COMPARATIVE STUDY OF ARITHMETIC & LOGIC UNITS ON FPGAs

Harshit Shrivastava, Department of Electronics & Communication Engineering, Sagar Institute of Research & Technology, Bhopal

Prof. Himanshu Nautiyal, Department of Electronics & Communication Engineering, Sagar Institute of Research & Technology, Bhopal

Prof. Sangeeta Shukla, Department of Electronics & Communication Engineering, Sagar Institute of Research & Technology, Bhopal

Abstract: *This paper deals with low power ALU design and its implementation Spartan 3 FPGA. Most of power is consumed in ALU in any processor and hence reduction in ALU power is needed. In this work, two ALUs are designed; the first design is conventional with all the logic blocks running all the time, in second design only those blocks are active which are used by currently selected operation, rest all blocks are inactive. This reduces the dynamic power consumption of the design.*

Keywords: *ALU, Low Power, Dynamic Power, Blocked I/O, FPGA.*

I. INTRODUCTION

This is an era of hand held devices and equipments, most of these devices run on battery, this puts a constraint on standby time, to increase standby time more and more battery life is needed, one way of solving this issue is to reduce power consumption of device or equipment. These days almost every device is intelligent, this intelligence came from using processors, and in forthcoming years this trend is likely to be increase. But these processors consume lot of the power of device as lot of switching activity is going inside. ALU (Arithmetic and Logic Unit) is the heart of any processor; this also consumes most of the processor power. In this work we worked in order to reduce power consumption of ALU. We have designed a sixteen bit optimized ALU, the size of the ALU can be easily increased to 32, 64 or more bits. In this work the prime focus is to reduce power consumption of the design. To achieve this we have used blocked I/O method.

II. DESIGN OF CONVENTIONAL ARITHMETIC AND LOGIC UNIT

The inputs to ALU are “A”, “B” (operands), “Clk” (clock), “sel” (to select one operation out of three operations). Outputs from ALU are “Z” (result). The steps in designing ALU are discussed as follows.

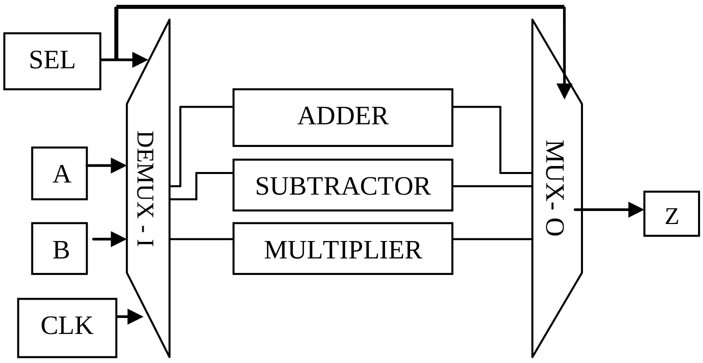


Fig. 2.1 Conventional ALU – High Level Block Diagram

Table 1. ALU operations

Sel	Operation
00	Addition (A+B)
01	Subtraction (A-B)
10	Multiplicaion (A x B)

In this design of conventional ALU, three operations can be performed namely addition, subtraction and multiplication. To implement these three units inbuilt adder, subtractor and multiplier of FPGA are used. Input demultiplexer “DEMUX - I” is used to route input data to the three units. All the units are running all the time but the output of selected unit depending upon the “SEL” value is passed to the output register “Z” via output multiplexer “MUX - O”.

At any point of operation only one of the operations can be performed, but all the three blocks get input data and are in operation. In this process all the three units are in operation all the time and this consumes power.

III. DESIGN OF LOW POWER ARITHMETIC AND LOGIC UNIT

In this section the design of low power ALU is discussed. Figure 3.1 depicts the high level block diagram of low power ALU.

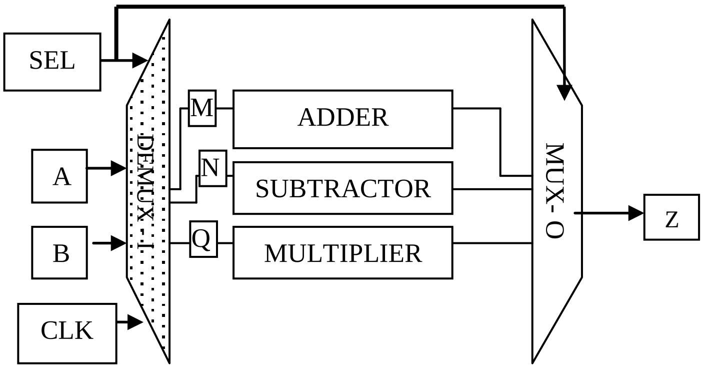


Fig. 3.1 Conventional ALU – High Level Block Diagram

To reduce the power consumption the "DEMUX - I" block is redesigned. Here "A" and "B" inputs are assigned only to currently selected block and rest two blocks are driven into tri-state. The output of tri-stated input does not change and the load capacitor does not change its state, this reduces dynamic power consumption. Table 2 shows the truth table of DEMUX – I unit.

Table 2. DEMUX – O Operation

"SEL"	**00**	**01**	**10**
Demux outputs			
M	A, B	high Impedance	high Impedance
N	high Impedance	A, B	high Impedance
Q	high Impedance	high Impedance	A, B

M, N and Q are internal bus connected to the inputs of adder, subtractor and multiplier respectively. When the "SEL" input is at "00" state then only adder block is assigned with the input "A", "B" and rest two blocks are driven into tri-state. When the "SEL" input is at "01" state then only subtractor block is assigned with the input "A", "B" and rest two blocks are driven into tri-state. And similarly when the "SEL" input is at "10" state then only multiplier block is assigned with the input "A", "B" and rest two blocks are driven into tri-state. The logic diagram of "DEMUX - I" is shown in figure 3.2. The "SEL" operation selection input is of two bits. It is abbreviated as s1 and s0 respectively in logic diagram of "DEMUX – I" shown in figure 3.2. s1 and s0 are connected to tri-state buffer. Tri-state buffer has two inputs and one output. Truth table of tri-state buffer is shown in table 3. When a particular operation is selected the control enable "CE" input of that tri-state buffer is 1 and the output of that channel is same as input, the control input of rest two channels is zero and output of those two channels are tri-state, and corresponding blocks are driven into high impedance.

Table 3. Truth Table – Tri-State Buffer

CE	Output
0	Z (high Impedance)
1	Input

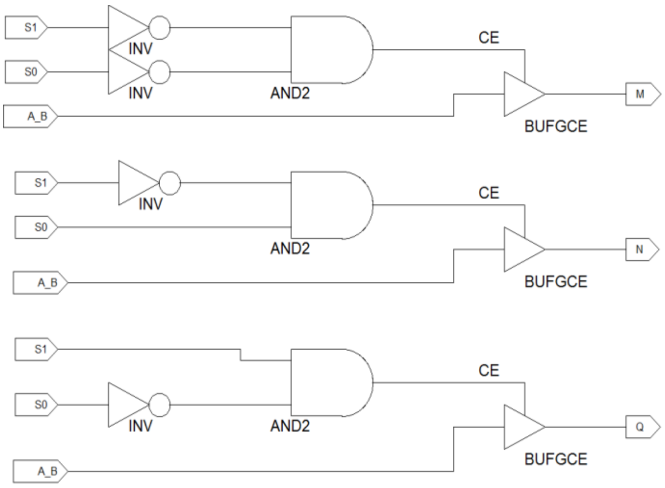


Fig. 3.2: DEMUX – I – Logic Diagram

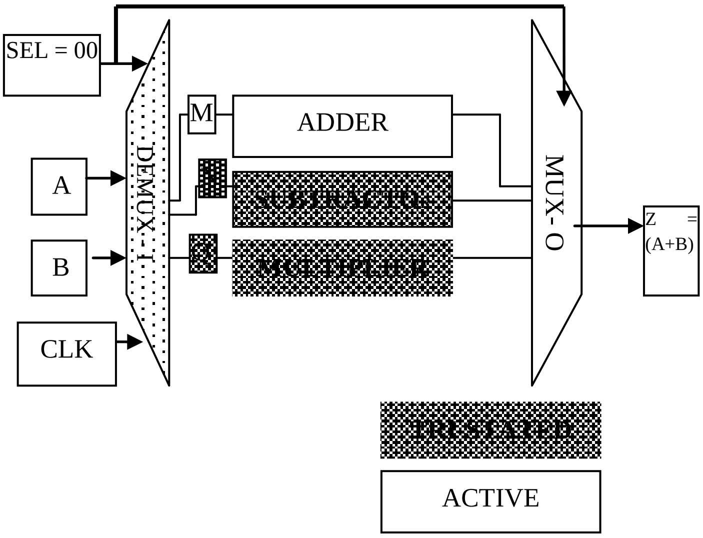


Fig. 3.3: Operation of ALU in addition

Figure 3.3 shows the working of ALU in addition operation. In this case the "SEL" input is 00 and hence addition operation is selected. So channel "M" is active and channel "N" and channel “Q” are tri-stated and corresponding blocks subtractor and multiplier respectively are also driven into high impedance. During this only adder is the active block, subtractor and multiplier are inactive and do not contribute to switching power (dynamic power). And hence dynamic power consumption reduces. The output multiplexer “MUX - O” selects the output of adder block and puts it onto output register Z.

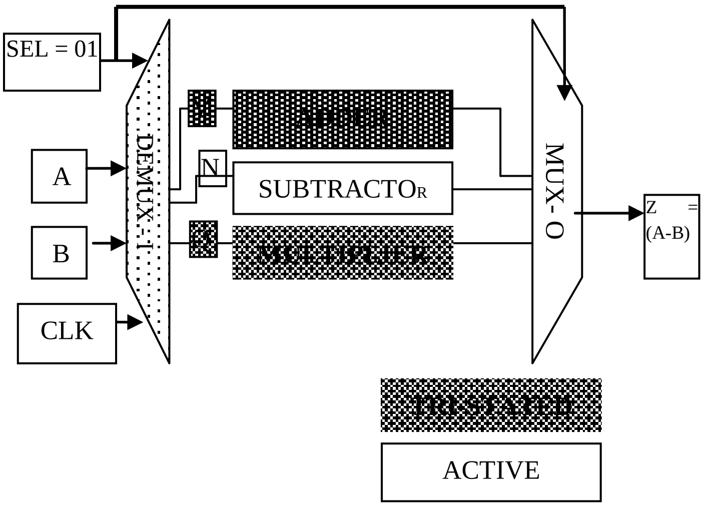


Fig. 3.4: Operation of ALU in Subtraction

Figure 3.4 shows the working of ALU in Subtraction operation. In this case the “SEL” input is 01 and hence Subtraction operation is selected. So channel “N” is active and channel “M” and channel “Q” are tri-stated and corresponding blocks adder and multiplier respectively are also driven into high impedance. During this only subtractor is the active block, adder and multiplier are inactive and do not contribute to switching power (dynamic power). And hence dynamic power consumption reduces. The output multiplexer “MUX - O” selects the output of subtractor block and puts it onto output register Z.

Figure 3.5 shows the working of ALU in multiplication operation. In this case the “SEL” input is 10 and hence multiplication operation is selected. So channel “Q” is active and channel “M” and channel “N” are tri-stated and corresponding blocks adder and subtractor respectively are also driven into high impedance. During this only multiplier is the active block, adder and subtractor are inactive and do not contribute to switching power (dynamic power). And hence dynamic power consumption reduces. The output multiplexer “MUX - O” selects the output of multiplier block and puts it onto output register Z.

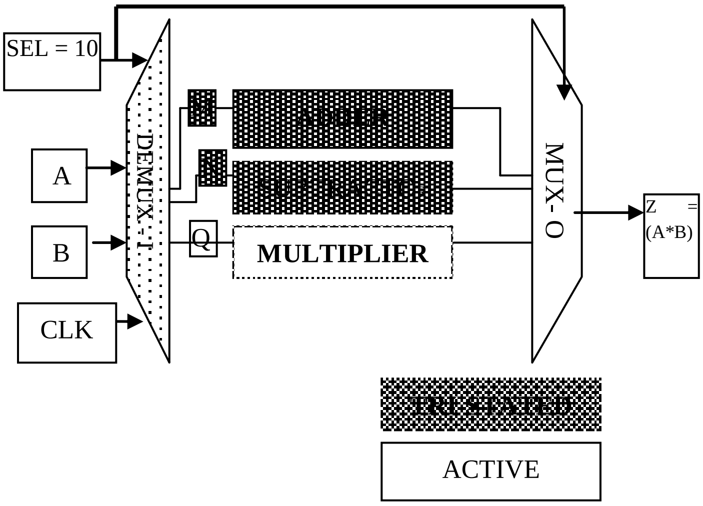


Fig. 3.5: Operation of ALU in multiplication

IV. SIMULATION AND RESULTS

The design is implemented on Spartan 3 FPGA, using xilinx 14.1i. It is tested using Xilinx ISIM and power is analyzed using Xpower analyzer.

Figure 4.1 shows the simulation of design II low power ALU in addition operation. The inputs “a” and “b” are assigned 4 and 3 respectively the “sel” input is 0 so the selected operation is addition. The output “zo” is 7. The inputs to unit sub and mul (“a_sub”, “b_sub” and “a_mul”, “b_mul”) are tri-stated and therefore the output of these blocks “z_sub” and “z_mul” are assigned to X(undefined). This reduces the power consumption.

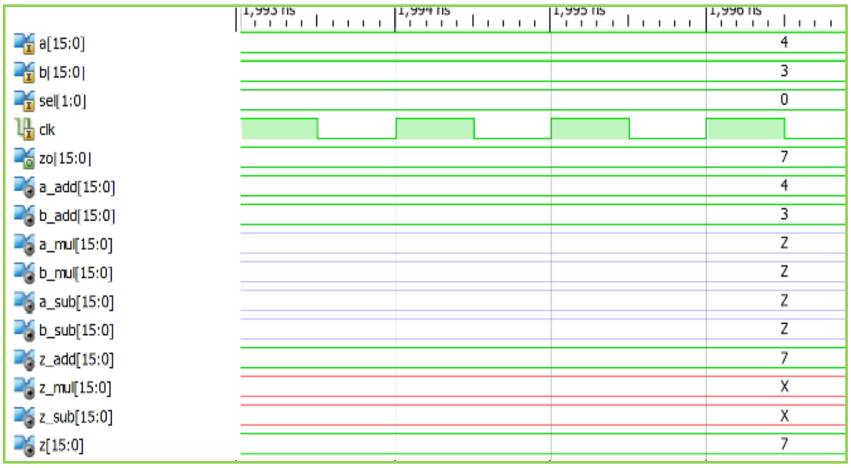


Figure 4.1: Simulation ALU - Addition

Figure 4.2 shows the simulation of design II low power ALU in subtraction operation. The inputs “a” and “b” are assigned 4 and 3 respectively the “sel” input is 1 so the selected operation is subtraction. The output “zo” is 1. The inputs to unit add and mul (“a_add”, "b_add" and "a_mul", "b_mul") are tri-stated and therefore the output of these blocks "z_add" and "z_mul" are assigned to X(undefined). This reduces the power consumption.

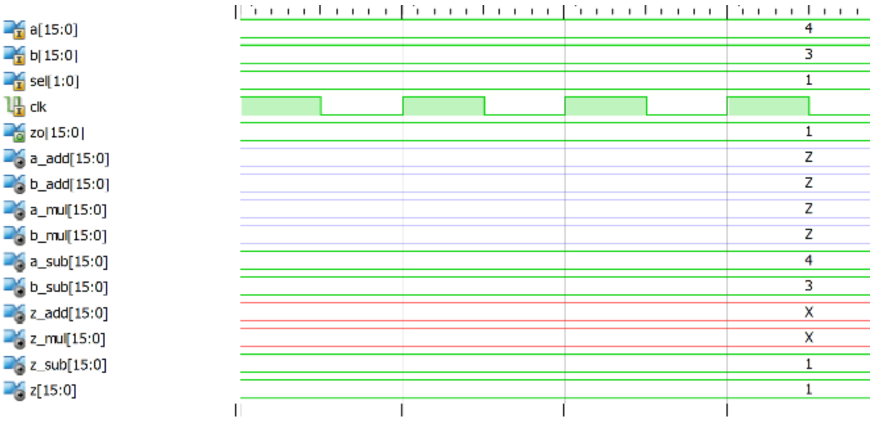


Figure 4.2: Simulation ALU - Subtraction

Figure 4.3 shows the simulation of design II low power ALU in multiplication operation. The inputs "a" and "b" are assigned 4 and 3 respectively the "sel" input is 2 so the selected operation is multiplication. The output "zo" is 12. The inputs to unit add and sub ("a_add", "b_add" and "a_sub", "b_sub") are tri-stated and therefore the output of these blocks "z_add" and "z_sub" are assigned to X(undefined). This reduces the power consumption.

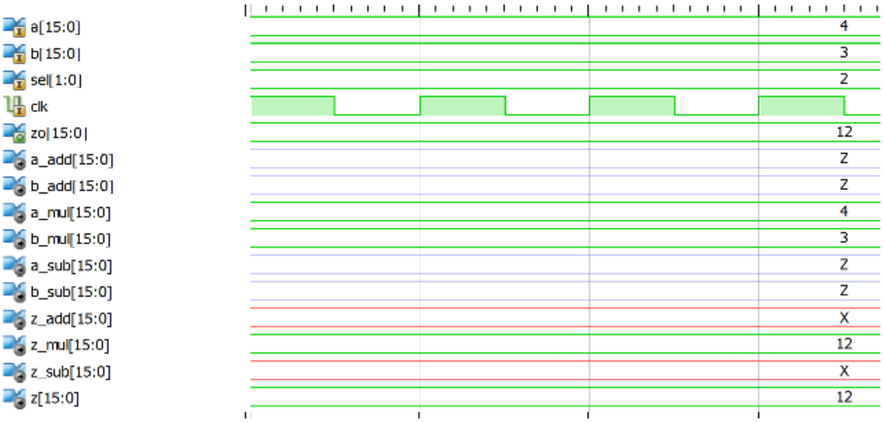
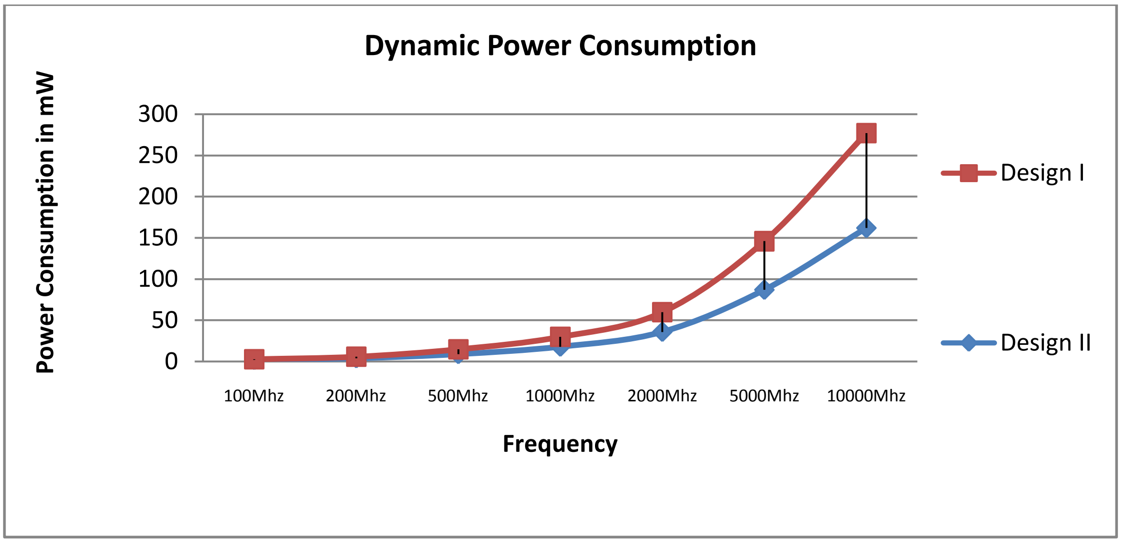


Figure 4.3: Simulation ALU – Multiplication

Table 3 depicts the power consumption summary of the two designs. Design I is the conventional ALU where all the units are active all the time. Design II is the low power ALU where blocks which are selected in current operation are active and rest units are tri-stated.

Table 3. Dynamic Power Consumption - Summary

S.no.	Frequenc y	Design I Conv. ALU	Design II Low Power ALU	% Change
1	100Mhz	2mW	1mW	-50%
2	200Mhz	4mW	2mW	-50%
3	500Mhz	9mW	6mW	-33%
4	1Ghz	18mW	12mW	-33%
5	2Ghz	36mW	24mW	-33%
6	5Ghz	87mW	59mW	-32%
7	10Ghz	162mW	115mW	-29%



Graph 4.1: Dynamic Power Consumption

As depicted in table 3 and graph 4.1 the dynamic power consumption of low power ALU design II is reduced by 30% to 50% compared to the conventional ALU design I.

V. CONCLUSION

ALU is the core of processor, and optimizing ALU can significantly improve the performance of processor. In this work we worked in order to reduce power consumption and resource utilization of FPGA. As we can conclude from power report that by disabling the inactive blocks, dynamic power consumption can be significantly reduced; this is because of decrease in switching activities inside ALU.

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] J. Shinde, and S. S. Salankar, “Clock gating-A power optimizing technique for VLSI circuits” *Annual IEEE India Conference (INDICON)*, pp. 1-4, 2011.

[3] J. Castro, P. Parra, and A. J. Acosta, “Optimization of clock-gating structures for low-leakage high-performance applications”, Proceedings of IEEE International Symposium on Efficient Embedded Computing, pp. 3220-3223, 2010.

[4] Frank Emnett, Mark Biegel, Power Reduction Through RTL Clock Gating, SNUG San Jose, 2000.

[5] Gary K. Yeap, Practical Low-Power Digital VLSI Design, Power , EE Times India, January 2008.

[6] John F. Wakerly, Digital Design Principles and Practices, Prentice Hall, 2005.

[7] Hubert Kaeslin, ETH Zurich, Digital Integrated Circuit Design from VLSI Architectures to CMOS Fabrication, Cambridge University Press, 2008.

[8] 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.

[9] 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, Oct. 1999.

[10] Safeen Huda, Muntasir Mallick, Jason H. Anderson, Clock Gating Architectures For FPGA Power Reduction , FPL 2009.

[11] Vojin G. Oklobdzjja, Vladlmlr M. Stojanovic, Dejan M. Markovic, Nikola M. Nedovic, DIGITA L SYSTEM CLOCKING High-Performance and Low-Power Aspects, Wiley Interscience, U.S., 2003.

[12] Vishwanadh Tirumalashetty, Hamid Mahmoodi, Clock Gating and Negative Edge Triggering for Energy Recovery Clock, ISCAS 2007, New Orleans, LA, pp. 1141-1144, 2007.

[13] Bishwajeet Pandey, Jyotsana Yadav, M Pattanaik, Nitish Rajoria “Clock Gating Based Energy Efficient ALU Design and Implementation on FPGA” 2013 IEEE.