INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Design of power efficient 8 bit arithmetic and logic unit on FPGA using tri-state logic

Siddharth Singh Parihar[1], Rajani Gupta[2]

[1] Kailash Narayan Patidar College of Science and Technology,
Baghmugaliya, Bhopal 462001, India
Siddharth.siddy.singh@gmail.com

[2]Kailash Narayan Patidar College of Science and Technology,
Baghmugaliya, Bhopal 462001, India
rajni_gupta2007@yahoo.com

Abstract*::This paper deals with design of power efficient 8 bit ALU and its implementation on 90nm Spartan 3, device xc3s50-5pq208 FPGA chip. Most of power is consumed in ALU in any processor and hence reduction in ALU power must required. To reduce power consumption we disabled all the blocks using tri-state logic which is not in used, except selected operation. This will cut down the FPGA resource usage and also reduce the power consumption. In this paper theoretically we analyze 93.75% power reduction. By using tri-state concept dynamic were reduced and less FPGA resources were consumed.*

Keywords: *ALU, FPGA, tri-state logic, dynamic power consumption*

1. INTRODUCTION

In modern days, power consumption is important term for chip designers. Generally, any circuit consumes two types of powers, first static power consumption and second is dynamic power consumption [1].Static power consumption due to leakage current while dynamic power consumption due to internal switching of the capacitance [1]. Clock power consumes most of the power in ALU. There are many techniques to reduce power consumption. In this work we reduce dynamic power and also FPGA resource usage. We have designed an 8 bit power optimized ALU and dynamic power consumption were reduced by the help of tri-state logic. The size of an ALU can be easily modified by 16 bit, 32 bit, 64 bit. This is due to the fact that power is directly Proportional to voltage and the frequency of the clock as shown in the following equation:

Power = capacitance × voltage × frequency (1)

Section II of the paper deals with the implementation of an ALU. Section III deals with the results which contain simulation result and power analysis. Section IV contains concluding remarks of this work. Section V contains future extension of research work.

2. IMPLEMENTATION

2.1 Functions of an ALU

Basically ALU performs 16 different types of functions refer to table 2.1.

There are first eight operations known as unary operations. Next four shows arithmetic function and last four functions are logical operations, performed by ALU.

2.3 Tri-state logic

Here, we designed low power 8 bit ALU, with the help of tri-state logic we can significantly reduce the consumption of dynamic power.

Table2.1. Function of an ALU

FUNCTIONS OF ALU				
S_3	S_2	S_1	S_0	FUNCTION
0	0	0	0	Clear
0	0	0	1	Hold B
0	0	1	0	Complement B
0	0	1	1	Hold A
0	1	0	0	Complement A
0	1	0	1	Decrement A
0	1	1	0	Increment A
0	1	1	1	Shift left A
1	0	0	0	Add
1	0	0	1	Subtract
1	0	1	0	Add with carry
1	0	1	1	Subtract with borrow
1	1	0	0	Logical AND
1	1	0	1	Logical OR
1	1	1	0	Logical XOR
1	1	1	1	Logical XNOR

Generally, Tri-state buffer is act as a switch. It is controlled by control signal, known as 'enable signal', refer figure 2.1. Here if control input i.e. En signal goes to '0', then it shows high impedance state Z. So it act as open circuit and input would disconnected to output.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

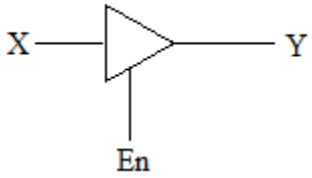

Figure 2.1: Tri-state buffer

In tri-state logic if control input En=1, then input is appeared at the output of tri-state buffer i.e. X=Y, that means it just pass the value of input, at the output. But if control input En=0, then it act as an open circuited and input is disconnected from the output i.e. input wouldn't appear at the output, refer figure 2.2.

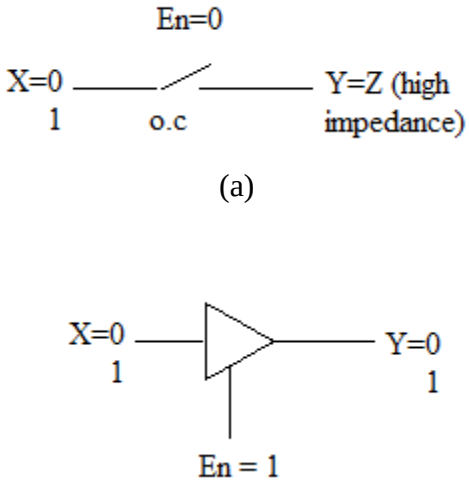

Figure 2.2: (a) when En=0 (b) when En=1

We used this property of tri-state logic, to reduce the power consumption of an ALU. We know ALU perform sixteen different operations. To perform 16 operation different 16 modules are implemented. At one particular time one operation perform, while all other fifteen operations (which are not select), consumes clock power at the same time. This will leads dynamic power consumption. In this work, we overcome the problem with the use of tri-state buffer. Here at a time one operation can perform by ALU, while all other fifteen operations are in tri-state i.e. high impedance Z state.

2.4 Design of demux using tri-state logic:

We can implement multiplexer circuit with the help of tri-state logic. So that ALU perform only one operation at a time, which is being selected by the selection input. While remaining 15 operations are in high impedance Z state. That means these fifteen operations are disconnected to output.

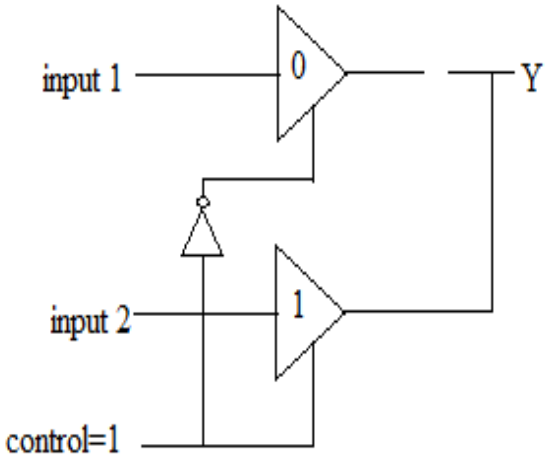

Figure 2.3: 2x1MUX using tri-state logic

In the figure 2.3, consider 2x1Mux implemented with tri-state logic which has two inputs and one output. If control input =1, then input 2 will be appear at the output i.e. Y=input 2. But input 1 doesn't appear at the output because an inverter makes control input 0 to input 1. So it act as open circuited path. similarly, we can implement 16x1MUX for 16 operation of an ALU. This will ensure that of all 16 operations, only one operation are selected at a time and performed by ALU, while rest of operations are in tri-state.

2.4.1 Clear function:

The selection input given to demux for Clear operation is '0000', as shown in table 2.1. When tri-state scheme are not used then all 16 operations are connected to clock blocks, refer figure 2.4. This will leads power consumption.

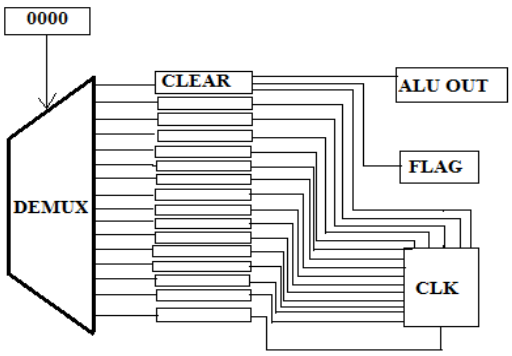

Figure 2.4: Clear function with sel '0000'

When we use tri-state scheme, then only 'Clear' operation will perform. refer figure 2.5, here all blocks are in high impedance state except Clear '0000' functions. Since inactive block are disconnected to clock hence, reducing in internal switching activities. So dynamic power consumption of an ALU will also reduce.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

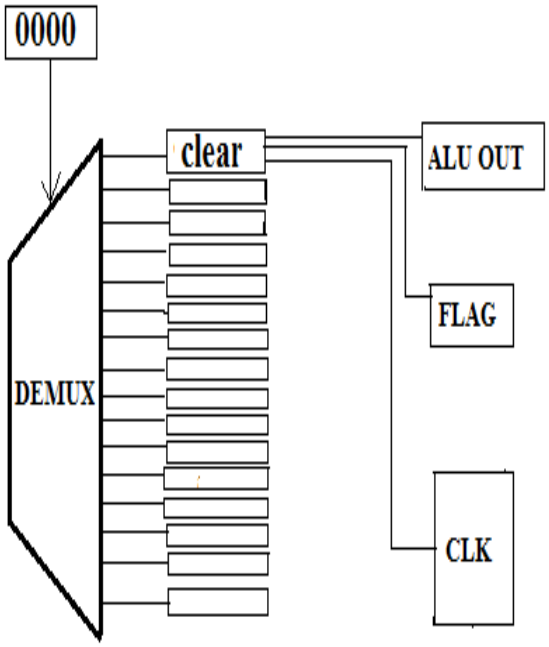

Figure 2.5: Clear function with tri-state logic

Selection line is used to control the function of ALU. There are different selections inputs are assigned for different ALU's function.

2.5 Top level schematic

In the schematic of ALU, there are four inputs. A(7:0), B(7:0), sel(3:0) and Clock. A and B are called operands.

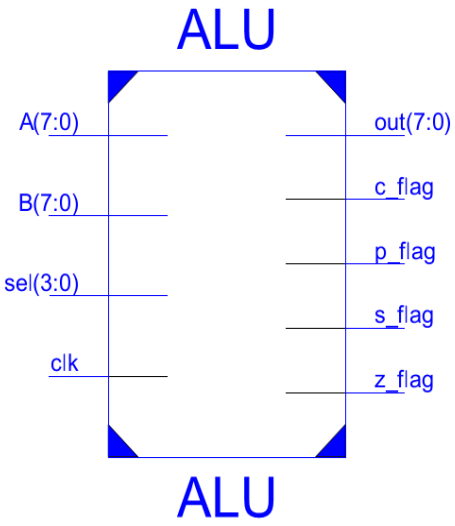

Figure 2.6: Schematic of an ALU

Input Sel(3:0) determines, which operation to be performed. Also there are four flags generated. Carry C, parity P, sign S, zero Z. All flags are affected by the other ALU functions.

2.6 RTL schematic

RTL view of ALU shown in figure below:

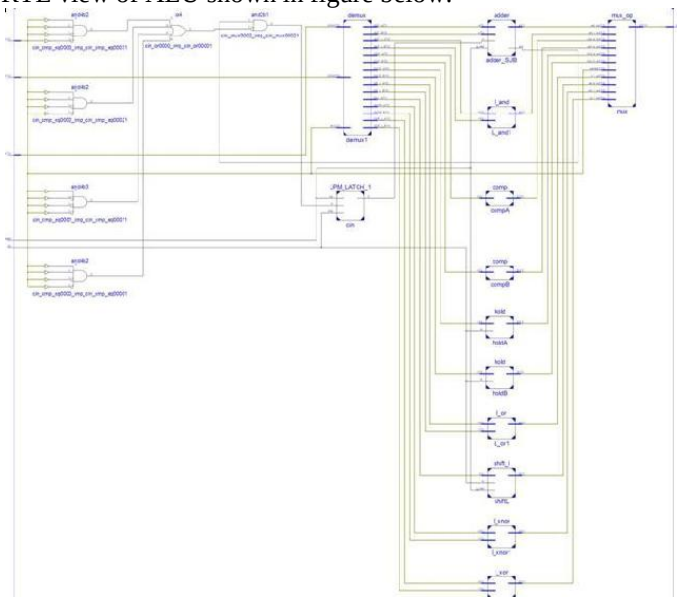

Figure 2.7: RTL schematic view of ALU

3. RESULTS

We have used Xilinx 14.1i to implement the design on 90nm Spartan 3 FPGA xc3s50-5pq208. Xpower analyzer is used to perform power analysis. ISIM is used to perform behavioral simulation.

3.1 Behavioral Simulation

We have Xilinx ISIM for behavioral simulation; we tested our design with different values of input and in all conditions our ALU is behaving as per the specifications. Figure3.1 shows the behavioral simulation of one of the operation i.e. Clear function. The input sel(3:0) of Clear function is '0000'.

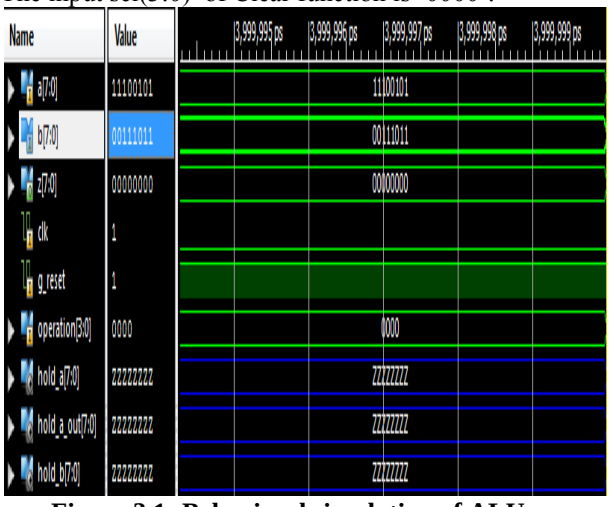

Figure 3.1: Behavioral simulation of ALU

Behavioral simulation of de-multiplexer is depicted in Figure 3.2.

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY
WINGS TO YOUR THOUGHTS.....

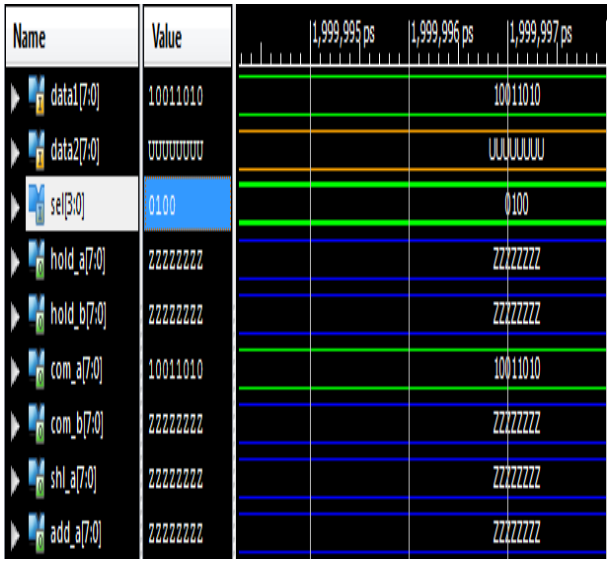

Figure 3.2: Behavioral simulation of demux

3.2 Power report

We used Xpower analyzer tool from Xilinx to calculate power consumption of device. The design is operated at different frequencies and power consumption is noted accordingly.

Table3.1.power report at different frequencies

Logic Utilization	Used
Number of Slice Flip Flops	27
Number of 4 input LUT"s	165
Number of occupied Slices	113
Number of bonded IO	30

3.3 Resource utilization

Table 3.2 shows resource utilization summary. In this work, we used same adder block for six operations. Because binary subtraction perform by adder circuit with the help of 2's complement representation. So addition, subtraction, add with carry, subtract with borrow, increment and decrement function can perform by single adder circuitry. This will leads to less FPGA resource usage.

4. 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. Next improvement we tried to implement is reduction in FPGA resource usage; 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 technique. This fulfills our two purposes; first, reduction of FPGA resource usage and second reduction in power consumption.

Table 3.2 Resource utilization Summary

Frequency	Clock (mW)	Logic (mW)	Signals (mW)	IO (mW)	Dynamic Power (mW)	Previous Design (mW)
100 Mhz	1	0	1	0	2	3
1 Ghz	8	5	8	2	23	28
10 Ghz	76	26	58	24	184	NA
100 Ghz	761	31	384	243	1418	1725

5. FUTURE SCOPES

In this work we implemented our design on 90nm Spartan 3 FPGA. One possibility of improvement is designing it on 28nm FPGA. Next possibility is optimizing the code; this will improve the performance of ALU. The size of the ALU can also be increased to 16, 32 or 64 bit. Our amendments will show a better performance on larger size.

REFRENCES

[1] Liril george1 and padmaja bangde2, design and implementation of low power consumption 32 bit alu using fpga, international journal for research in emerging science and technology, volume-1, issue-5, october-2014.

[2] 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.

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

[4] 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

INTERNATIONAL JOURNAL FOR ADVANCE RESEARCH IN ENGINEERING AND TECHNOLOGY

WINGS TO YOUR THOUGHTS.....

Technologies (ICCEET), pp.772-776, 2012.

[5] S. Cisneros, J. J. Panduro, J. Muro, and E. Boemo, "Rapid prototyping of a self-timed ALU with FPGAs", International Conference on Reconfigurable Computing and FPGAs, pp. 26-33, 2012.

[6] B. S. Ryu, J. S. Yi, K. Y. Lee and T. W. Cho,"A design of low power 16-bit ALU", Proceedings of the IEEE TENCON Conference, pp.868-871, 1999.

[7] T. Lam, X. Yang, W. C. Tang and Y. L. Wu; , "On applying erroneous clock gating conditions to further cut down power," Design Automation Conference (ASP-DAC), 2011 16th Asia and South Pacific , vol., no., pp.509-514, 25-28 Jan. 2011.

[8] B. Pandey and M. Pattanaik, "Clock Gating Aware Low Power ALU Design and Implementation on FPGA", 2nd International Conference on Network and Computer Science (ICNCS), Singapore, April 1-2, 2013

[9] E. Arbel, C. Eisner and O. Rokhlenko, "Resurrecting infeasible clockgating functions," Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE , vol., no., pp.160-165, 26-31 July 2009

[10] J. Monteiro, J. Rinder knecht, S. Devadas and A. Ghosh, "Optimization of combinational and sequential logic circuits for low power using precomputation," Advanced Research in VLSI, 1995. Proceedings.

[11] Bishwajeet Pandey, Jyotsana Yadav, M Pattanaik," Clock Gating Based Energy Efficient ALU Design and Implementaion on FPGA" Annual IEEE India conference 978-1-4673-6150-7/13/$31.00 ©2013 IEEE.