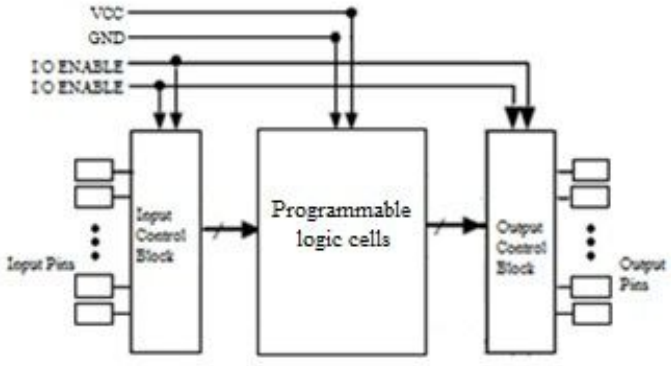

Figure 1. Block diagram

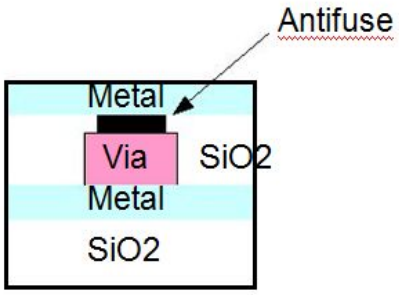

Figure 2. Antifuse switch

The figure 2 shows antifuse based cell is the highest density interconnect by being a true cross point. Thus the designer has a much larger number of interconnects so logic modules can be smaller and more efficient. Automatic place and routevsoftware also has a much easier time. These devices however are only one-time programmable and therefore have to be thrown out every time a change is made in the design.

The Antifuse has an inherently low capacitance and resistance such that the fastest parts are all Antifuse based. When unprogrammed, the insulator isolates the top and bottom layers, but when programmed the insulator changes to become a low-resistance link. It uses Poly-Si and n+ diffusion as conductors and ONO (oxide nitride oxide) as an insulator, but other antifuses rely on metal for conductors, with amorphous silicon as the middle layer.

2. HARDWARE DESIGN

The programmability, consisting of a programmable "wired" AND plane that feeds fixed OR-gates.

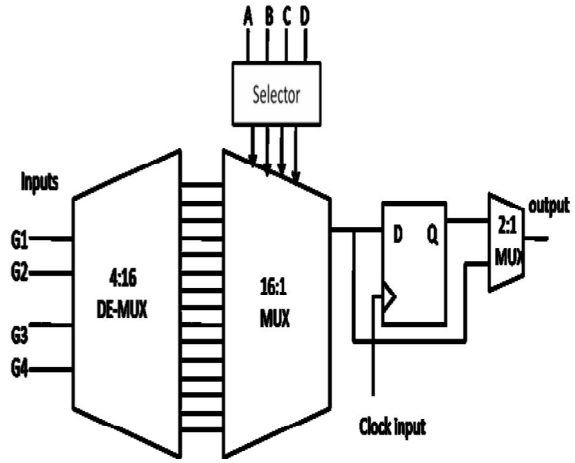

Figure 3: Hardware design

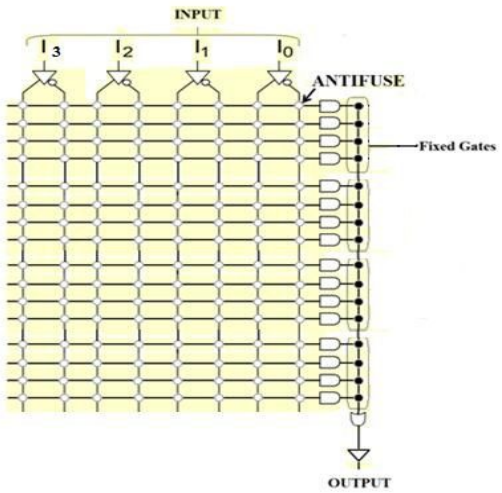

Figure 4: Programmable structure

This figure 3 it shows a hardware design in that we using an De-multiplexers its used to convert all serial data's into parallel data's and multiplexers are used to combine many input into one output. Inputs to gates are given from the Peripheral interface controller which can be connected to PC through Universal serial bus. A matrix of input signals (true and inverse via input buffers) is therefore formed at the input to the AND array with a second matrix being formed at the input to the OR array. The D-flip flop which used to perform sequential operations in FPGA's. Therefore these hardware design is sufficient to perform mostly all digital operation.

This mode of construction is ideally suited to implementing logic expressions of the form :-

F =BCD + ABCD + ABCD + ⋯(REDO AS EQUATION)

often referred to as Sum of Products expressions.

This figure 4 connections to gates in each array matrix can be fixed or programmable. A fixed connection is hardwired whilst a programmable connection is implemented by a fusible link. Devices are manufactured with all fusible links intact, the device is customized to implement the required logic functions by blowing those fuses. The required function that can be done by software design, So user can fuse a link by selecting a required node in software.

3. SOFTWARE DESIGN AND DEVELOPMENT

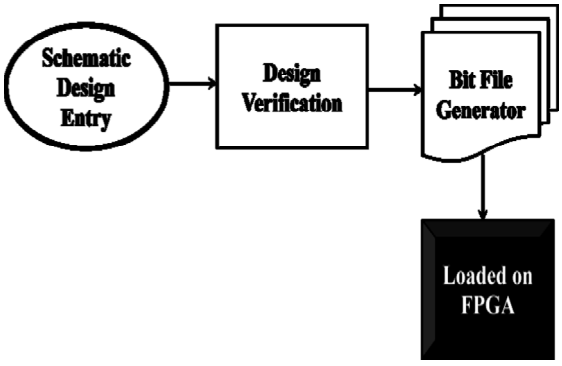

Figure 5: Software design

The process of creating digital logic is not unlike the embedded software development process. A schematic capture is an option for design entry. All of the input signals are processed by fusing, as they travel through a set of execution engines each one a send in series to macrocells and interconnections toward their destination output signals. Therefore, the schematics of a design are create structures, all of which are "executed" at the very same time.

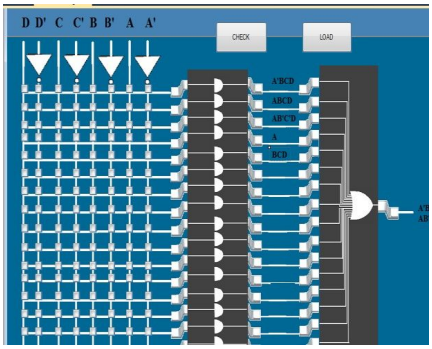

Figure 6: Software design snapshot

Typically, a design check is used to verify and execute the design and confirm that the correct outputs are produced for a given set of inputs. The designer can at least be sure that his logic is functionally correct before going on to the next stage of development. The next step is a load which actually macrocells, interconnections, and input and output pins. This process is similar to the equivalent step in the development of a printed circuit board, this result of the process is a bitstream. This name is used generically, despite the fact that each FPGA has its own, usually proprietary, bitstream format. Suffice it to say that the bitstream is the binary data that must be loaded into the FPGA to cause that chip to execute a particular hardware design. The hardware design as it executes in the programmable logic device. Obviously, this type of integration of device-specific information into a generic simulator requires a good working relationship between the chip and simulation tool.

4. CONCLUSION & FUTURE WORK:

In this paper, we presented 4input and 1output -based Antifuse FPGA. The associated design tools are based on a standard cell compatible schematic design flow using software. This approach allows the entire flow to be executed at the user's site. Antifuse based FPGAs consume much less power, then terms of area and speed, and both were significant improvements, it can only be programmed once.

The metal-to-metal antifuse FPGA is so far the best programmable switch FPGA device. However, the relative difference in sophistication between FPGA devices and FPGA software described .And future work we also plan to compare different families of FPGAs by implementing identical designs with different devices and studying cost/performance tradeoffs.

REFERENCES

1. D. Marple and L. Cooke, "Programming Antifuses in CrossPoint's FPGA," *Proc. IEEE Int'l Custom Integrated Circuits Conf.*, IEEE, Piscataway, N.J., 1994, pp. 185-188.

2. *Field-Programmable Gate Array Technology*, S. Trimberger, ed., Kluwer Academic Publishers, Norwell, Mass., 1994. Discussion of three FPGA/CPLD architectures.

3. Application Specific Integrated Circuits, by smith, Pearson Education India,1997.

4. Introductory VHDL: from Simulation to Synthesis, by Yalamanchili, Pearson Education India.

5. FPGA synthesis using Look-Up Table and Multiplexor Based architectures

6. A new logic minimization method for multiplexor- based FPGA synthesis.

7. BOOKS:Microsoft Visual Basic .Net Step by Step.
8. by Michael Halvorson