

Design of Energy Efficient ALU on FPGA

Vatsala Sharma^{*}, Kamal Nayanam^{*}, Subham Shukla[#] and Nitin Bhatia^{*}

^{*}Department of Electronics & Communication
BMIET, Sonapat, Haryana, India

[#]Department of Electronics & Communication
Surabhi Group of Institution, RGPV Bhopal, India

Date of Submission: 07-07-2020

Date of Acceptance: 21-07-2020

ABSTRACT: FPGA based Arithmetic Logic Unit has critical role in all processors; it is an inbuilt part of Central processing unit. Arithmetic Logic Unit is used to calculate the outputs of a large form of basic logical and arithmetic computations. In this paper, design and implementation of an optimized 64 bit ALU is proposed whose size can be reduced effortlessly to the 16 bit or 32 bit ALU. The proposed design is implemented using two level optimization, initially the resource utilization of FPGA is decreased with the help of recycling and reusing them for variety of operations. As a result of reduced FPGA utilization, FPGA resources will be saved as well as power consumption is also reduced. Different types of blocks are designed in order to carry out 16 operations. In final stage of optimization only single block is active at an instance and remaining blocks are inactive, which decreases the dynamic power utilization of the device and the design obtained is much energy efficient.

KEYWORDS: ALU; FPGA; Clock gating; 64-bit, Energy efficient;

I. INTRODUCTION

Clock gating (CG) is a capable way to deal with reduced utilization of power in any circuit. CG approach utilized by numerous researchers [1] and is generally outlined at gate level by power unify mechanism. In clock gating approach we adequately utilized controlled clock pulses on both sequential and synchronous circuits, which are for the most part utilized as a part of processors. For the most part, CG approach is planned as coordinated CG cells [2]. It keeps up clock tree in such a way that it utilizes less chip area of the hardware, so diminishing in internal switching. This will expand sparing of power, i.e. already acquired by inward exchanging. It additionally expends less chip zone, as its replaces clock gating rather than multiplexers [3]. With headway in innovation, the quantity of transistor depend on a solitary CPU has expanded.

Increasing the number of transistors leading the power dissipation in the gadget [4] [5]. Since the majority of the convenient gadgets runs on battery power. The power consumption of these gadgets should be less such that the life of the battery enhances, dependability enhances and so on. In light of these reasons, control administration has turned into vital outline requirements for much computationally concentrated as well as complex application. ALU is a standout amongst the most vital units in a microchip based gadgets, it also executes a large portion of the computational function in a CPU and henceforth control utilization is an essential consequence in an arithmetic logic unit. Two primary modes of power dispersals occurring in CMOS circuit are static power and dynamic power. Static power is caused because of leakage current and dynamic power is produced because of charging and discharging of capacitor or because of exchanging exercises of circuit [6] [7]. Khurana & Kaur, 2013 applied unlatched latch free clock gating methods in ALU to reduce clock power and dynamic power consumption of ALU [8]. Khaturia *et al.*, 2015 deals with the synchronous design, which revolve with high frequency which runs a huge load as a large number of serial components needs to be passed throughout the chip [9]. Pandey & Pattanaik, 2016 includes the design and implementation of a Low Power Arithmetic and Logic Unit using CG technique which can be utilized as part of low power processor design [10]. Emmett & Biegel, 2017 deals with design procedure to minimize ASIC power utilization with the implementation of RTL CG approach in Synopsys Power Compiler, because of this property disabled clock components have CG approach applied by default, that decreases the usage of power by those components to zero, at the instance when their values are constant [11]. George & Bangde, 2018 noticed that CG decreases the dynamic power utilization by 10% with respect to the power utilized by the 32 bit ALU with no CG [12]. Singh & Goel in 2019 gives the survey of trends in clock gating

methods in ALU design and implementation as well as their pros and cons along with a demonstration of D flip-flop and 4 bit false random binary series generator. In this paper we utilized tri-state rationale rather than clock gating procedure [13].

II. METHODOLOGY

ALU is the essential section of the microprocessor and the main functional block of central processing unit. ALU incorporates the combinational logic that carries out the logical functions which includes 'AND', 'OR', etc. and mathematical functions which include Addition, Multiplication, etc. That's way it can perform mathematics operations as well as Logical operations [14] [15]. Fig. 1 suggests the symbolic illustration of an ALU and Fig. 1 shows the functions of an ALU. First four functions are the logical operations and the next four are the arithmetic operations.

The bits need to conduct the arithmetic and logical capabilities are specified as inputs, and known as operands, from the assigned CPU registers [16] [17]. The ALU is based on fundamental items to carry out its calculations. Those consists of number systems, information expelling circuits (adders/subtractors), timing, instructions, operands and registers. Fig 1 shows a representative block diagram of an ALU.

ALU can exhibit two types of functions; one of them addresses arithmetic calculations known as Arithmetic Unit which can execute addition, subtraction, multiplication, division, increment and decrement. Other type consider gated outputs in terms of AND, OR, XOR, etc. that is known as logical unit [18] [19]. The operations are fully manipulated and performed by the input of select lines or control bits. Generally, Arithmetic and logic unit opcode is distinct from an opcode of machine language, but in few situations ALU opcode can immediately be converted to a machine code known as opcode [20]. Additional information is allowed due to status input that is sent to the ALU when executing a function. At last, single or many status results obtain auxiliary indexes grading the output of an ALU function that could be useful in future.

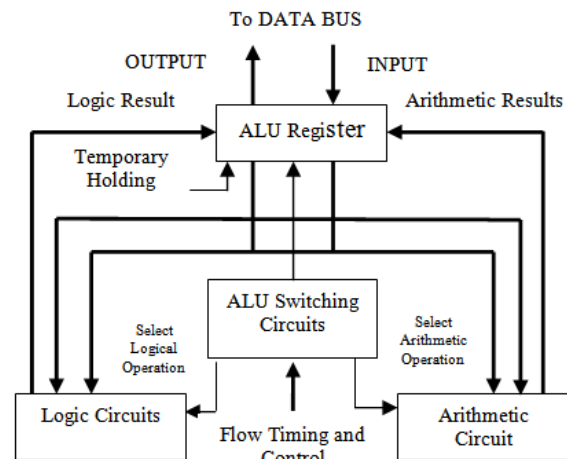


Fig 1.1 Block diagram of an ALU

In this paper we use tri-state buffer logic approach to optimization of an ALU. In tri-state logic, we have to activate one operation (which is working) at a time, remaining all other operations (which isn't working) are in high impedance state i.e. in buffer state. This will decrease the power utilization of the section.

In preceding researches and designers essentially used 3 styles of strategies to prevent extra switching activity in ALU i.e. clock gating, clock allow, blocking off the inputs [21].

For clock gating, we used only clock control alerts signal. In this approach, because of clock gating putting logic factors in clock tree isn't advocated due to additional skew and the problem in producing a glitch unfastened sign, whilst Clock permitting technique the use of the 'clock permit' enter of the flip-flop. It simply turnoff the power of flip-flop and logic but doesn't reduce the power of clock tree. Third technique primarily based on disabling the data path or input route. That is performed by means of using tri-state buffer logic.

III. IMPLEMENTATION

ALU has the input variables A, B, Clk (clock), and selection. Results obtained from ALU are represented as Z and flags. There are different flags are generated and it depends upon the operations result.

There are sixteen operations, so 8 modules are implemented. Among all of them, there are basically four operations that approximately has similar characteristics and can be implemented with same adder, they are: Addition with and without carry, Subtraction with and without borrow, Decrement and Increment. As mentioned earlier, the subtraction operation can be performed by

implementing 2's complement. One has to use such feature and design it with the help of adder block. In order to evaluate $A - B$, it can be simply designed as $A + (2\text{'s complement of } B)$. In this case smaller amount of FPGA resources are utilized by implementing common adder block. In the same manner, Increment ($A + 1$), Decrement ($A - 1$), Addition with carry ($A + B + 1$) and Subtraction with borrow ($A - B - 1$) is designed with the help of adder block as well as 2's complement block. As per the discussed concept, six functions are designed utilizing common adder block and 2's complement block, and that will give result less hardware resources usage. The block diagram of ALU is shown in figure 2.

Different blocks are implemented in order to execute various functions. At an instance, single function is to be executed on the basis of the measure of "selection". So one can use these features of ALU to get profit, by enabling single block at a given instance as well as deactivating all remaining blocks, as a result the internal switching actions of deactivated blocks because all remaining 7 blocks are inactive on that time and help in reducing dynamic power consumption. Deactivating the unwanted blocks, dynamic power utilization can be considerably decreased that is due to reduction in internal switching actions at the bottom of an Arithmetic logic unit.

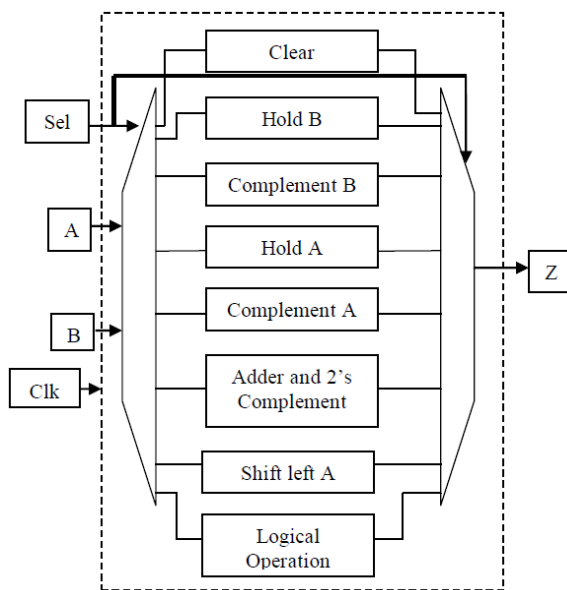


Fig. 2: High level design flow of ALU implementation

A digital circuit consists of triple levels logic '1', logic '0' and HIGH impedance 'Z'. When

the circuit is in high impedance level Z, which behaves as an open circuit i.e. output would appear to be disconnected and would have no logic significance. This circuit can perform any conventional logic (AND or NAND) and is also commonly used to implement multiplexers.

The gate has two inputs-the normal input A and control input C. The value of output Y is decided by these two inputs.

- If $C = 1$; the gate is active and result $Y = A$ (either 1 or 0)
- If $C = 0$; the gate is inactive and result $Y = Z$ (HIGH impedance)

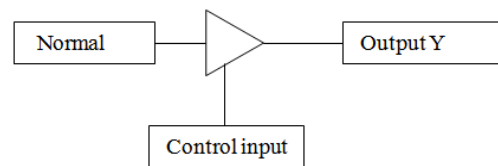


Fig. 3: TRI-State Buffer Logic with Control Input

With the help of this concept we can implement MUX.

To implement 4X1 MUX we use a 2X4 decoder. Decoder is required to appropriate select the control inputs. Here, decoder selects the appropriate control input. When $S_1 = 0$ and $S_2 = 0$ then it select the control inputs 1000 for all 4 input blocks respectively. So only input 1 is active at a time, while all other input block are in high impedance state (open circuited) and disconnect to output Y, so at the output only input 1 is obtained. When $S_1 = 0$ and $S_2 = 1$ then it will select control input 0100 for all 4 input blocks respectively. So only input 2 is active at a time, While all other input block are in high impedance state (open circuited) and disconnect to output Y, so at the output only input 2 is obtained.

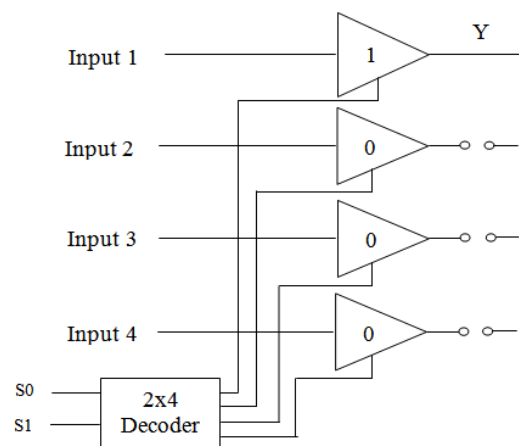


Fig. 4: 4X1 MUX

When $S_1 = 1$ and $S_2=0$ then it will select control input 0010 for all 4 input blocks respectively. So only input 3 is active at a time, while all other input block are in high impedance state (open circuited) and disconnect to output Y, so that the output contains only input 3.

When $S_1 = 1$ and $S_2=1$ then it will select control input 0001 for all 4 input blocks respectively. So only input 4 is active at a time, While all other input block are in high impedance state (open circuited) and disconnect to output Y, so at the output only input 4 is obtained.

Table 1 Truth table of 4x1 MUX with tri-state logic

S_1	S_2	CONTROL BITS	OUTPUT (Y)
0	0	1000	INPUT 1
0	1	0100	INPUT 2
1	0	0010	INPUT 3
1	1	0001	INPUT 4

IV. SIMULATION RESULTS

Set default all values which are given to inputs in hexadecimal notation.

Initially all the inputs are at high impedance state. If assigned value of input 1 is '5' and input 2 is '2' and select the operation 4 (binary equivalent "0100") i.e. addition of input 1 and input 2, and clk is '1' then we get ADD(input 1 + input 2) = 7 at the output. The output values as estimated by ISIM tool by Xilinx. Simulation result is illustrated in Fig. 5, where at a time only one operation is executed by ALU, remaining all 7 functions are in a high impedance state show by blue line. This is done by demux. If sel "100" are choose ($S_0 = 0, S_1=0, S_2= 1$) then only ADD operation are selected at the output of the demux.

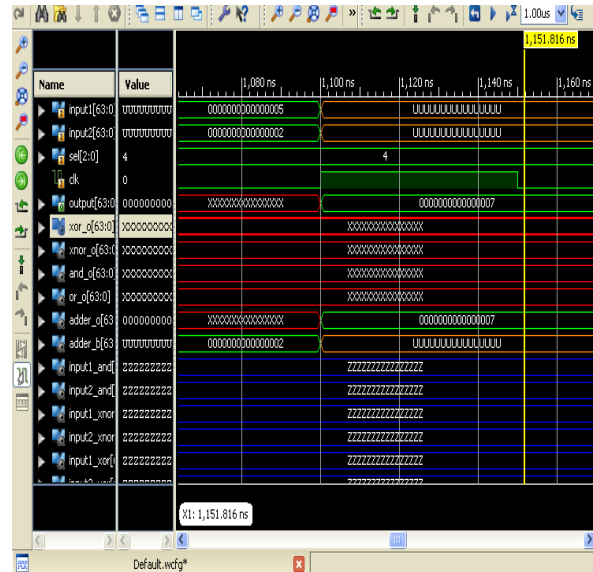


Fig. 5: Simulation of de-mux when sel "100"

If assigned value of input 1 is "4" (in hexadecimal) for both increment and decrement operation and select operation "110" for increment operation and assigned value of select operation is "111" for decrement operation, that will select increment and decrement operation and gives the output '5' and '3' respectively.

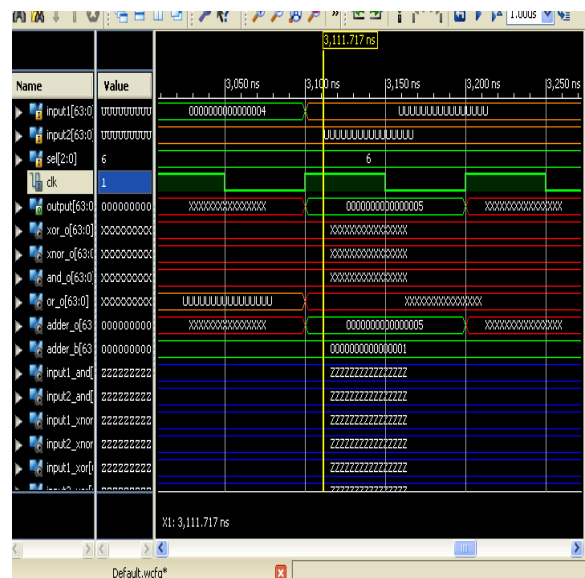


Fig. 6: Simulation of an ALU when sel "110"

Table 2: Synthesis Report

ON CHIP	CONVENTIONAL DESIGN		PROPOSED DESIGN	
	POWER (mW)	USED	POWER (mW)	USED
SIGNAL	6.63	752	4.0	736
IOS	35.98	197	31.0	196
CLOCK	0.16	-	2.0	1
LOGIC	0.76	496	1.0	415
STATIC POWER	45.32	-	45.0	-
DYNAMIC POWER	43.53	-	38.0	-
TOTAL	88.85	-	83.00	-

The proposed design is implemented using Xilinx 14.1 on Spartan 3 FPGA xc7k70t-2fbg676. Xpower analyzer evaluates the performance of the design in terms of power consumption. ISIM feature of Xilinx is utilized to execute behavioral simulation. The performance is measured in terms of power consumption at various frequencies and illustrated simultaneously. The idea of tri-state logic is implemented here, that forces to switch off all the remaining blocks that are not in use in the implementation of present selected function, for instance, let us implement an addition function which only need single block "ADD" and remaining blocks are not needed, therefore the inputs of the remaining blocks are tri-stated which does not permit the output capacitor to discharge. As a result, the internal switching power consumption of the FPGA is decreased which causes the proposed design more energy efficient. The power report of this proposed design is depicted in table 2. It is clear from the analysis of results between conventional design and proposed design, dynamic power is significantly reduced i.e. from 43.5mW to 3mW. Total power consumption is also reduced from 88.85mW to 83mW.

V. CONCLUSION AND FUTURE SCOPE

ALU is the main elemental part of processor, and updating an ALU, in sense of power reduction, can enhance the capability of processor. This can be attained from reduction in power consumption and FPGA resource consumption. As we seen from the simulation result and power report that our model are working as per the specifications. We have applied different sets of input and output sets are verified. In all the circumstances we obtained correct result as per specification. We can

also conclude from power report that with the help of tri-state logic, disabling the unwanted blocks, so dynamic power consumption can be reduced at large extent; this is because of decrease in internal switching activities inside ALU.

Next improvement we tried to implement a design in which less FPGA resource used; we removed few blocks such as subtract, increment, decrement, add with carry and subtract with borrow and implement all these functions using single adder and 2's complement block. This fulfills our two purposes; first, reduction of FPGA resource usage and second reduction in power consumption.

As we can see from the synthesis report that our design has low power consumption as compared to base paper design. The delay response of our design is much better than the base paper delay response. This is because we have made certain changes in algorithm.

There are certain amendments that we have made to improve the design response in terms of delay, power and area. But still many improvements can be made.

Power response: The power consumption of the design can still be improved. In this work we use simple add/shift multiplier. We can use advance version of multiplier circuit such booth multiplier. In booth's multiplication algorithm, no partial products are generated. So with the help of booth multiplication algorithm, we can improve the performance of multiplier. This will leads to reduction in power consumption.

Delay: In the last stage of bit multiplier is ripple carry adder (RCA), this RCA is slow as the final output is obtained only when all the carry bits are generated. This will add some delay in the design, this can be improved by using fast adders like carry look ahead adder (CLA), Kogge-Stone adder, carry skip adder or any other fast adder.

REFERENCES

- [1]. J. P. Oliver, J. Curto, D. Bouvier, M. Ramos and E. Boemo, "Clock gating and Clock enable for FPGA power reduction", 8th Southern Conference on Programmable Logic (SPL), pp. 1-5, 2012.
- [2]. Sato, Fumiki, and Kouichi Fujita. "Arithmetic and logic unit", U.S. Patent No. 5,442,801. 15 Aug. 2005.
- [3]. Hyeongseok Yu and Jun-Dong Cho, "Low-power design and architecture", Potentials *IEEE*, vol.20, no.3, pp.18-22, 2019.
- [4]. Jitesh Shinde, Dr. S.S.Salankar, "Clock Gating – A Power Optimizing Technique for

- VLSI Circuits”, India Conference (INDICON), 2011 Annual IEEE, pp. 1–4, 16-18, 2015.
- [5]. Dushyant Kumar Sharma, “Effects of Different Clock Gating Techniques on Design”, International Journal of Scientific & Engineering Research Volume 3, Issue 5, May 2012.
- [6]. Bishwajeet Pandey, Jyotsana Yadav, M Pattanaik, Nitish Rajoria,” Clock Gating Based Energy Efficient ALU Design and Implementation on FPGA”, IEEE, 2013.
- [7]. V. Khorasani, B. V. Vahdat, and M. Mortazavi, “Design and implementation of floating point ALU on a FPGA processor”, IEEE International Conference on Computing, Electronics and Electrical Technologies (ICCEET), pp.772-776, 2012.
- [8]. Shikha Khurana, Kanika Kaur, “IMPLEMENTATION OF ALU USING FPGA”, International Journal of Emerging Trends & Technology in Computer Science (IJETTCS), Volume 1, Issue 2, July – August 2013.
- [9]. Jagrit kathuria, M.Ayoubkhan, Arti Noor, “A Review of Clock Gating Techniques,” MIT International journal of electronics and communication engineering, vol.1 no.2, PP.106-114, 2015.
- [10]. Bishwajeet Pandey and Manisha Pattanaik, “Clock Gating Aware Low Power ALU Design and Implementation on FPGA”, International Journal of Future Computer and Communication, Vol. 2, No. 5, 2016.
- [11]. Frank Emnett and Mark Biegel, "Power Reduction through RTL Clock Gating" SNUG San Jose 2017.
- [12]. Liril George and Padmaja Bangde, “Design and
- [13]. Implementation of Low Power Consumption 32 Bit ALU using FPGA”, international journal for research in emerging science and technology, volume-1, issue-5, 2018.
- [14]. Priya Singh, Ravi Goel, “Clock Gating: A Comprehensive Power Optimization Technique for Sequential Circuits”, International Journal of Advanced Research in Computer Science & Technology, ISSN: 2347 – 8446, Vol. 2, Issue 2, 2019.
- [15]. S. Cisneros, J. J. Panduro, J. Muro, and E. Boemo, “Rapid prototyping of a selftimed ALU with FPGAs”, International Conference on Reconfigurable Computing and FPGAs, pp. 26-33, 2019.
- [16]. Hubert Kaeslin, ETH Zurich, Digital Integrated Circuit Design from VLSI Architectures to CMOS Fabrication, Cambridge University Press, 2018.
- [17]. Vojin G. Oklobdzija, Vladlmlr M. Stojanovic, Dejan M. Markovic, Nikola M. Nedovic, DIGITA L SYSTEM CLOCKING High-Performance and Low-Power Aspects, Wiley Interscience, U.S., 2003.
- [18]. Haghparast Majid and Ali Bolhassani, "Optimization Approaches for Designing Quantum Reversible Arithmetic Logic Unit", *International Journal of Theoretical Physics* 55.3: 1423-1437, 2016.
- [19]. L. Benini, G. De Micheli, E. Macii, M. Poncino, and R. Scarsi, “Symbolic Synthesis of Clock-Gating Logic for Power Optimization of Synchronous Controllers”, ACM Trans. Des. Autom. Electron, 2000.
- [20]. Vojin G. Oklobdzija, Vladlmlr M. Stojanovic, Dejan M. Markovic, Nikola M. Nedovic, DIGITA L SYSTEM CLOCKING High-Performance and Low-Power Aspects, Wiley Interscience, U.S., 2003.
- [21]. P. J. Shoenmakers, J. F. M. Theeuwen, “Clock Gating on RT- Level VHDL”, Proc. Of the int. Workshop on logic synthesis, Tahoe City, CA, pp. 387-391, June 7- 10, 1998.
- [22]. Vishwanadh Tirumalashetty, Hamid Mahmoodi, “Clock Gating and Negative Edge Triggering for Energy Recovery Clock”, ISCAS 2007, New Orleans, LA, pp. 1141-1144, 2007.



**International Journal of Advances in
Engineering and Management**

ISSN: 2395-5252



IJAEM

Volume: 02

Issue: 01

DOI: 10.35629/5252

www.ijaem.net

Email id: ijaem.paper@gmail.com