International Journal of Scientific Research & Engineering Trends
Volume 4, Issue 4, July-Aug-2018, ISSN (Online): 2395-566X

A Review Article of FPGA ALU Unit Design Based on GA

Sakshi Samaiya
M. Tech. Scholr of VLSI Department of (E&C)
ADINA Institute of Science and Technology, Sagar (M.P.)

Anupreksha Jain
Asst. Prof. of VLSI Department of (E&C)
ADINA Institute of Science and Technology, Sagar (M.P.)

Abstract – **The paper primarily deals with review for the construction of arithmetic Logic Unit (ALU) using Hardware Description Language (HDL) using Xilinx Vivado 14.7 and implement them on Field Programmable Gate Arrays (FPGAs) to analyze the design parameters. ALU of digital computers is an aspect of logic design with the objective of developing appropriate algorithms in order to achieve an efficient utilization of the available hardware. Speed, power and utilization of ALU are the measures of the efficiency of an algorithm. In this paper, we have simulated and synthesized the various parameters of ALUs by using VHDL on Xilinx Vivado 14.7 and Basys 3 Artix 7 FPGA board.**

Keywords: — **FPGA, ALU, XILINX Vivado 14.7, Basys 3 Artix 7 FPGA board.**

I. INTRODUCTION

The Design and implementation of FPGA based Arithmetic Logic Unit is of core significance in digital technologies as it is an integral part of central processing unit. ALU is capable of calculating the results of a wide variety of basic arithmetical and logical computations [1]. The ALU takes, as input, the data to be operated on (called operands) and a code, from the control unit, indicating which operation to perform [2]. The output is the result of the computation. Designed ALU will perform the following operations: A. Arithmetic operations B. Bitwise logic operations all the modules described in the design are coded using VHDL which is a very useful tool with its degree of concurrency to cope with the parallelism of digital hardware. The top level module connects all the stages into a higher level at Register Transfer Logic (RTL). RTL describes the requirements of data and control units in terms of digital logic to execute the desired operations. Each instruction from the architecture's instruction set is defined in detail in the RTL [3-4]. Once identifying the individual approaches for input, output and other modules, the VHDL descriptions are run through a VHDL simulator and then is downloaded the design on FPGA board for verification[4]. As FPGA has an application that it can incorporates much logic on a single FPGA. So as floating point ALU has many operations to be performed in the computer we are using an FPGA IC to implement it. The operations performed by the FPU are addition, subtraction, multiplication, division and logical operations as AND, OR, NOT etc[5]. FPU mainly work on Real as well as integers value.FPGA is an integrated circuit designed to be configured by the customers or designer after manufacturing- hence "Field Programmable" [6]. The FPGA configuration is generally specified using a hardware description language, similar to that used for an application specific integrated circuit (ASIC). FPGA contain programmable logic components called "Logic Blocks", and a hierarchy of reconfigurable interconnects that allows the block to be wired together. Logic blocks can be configured to perform complex combinational function or merely simple logic gates like AND and OR [7]. In most FPGA's, the logic blocks also include memory elements which may be simple flip flops or more complete blocks of memory.

II. LITERATURE REVIEW

Jijil Kurian: Arithmetic logic unit is the core of any CPU that can be part of a programmable reversible computing device such as a quantum computer. The major concern for ALU design, using normal gates is heavy power consumption. The main reason for power consumption is the normal irreversible gates. In order to ensure low power design constraint a new type of gates called reversible gates were introduced. In reversible gates the number of inputs is equal to the number of outputs and there is a one to one mapping between the inputs and outputs. Here in this paper we discuss the design of a low power ALU using reversible gates and its implementation on FPGA.

Lijo Antony Alex: Moore's law provides a platform for the development of integrated circuits and the law will stop its function sooner. So some dramatic evolution has to happen in microelectronics in near future. Power consumption of CMOS circuits has to become a major problem as the complex digital circuits being built turns to much faster and complex.Landauer [1961] provides proofs

for the fact that power loss is an inevitable drawback of irreversible circuits [1].

Bennett [1973]: proved that incorporation of reversible gates keeps a circuits from dissipating any power. The major feature of reversible gates is that, the number of inputs for a reversible is same as the number of outputs. Not only that, there is one to one mapping between the inputs and outputs of a reversible circuit [2]. Arithmetic logic unit is the vital part of CPU since it allows computer to perform arithmetic and logic operations. In simple sense arithmetic logic unit is a combinational logic circuit having one or more inputs and a single output. Which implies that the present value of output depend only on the present input values only. The complexity of Arithmetic logic unit depends on the processor variations. It can be a simple one or complex structure. Thomson [2010] designed an arithmetic logic unit made up of reversible gates for performing modular operations [3].

Y Syamala et.al [2011]: designed a reversible ALU for performing logic and arithmetic operations here in this paper we propose a new design of ALU made up of reversible gates with better power saving property.

Venugopal G: In this work reversible ALU for a computing device is designed for performing one arithmetic operation and three logic operation. Our design shows better quality in the design parameters like number of gates number of garbage outputs and quantum cost .The proposed ALU is expressed in VHDL hardware description language and in implemented on Spartan 3 FPGA.

Earlier, the ALU and full adder circuit was implemented for area and delay, each with their distinct features that bring about best area and delay. Out of them some was briefly described to give us an idea of earlier work and different optimization techniques. Past work gives us a suggestion about its operation, performance, design and simulation issues. In G. Karthik Reddy[1], the 1-bit ALU was design using the Pass Transistor Logic (PTL) technology and 1-bit full adder is used based on low power 6Transistors. ALU circuit was designed using 6T full adder cell and Pass Transistor Logic which was based on logic blocks such as AND, OR, XOR. The simulation was done on 65nm single n-well CMOS bulk technology, in virtuoso platform of cadence tool. As per the nature of work, they have applied the technique which may have faces the problem in designing the ALU with such technology and it was also not easy to reduce the power. In Geetanjali1 and Nishant Tripathi2 [2], the 32-bit ALU was implemented using VHDL and simulation was carried out by using modelsim 5.4a tool. The ALU was designed to perform the arithmetic operation such as addition, subtraction, increment, decrement, transfer and logical operation such as AND, NOT, OR, NAND, NOR, EX-OR, EX-NOR. How the operation take place is shown by behavioral modeling style, the behavioral capabilities of VHDL was more powerful and more convenient for this design. As per the nature of behavioral modeling it is easy to implement the 32-bit ALU. This paper has the limitations for the making the changes in the parameters. In Liril George1 and Padmaja Bangde2 [3], the 32-bit ALU was design using VHDL and power consumption was gained by the clock gating technique. The complete design was implemented on Xilinx Spartan 3E FPGA. The clock gating is mostly used in synchronous circuits, therefore it faces lots of problem in reducing the power consumption. He showed that there would be no lower limit for power consumption. It also reduces the dynamic power consumption by 10 percent compared to the power consumed by a 32 bit ALU without clock gating. Somewhere it may be difficult problem in reducing the power consumption with this technology.

III.ARCHITECTURE

The proposed 16-bit ALU of the RISC processor consists of one arithmetic unit and logic unit. Using the concept of regularity, the arithmetic unit is divided into different blocks which perform the operations such as: addition, subtraction, multiplication, division. But, the main focus of concern in this arithmetic unit is the multiplier unit which has been implemented using Booth algorithm and bit pair recording techniques.
The reason for choosing this technique is to increase the speed of 16-bit ALU. Similarly, the logic unit is also divided into various blocks which perform the operations such as: AND, OR, NAND, NOR, NOT, XOR, XNOR, INCREMENT, DECREMENT, ROTATE LEFT AND ROTATE RIGHT.

The ALU will be design for arithmetic and logical operation. It will comprises the addition, AND logic, XOR logic, and XNOR logic using 4:1 multiplexer. All the operation will be successfully designed in the tanner tool using BSIM4 model which will give the better performance. The very important part of ALU which determine the overall performance of the design is the full adder. To reduce the power consumption from ALU full adder plays an important role. To provide the efficient way to design the 32-bit ALU with low power consumption, the adiabatic logic is more efficient. Adiabatic logic achieve the low power consumption to decrease the flow of current across the devices with minimum voltage drop to reuse the energy stored in there capacitor.

Optimization problems

In a **genetic algorithm**, a population of candidate solutions (called individuals, creatures, or phenotypes) to an optimization problem is evolved toward better solutions. Each candidate solution has a set of properties

(its chromosomes or genotype) which can be mutated and altered; traditionally, solutions are represented in binary as strings of 0s and 1s, but other encodings are also possible.
The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a *generation*. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.
A typical genetic algorithm requires:

1. a genetic representation of the solution domain,
2. A fitness function to evaluate the solution domain.

A standard representation of each candidate solution is as an array of bits.[2] Arrays of other types and structures can be used in essentially the same way. The main property that makes these genetic representations convenient is that their parts are easily aligned due to their fixed size, which facilitates simple crossover operations. Variable length representations may also be used, but crossover implementation is more complex in this case. Tree-like representations are explored in genetic programming and graph-form representations are explored in evolutionary programming; a mix of both linear chromosomes and trees is explored in gene expression programming.
Once the genetic representation and the fitness function are defined, a GA proceeds to initialize a population of solutions and then to improve it through repetitive application of the mutation, crossover, inversion and selection operators.

IV.EXISTING WORK ON EFFICIENT ALU MODULE

A new algorithm based design technique is suggested by Suchita Kamble, Prof. N.N. Mhala focused on the design for 8 bit ALU and it was implemented using VHDL Xilinx Synthesis tool ISE 13.1. ALU was designed to perform arithmetic operations and logical operation. The maximum propagation delay is 13.588ns and power dissipation is 38mW. They have implemented lower number of bit i.e. 8- bit [1]. A novel algorithm to design and implementation of low power 16-bit ALU with clock gating is presented by Ankit Mitra. Authors has include the power optimization at architectural level which demonstrated by the design of a 16- bit ALU using clock gating. The arithmetic unit of the ALU is based on a carry skip adder to reduce carry propagation delays. It is observed that clock gating reduces dynamic power dissipation of the ALU approximately [4]. Geetanjali, Nishant Tripathi introduced the VHDL implementation of 32-bit ALU. They suggest the behavioral design method for VHDL implementation of a 32-bit ALU using Modelsim 5.4a tool. Its functionality was discussed for all the operations specified. As per the nature of behavioral description, it is easy to convert the precision to 64- bit or more. But, this paper has some limitations regarding the improvement in the various parameters [3]. P Bhanusree,G Bhargav Sai,Y Ashwanth Kumar,K Sravan Kumar have designed VHDL implementation of 64-bit ALU. They introduced the behavioral modeling and structural modeling used for the implementation of 64-bit ALU. All the mathematical operations in the ALU are performed by means of repeated additions. Along with the basic operations of ALU multiplication and comparison are incorporated and designed as a single unit. The module is designed and implemented using VHDL and is simulated using Xilinx9.2i ISE [5].

V. BASIC OF MICROPROCESSORE

Like all good things, this powerful component i.e., Microprocessor is basically very simple. It is made by mixing tested and high- quality "ingredients" (components) as per following recipe:
1. The simplest computer processor is used as the "brain" of the future system.
2. Depending on the taste of the manufacturer, a bit of memory, a few A/D converters, timers, input/output lines etc. are added.
3. All that is placed in some of the standard packages.
4. Simple software able to control it all and which everyone can easily learn about has been developed. On the basis of these rules, numerous types of Microprocessors were designed and they quickly became man's invisible companion. Their incredible simplicity and flexibility conquered mankind a long time ago and if one tries to invent something about them, one should know that they are probably late; someone before you, has either done it or at least has tried to do it [6].

Operation of Microprocessor Even though there are a large number of different types of Microprocessors and even more programs created for their use only, all of them have many things in common. Thus, if you learn to handle one of them you will be able to handle them all. A typical scenario on the basis of which it all functions is as follows:
1. Power supply is turned on and everything starts to happen at high speed! The control logic unit keeps everything under control. It disables all other circuits except quartz crystal to operate. While the preparations are in progress, the first milliseconds go by.
2. Power supply voltage reaches its maximum and oscillator frequency becomes stabilized. SFRs are being

filled with bits reflecting the state of all circuits within the Microprocessor. All pins are configured as inputs. The overall electronics starts operation in rhythm with pulse sequence. From now on the time is measured in micro and nanoseconds.

3. Program Counter is set to zero. Instruction from that address is sent to instruction decoder which recognizes it, after which it is executed with immediate effect.

4. The value of the Program Counter is incremented by 1 and the whole process is repeated several million times per second.

As one can observe, all the operations within the Microprocessor are performed at high speed and quite simply, but the Microprocessor itself would not be so useful if there are not special circuits which make it complete. The main components which aid in making any Microprocessor powerful are as follows.

•Read Only Memory (ROM) Read Only Memory (ROM) is a type of memory used to permanently save the program which is being executed.

•Random Access Memory (RAM) Random Access Memory (RAM) is a type of memory used for temporary storing data and intermediate results created and used during the operation of the Microprocessors.

•Electrically Erasable Programmable ROM (EEPROM) The EEPROM is a special type of memory not contained in all Microprocessors. Its contents may be changed during program execution (similar to RAM), but remains permanently saved even after the loss of power (similar to ROM)[7].

•Special Function Registers (SFR) Special function registers are part of RAM memory. Their purpose is predefined by the manufacturer and cannot be changed. Since their bits are physically connected to particular circuits within the Microprocessor, such as A/D converter, serial communication module etc., any change of their state directly affects the operation of the Microprocessor or some of the circuits. For example, writing zero or one to the SFR controlling an input/output port causes the appropriate port pin to be configured as input or output. In other words, each bit of this register controls the function of one single pin [8-9].

•Program Counter Program Counter is an engine running the program and points to the memory address containing the next instruction to execute. After each instruction execution, the value of the counter is incremented by 1. For this reason, the program executes only one instruction at a time just as it is written [10-11].

•Central Processor Unit (CPU) as its name suggests, this is a unit which monitors and controls all processes within the Microprocessor and the user cannot affect its work. It consists of several smaller subunits, of which the most important are:

1. Instruction decoder 2. Arithmetical Logical Unit (ALU) 3. Accumulator
2. Input/output ports (I/O Ports) In order to make the Microprocessor useful, it is necessary to connect it to peripheral devices. Each Microprocessor has one or more registers (called a port) connected to the Microprocessor pins.

VI. OLD RESULT AND SIMULATION

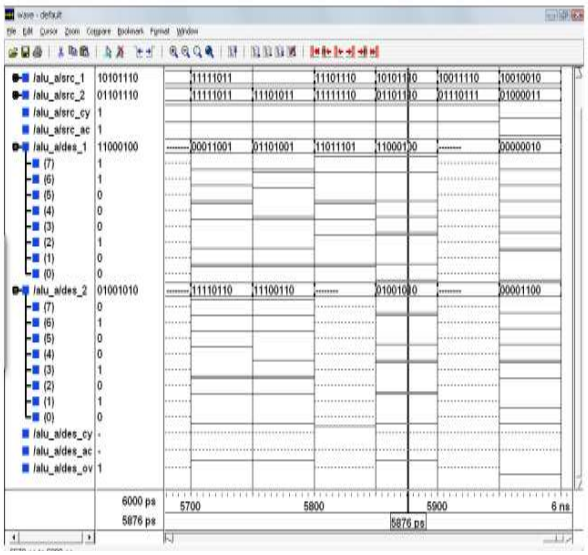

Figure 1: Simulation result for Arithmetic and Logic Unit (ALU)

V. CONCLUSION

An efficient operational unit could be designed with the help of simulation tool and the functional units with optimized count of slices, count of flip-flops and input LUTs. The maximum frequency can be obtained from the analysis of timing view. Power count of the system should be proportional to the respective frequency. Time delay can be optimized using Vedic mathematics technique in implementation of multipliers as the speed of ALU depends prominently on the speed of multiplier. Further in the future work the speed of ALU can be increased by using various fast and efficient multipliers available in the literature.

REFERENCES

[1] G. Karthik Reddy (2015) "Low Power-Area Pass Transistor Logic Based ALU Design Using Low Power Full Adder Design" IEEE Sponsored 9th (ISCO)

[2] Geetanjali1 and Nishant Tripathi2 (2012). "VHDL Implementation of 32-Bit Arithmetic Logic Unit (ALU)." IJCSCE Special issue on "Emerging Trends in Engineering" ICETIE

[3] Liril George1 and Padmaja Bangde2 . (2014, July). "Design and Implementation of Low Power Consumption

32 Bit ALU using FPGA." VOLUME-1, ISSUE-5, IJREST

[4] N. Ravindran, R. Mary Lourdes. (2015, May 23). "An Optimum VLSI Design of A 16-Bit ALU." 978-1-4799-8966- 9/15/$31.00 IEEE

[5] Weng - Geng Ho, Kwen - Siong Chong, Bah- Hwee Gwee, and Joseph S. Chang. (2013, March). " Low Power Sub Threshold Asynchronous QDI Static Logic Transistor-level Implementation (SLTI) 32-bit ALU." 978-1-4673-5762- 3/13/$31.00 IEEE

[6] N. Anuar, Y. Takahashi and T. Sekine(2009,March) "Two phase clocked adiabatic static logic circuit: a proposal for digital low power applications". Proc.IEICE Gen. Conf.

[7] Suma T.Hegde, Dr. Siva Yellampalli and Nandeesh R, "Design and Implementation of ALU using Redundant Binary Signed Digit", International Conference on VLSI, Communication & Instrumentation (ICVCI) 2011, Proceedings published by International Journal of Computer Applications® (IJCA), pp: 30-35.

[8] Geetanjali and Nishant Tripathi, "VHDL Implementation of 32-Bit Arithmetic Logic Unit (ALU " International Journal of Computer Science and Communication Engineering, IJCSCE Special issue on "Emerging Trends in Engineering" ICETIE 2012. Pp: 41-44

[9] Mahendra Kumar and Kailash Chandra "Low Power High Performance SRAM Design Using VHDL", Global Journal of Researches in Engineering, Volume 11 Issue 1 Version 1.0 February, 2011, ISSN: 0975-5861, pp: 21- 24

[10] Shabana Aqueel and Kavita Khare, "Design and FPGA Implementation of DDR3 SDRAM Controller for High Performance" , International Journal of Computer

[11] Prof. Vikas Gupta, Dr Kavita Khare and Dr R. P. Singh "Efficient FPGA Design and Implementation of Digital PID Controllers in Simulink®", International Journal of Recent Trends in Engineering, Vol 2, No. 6, November 2009, pp:147-150