International Journal of Innovative Research in Computerand Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

Development of Programmable Logic Devices

Sourabh V Bhat[1], Rainak Sharma[2], Pooja M P[3], Priyanka P[4], Pavan Kumar E[5]

Student, Department of Electronics and Communication Engineering, Sai Vidya Institute of Technology Bangalore, India[1,2,3,4]

Assistant Professor, Department of Electronics and Communication Engineering, Sai Vidya Institute of Technology Bangalore, India[5]

ABSTRACT: In this paper, we have discussed about the development of programmable logic devices, the programming techniques. An insurgency is occurring over the previous decade. The most extreme number of gates in an FPGA is at present around 20,000,000 and multiplying at regular intervals, this we call it as Moore's Law. PLD's are the primary chips that could be utilized to actualize an adaptable digital logic plan in hardware. The rise of PLD has improved the presentation of the procedure as well as has decreased the time expended for the assembling. The interest in expanding the functionalities and its complexity has likewise lead to the improvement of ports, busses, memories that are multifunctional in nature. The lines of source code that they compose are constantly executed in a specific order. On the off chance that there is a working framework it is utilized to make the presence of parallelism as the info signals are handled in parallel. Thusly, the statements of a hardware description language structures which are "executed" at exactly the same time. The different applications incorporate motor control squares, power electronic control, movement control inside the framework plan, fuzzy controller, artificial neural network-based controller, state machines, synchronization, decoders, counters, bus interfaces, parallel to serial, serial to parallel, subsystems, and so on.

KEYWORDS: PLD, ASIC, FPGA, HDL

I. INTRODUCTION

PLDs have risen to be incredible in different spaces including the motion control applications, control building, mechanical gadgets, power change insightful engine, acceptance engines, lasting magnet AC engines, exchanged hesitance engines, stepper engines, and numerous others not just with the assistance of field-programmable Gate array (FPGA), hardware description languages(HDL), complex programmable logic devices (CPLD) yet additionally as they acquired the front end configuration apparatuses and thought behind application-specific integrated circuits(ASIC). The rise of programmable logic devices has prompted the assembling of quick double converter current circle which really lead to the wide discovery with respect to the presentation of the gadget which was at that point existing previously.

This has likewise assumed an indispensable job in decreasing the over-burden on the processor as the vast majority of the guidelines are executed without the requirement for processor mediation and consequently diminishing the heap on the processor. At the point when the digital signal processor is interfaced with PLDs the impressive measure of CPU burden can be decreased. [1] Programmable logic devices have altered how digital circuits are manufactured, FPGAs, and CPLDs have become the guidelines for actualizing digital frameworks. FPGAs and CPLDs offer a lot higher circuit thickness, improved dependability. FPGAs and CPLDs execute answers for digital plan issues rapidly and financially, the two characteristics that are significant in a modern setting. PLD is a mix of a logic gadget and a memory gadget. The memory is utilized to store the example that was given to the chip during programming. SRAM is an unpredictable sort of memory, implying that its substance is lost each time the force is turned off. SRAM-based PLDs, in this manner, must be customized each time the circuit is turned on. This is generally done naturally by another piece of the circuit. These are less board space, quicker, lower power necessities, less expensive get-together procedures, higher dependability, and accessibility of structure software.[2]

The standard way to deal with equal controller configuration utilizes successive controller plan methods. The strategy is shown with the plan of an equal controller for a transputerinterface connector and its usage on a PLD'S. PLD'S are first chips that could be utilized to execute an adaptable digital logic structure in Hardware. programmable circuits are elective for standard components of medium and huge combination scale. These days PLD'S circuit contains memory squares, particular processor squares, and even analog squares. FPGA's and CPLD's offer a lot higher circuit thickness, improved unwavering quality. Movement control application includes the use of HDL, field-

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

programmable gate array, complex PLDs. The crisis of PLD and utilization of it in the critical situation has empowered to lessen the heap on the microprocessor and thusly empower effective and quick handling of the instructions.[3]

II. ARCHITECTURE

A. *OKITO ARCHITECTURE*

OKITO architecture is one among the various architectures which are present and as in the case of other architectures there is a tradeoff between various factors like speed and controlling ability. This architecture is well known for the controlling applications rather than the speed of processing. Hence the regular and timely maintenance of the architecture with minimal consumption of time and its software adaptability to application specific scenario is a vital factor. The C language is used in order to define its functionality.

Architecture consists of the following components:

1.Bus
2.Control block
3.Control processes and processor connection to control block
4.Template interface to bus
5.Connection interface for third party IP-block(Intellectual Property) connections
6. Test modules
7. Interface between OPB (On-Chip Peripheral Bus) and OKITO-bus
8. Software processor, C-coded user module
9. User module

The dependence of flexibility of architecture on the configurations can be overcome using the parameters like number of bits which can simultaneously travel i.e. not only address bits but also data bits, number of blocks and functionalities addressed in each block along with the involvement of amount of registers, and the block addresses.
Each module is an independent block. Control processes make the connection between modules and generate the system activities.

Data transfer is exclusively done through registers between each module which are configurable. Also there exists special registers for every module. The used bus clock frequency and data width become the parameters which have the ability to alter the efficiency of the bus. The theoretical rate of 88Mbits/s is achieved making the assumptions on the frequency of the bus clock to be 24MHz and the amount of data sent would be 16 bits. Based on the magnitude of the data transfer involved there are chances of choosing between the need of s separate connection or the usage of common bus for the operation.

Since this architecture aims at the controlling application it has to work on reducing the data load and increasing the speed of data transfer. Hence there is a limitation on assumptions being made during theoretical calculations as practical implementation is a function of block functionalities and amount of memory units involved which is estimated using the below equation

$$R = 13Nm + 7.5Nr + 423Np + 50,$$

where R is needed recourses in slices (Xilinx) for the OKITO-architecture,
Nm is modules numbers,
Nr is registers numbers
Np is processor connections numbers.

It is just an approximate model because the synthesis settings have an influence on the final implementation. [4]

III.PROGRAMMING TECHNIQUES

Programmable switches play a key role is altering the functionalities of FPGA. Size of the programmable switch, on-resistance, and capacitance direct form the important properties which need to be altered depending on the application. In this segment we portray the most ordinarily utilized programmable switch innovations and toward the end will balance every innovation concerning instability, re-programmability, size, arrangement on-opposition, parasitic capacitance, and procedure innovation complexity.

A. SRAM Programming Technology

Xilinx [6], Plessey [7] Algotronix, [8], Concurrent Logic [9] and Toshiba [10] are some of the key players in the domain who make the usage of controlling action of Static RAM cells on multiplexers and thereby apply SRAM programming as shown in Fig3.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

At the point when one is put away in the SRAM cell in Fig. 3(a), the pass gate goes about as a shunt switch and can be utilized to make an association between two wire fragments. At the point when a low signal is put away, it leads to an open circuit and in turn the transistor between the two-wire fragments offers infinite resistance. For the multiplexer, the condition of the SRAM cells associated with the select lines controls which one of the multiplexer inputs is associated with the yield, as appeared in Fig. 3(b).
The volatile nature of the SRAM, burdens the FPGA as it expects to be powered at regular intervals. PROM, EPROM, EEPROM, or magnetic disk which provide the programming bits for outer changeless memory. But the usage of SRAM not only comes with a drawback of its bulky nature but also involves more transistor usage which is turn leads to high heat dissipation due to the extreme opposition in cut-off condition. However bearing these constraints will open door to impressive re-programmability with the minimum usage of just standard IC process innovation.

B. Anti-fuse Programming Technology

An anti-fuse is a device which has the ability to showcase infinite amount of resistance between its 2 terminals when it is not defined its functioning. The application of high voltage would lead the anti-fuse to "blow" and thereby facilitate a low resistance interface. This connection is perpetual. Anti-fuses being used today are fabricated either utilizing an Oxygen-Nitrogen-Oxygen (ONO) dielectric between N+ dispersion and poly-silicon [191, [151, [11 or poly-silicon between metal layers [13] or among polysilicon and the principal layer of metal [14].

The additional hardware requirement in order to define the actions to an anti-fuse through its programs will cater to the high requirements of voltage and current as seen in [11] through pass transistors which are adequately sizeable to give tending to every anti-fuse.[12]. Some of the key manufacturers like Actel [151[I], Quicklogic [13], and Crosspoint [14] use the FPGA's which avail the antifuse technology.

A little size and the minimal cross-sectional separation of two metal wires is the advantage of using anti-fuse technology. The usage of bulky transistors which have vital role in programming and those transistors which are meant for creating necessary separation will play a vital role in ensuring that the high voltage won't affect the low programming transistors as they are closely placed. A subsequent significant favorable position of an anti-fuse is its moderately low arrangement resistance. As seen in [13] the resistance of ONO anti-fuse when its on would be in the range of 0.3 KΩ to 0.5 KΩ , while the Si anti-fuse amounts to 0.05 KΩ to 0.1 KΩ. The good thing about it is that the capacitance which is automatically created due to flaws in fabrication or design can be easily over come as they are in a negligible amount.

C. Floating Gate Programming Technology

These are the key techniques which are employed not only in the most commonly used Erasable EPROM but also in the most recent advancement i.e. Electrically Erasable EEPROM devices. The Altera [15] and in plus Logic [16] employs the usage of floating gate for the purpose of programming and thereby implement the usage of EPROM.

The semiconductor unit which functions as a device whose working can be controlled, outlined in Fig. 6, can be forever "disabled" by applying appropriate biasing on it, utilizing a logic 1 between the 1_{st} controlling device and the output terminal of the semiconductor unit which in turn enhances the restricted potential difference of the semiconductor device with the goal that it kills. The exposure of the central terminal to Ultraviolet spectrum would lead to expelling of the charge which in turn reduces the restricted potential difference of the semiconductor device and there by makes it to 0work regularly.

When the "word line" is connected to high voltage, EPROM plays the role of a switch with a defined set of operations and thereby it grounds the "piece line" as outlined in Fig. 6. This methodology can not only be used to make a coupling between the set of 16 bit and single bit lines, but also plays a key role in implementing a hardware circuit of functionality like that of AND gate, along these lines giving both logic and routing.

Similarly the wide usage of any technique over a long duration with maximum flexibility is achieved through its ability to be programmed multiple number of times with minimum efforts. The usage of this technology overcomes the need to redefine the functioning every time its programmed. EPROM innovation, nonetheless, requires three extra handling ventures over a standard working of a Complementary MOS device. Two different impediments are the infinite amount of opposition offered by an EPROM semiconductor device and the high static force utilization because of the draw up resistor utilized (see Fig. 6).

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

The process of defining the operations based on EEPROM is utilized in the devices from AMD [16] and Lattice [16]. However this means that it should be capable of functioning with electrical without the usage of UV light. This gives the additional favorable position of simple programmability, that turns out to be exceptionally useful in certain use cases, for example, the necessary updating to the hardware to gear in remote areas. The bulky nature of EEPROM cell can be approximately equational to double the size of EPROM cell.

V. RESULT

The table I below shows the comparison of various programming technologies studied. It involves the usage of CMOS fabricated using 1.2 µm technology. It also comments on the volatile nature of the respective technologies, area occupied by the circuits of those technologies and also the reprogramming capability. There is a tradeoff between all these parameters to select the appropriate technology for the respective application.

Technology and process	Volatile	Reprogramming Capability	Area
SRAM Mux Pass Transistor	YES	Yes in circuit	Large
Amorphous Antifuse	Nix	Nix	Fuse small (via) Prog. Tran Large
ONO Antifuse	Nix	Nix	Fuse small (via) Prog. Tran Large
EPROM	Nis	Yes out of Circuit	Small in array
EEPROM	Nix	Yes in Circuit	2x EPROM

Table 1: COMPARISON OF PROGRAMMING TECHNIQUES

IV. CONCLUSION

Later on, we hope to see a significantly more noteworthy expansion of engineering developments just as research that responds to essential inquiries concerning FPGA design. We expect these to include:

A. Advancement of "unique reason for existing" FPGA's tuned toward explicit applications, for example, circuits which make way for data flow, applications in which the signal is being processed in the digital domain, machines with limited number of stages in its functioning, and Field Programmable Gate Array-based figure motors [5].

B. The advancement of field-programmable frameworks embedded with multiple units for not only copying, but also increasing speed, and fast prototyping[5]

C. Advancement of nonhomogeneous rationale square structures. These hold the guarantee of giving better are execution trade offs on the grounds that bigger granularity squares accomplish better execution while littler granularity squares accomplish higher thickness.

D. Divided steering models give significant execution and thickness preferences over non segmented designs (those with just a couple of lengths of the fragment). New understanding is required in deciding fragment length dissemination.

E. Looking at the amazing contributions of the FPGA to the digital or the discrete domain the analog domain has made its counterpart named Field Programmable Analog Array.

Also, PC supported plan instruments for FPGA's will turn out to be progressively complex, empowering the advancement of design highlights planned for improving execution and thickness.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

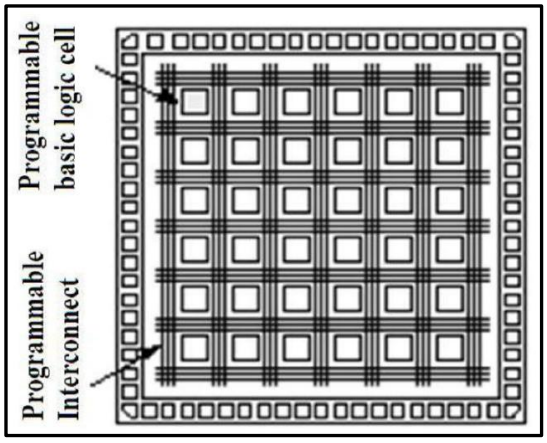

Fig.1. FPGA Architecture

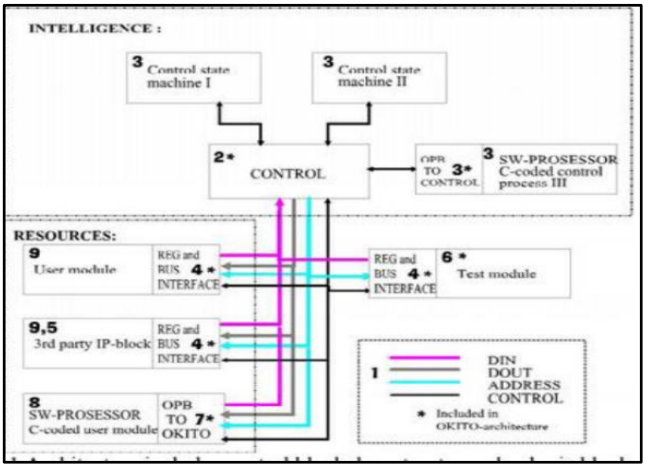

Fig. 2. OKITO Architecture

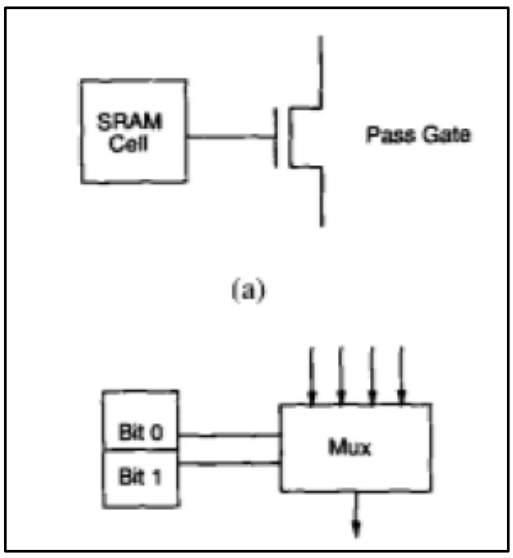

Fig. 3. Static SRAM Programming Technology

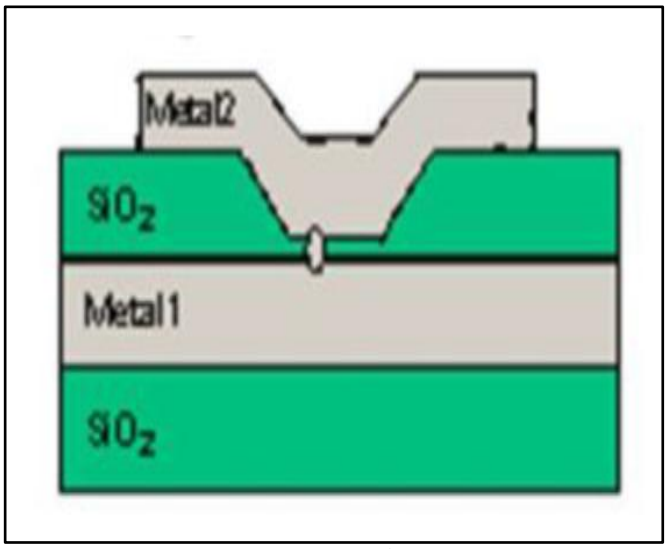

Fig 4 Metal-Metal Antifuse

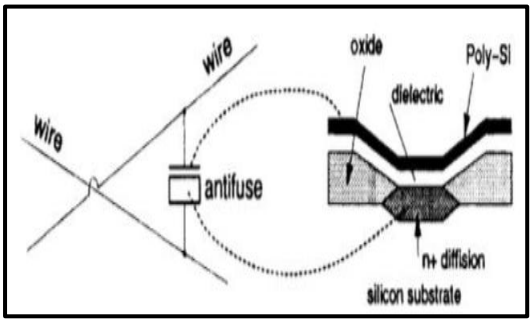

Fig. 5. Actel Anti-Fuse Structure

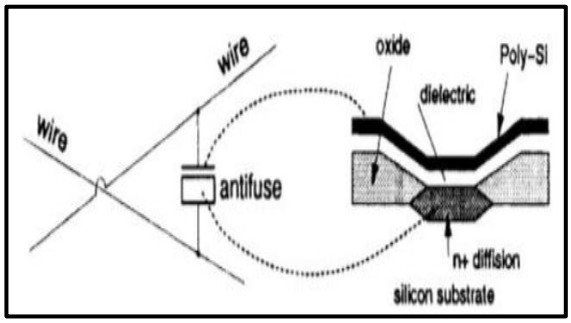

Fig. 6. Floating Gate Programming Technology

VI. ACKNOWLEDGEMENT

We thank the college management, principal of Sai Vidya Institute of Technology for having given us the support, guidance in effective utilization of this opportunity. We thank the Honorable, Vice Chancellor, VTU for encouraging us in doing such research work.

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | Impact Factor: 7.194 |

||Volume 8, Issue 4, April 2020||

REFERENCES

[1] Rahul Dubey, Pramod Agarwal, M. K. Vasantha, "Programmable Logic Devices for Motion Control—A Review," IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, VOL. 54, NO. 1, FEBRUARY 2007

[2] Zdenek E. Skokan, "Programmable Logic Machine," IEEE JOURNAL OF SOLID-STATE CIRCUITS, VOL. SC-18, NO. 5, OCTOBER 1983

[3] James Pardey, Alain Amroun,t Martin Boltont and Marian Adamski, "Parallel controller synthesis for programmable logic devices," Microprocessors and Microsystems Volume 18 Number 8 October 1994

[4] James Pardey, Alain Amroun,t Martin Boltont and Marian Adamski, "Parallel controller synthesis for programmable logic devices," Microprocessors and Microsystems Volume 18 Number 8 October 1994

[5] P. Bertin, D. Roncin, and J. Vuillemi, ""Third-generation architecture boosts speed and density of field-programmable gate array," ACM First international Workshop on Field-Programmable Gate Arrays, pp. 57-59, Feb. 1992

[6] H. Hsieh, et al, "Programmable active memories: Performance measurement, in Proc. 1990 CICC, pp. 31.2.1-31.2.7, May 1990

[7] Plessey Semiconductor ERA60100 preliminary data sheet, Swindon, England, 1989.

[8] CAL 1024 Datasheet, Algotronix Ltd. Edinburgh, Scotland, 1989

[9] Concurrent Logic CFA6006 Field-Programmable Gate Array Dura Sheet, Sunnyvale, CA, 1991.

[10] H. Muroga, H. Murata, Y. Saeki, T. Hibi, Y. Ohashi, T. Noguchi, and T. Nishimura, "A large scale FPGA with 10K core cells with CMOS

0.8 pm 3-layered metal process," Custom Integrated Circuits Con6 '91, CICC, pp. 6.4.1-6.4.4, May 1991.

[11] A. El Gamal, et al., "An architecture for electrically configurable gate arrays," IEEE JSSC, vol. 124, No. 2, pp. 394-398, Apr. 1989

[12] J. Greene, "Antifuse field programmable gate array," Proc. IEEE, vol. 81, no. 7, July 1993..

[13] J. Birkner, A. Chan, H. T. Chua, A. Chao, K. Gordon, B. Kleinman, P. Kolze, and R. Wong, "A very high-speed field programmable gate array using metal-to-metal anti-fuse programmable elements," in New Hardware Product Introduction at CICC '91.

[14] D. Marple and L. Cooke, "An MPGA compatible FPGA arCA, pp. 361-367. chitecture," in FPGA 92, ACM First In?. Workshop on FieldProgrammable Gate Arrays, pp. 3944, Feb. 1992.

[15] S. C. Wong, H. C. So, J. H. Ou, and J. Costello, "A 5000-gate CMOS EPLD with multiple logic and interconnect arrays," in Proc. I989 CICC, pp. 5.8.1-5.8.4, May 1989.

[16] Plus Logic FPGA2040 Field-Programmable Gate Array Data Sheet, Plus Logic, San Jose, CA, 1989.