Open access Journal International Journal of Emerging Trends in Science and Technology
Impact Factor: 2.838 DOI: http://dx.doi.org/10.18535/ijetst/v3i02.08

A Review Paper on Arithmetic and Logical Unit for Graphics Processor

Authors
Mamta Ratre[1], Jitendra Singh[2]
[1]Department of Electronics and Telecommunication,Takshshila Institute of Engg. & Technology Jabalpur, M.P, India
Email: *ratre.mamta@gmail.com*
[2]Department of Electronics and Telecommunication,Takshshila Institute of Engg. & Technology Jabalpur, M.P,India
Email: *jitendrasingh@takshshila.org*

Abstract
In this era of high end devices, the power efficient architecture is taking responsibility so as to reduce the cost for maintenance. This becomes the critical task for embedded based device, graphical based processor and similar DSP processors which suffers with low power. The core of every embedded device and processor which in turn uses ALU as the workhorse. As we know if workhorse require less power, speed and area so based on that workhorse complete system will make justice with SPAA metrics (Speed, Power, Area and Accuracy). This paper present a review study on different type of exsisting ALU design. Here we also implement that different ALU logic by using of Verilog HDL on Xilinx tool. The synthesized architecture is implemented by Hardware descriptive language (Verilog). Analysis is performing on FPGA (Field Programmable Gate Array) level.

1.INTRODUCTION

Due to wide spread use of microprocessors and signal processors, implementation of high performance arithmetic hardware has always remained an attractive design problem. Arithmetic and Logic Unit (ALU) is the workhorse of microprocessors and determines the speed of operation of the processor. All modern processors include stand alone hardware for computation of basic arithmetic operations. In addition to fast arithmetic hardware, processors are also equipped with on-chip memory (cache) to achieve significant performance improvement by avoiding delay due to data access from main memory.

The Arithmetic Logic Unit is essentially the heart of a CPU. It has more applications in DSP and micro processors. In the past, VLSI designers concentrated more on area, performance, cost and reliability. The least importance was given to power. Now a day's power is given primary importance than area and speed. The two low power logic styles used in ALU are CMOS logic and PTL logic. In present era every portable devices are battery operated and as we know ALU is the brain of the whole system. If brain require heavy power, are and latency so the complete system is require heavy area, power, and latency. The fundamental and most commonly used arithmetic operation in many VLSI systems such as DSP architectures, microprocessors, etc., is the most speed limiting element .Main task of this operation is to add two binary numbers, and it is implemented by a full adder cell. Furthermore, an addition as an operation is used as a basis in many other useful and more complex operations, e.g., multiplication, division, and address calculation. All these operations are realized by complex systems of transistors.

Typically, the ALU has direct input and output access to the processor controller, main memory (random access memory or RAM in a personal

computer), and input/output devices. Inputs and outputs flow along an electronic path that is called a bus. The input consists of an instruction word (sometimes called a machine instruction word) that contains an operation code (sometimes called an "op code"), one or more operands, and sometimes a format code. The operation code tells the ALU what operation to perform and the operands are used in the operation. (For example, two operands might be added together or compared logically.) The format may be combined with the op code and tells, for example, whether this is a fixed-point or a floating-point instruction. The output consists of a result that is placed in a storage register and settings that indicate whether the operation was performed successfully.

The arithmetic logic unit (ALU) is the brain of the computer, the device that performs the arithmetic operations like addition and subtraction or logical operations like AND and OR. This section constructs an ALU from four hardware building blocks (AND and OR gates, inverters, and multiplexors) and illustrates how combinational logic works. In the next section, we will see how addition can be sped up through more clever designs. Because the MIPS word is 32 bits wide, we need a 32-bit-wide ALU. Approximation is a area which will help to make justification with SPAA (Speed, Power, area and accuracy) metrics. In present era energy aware system is most important due to battery operated systems. Using approximation concept we can decrease some amount of accuracy which is tolerable by human eye.

2. Literature Review

In present era computer engineers have never stopped trying to improve system performance by optimizing arithmetic units. In 1951, researcher Booth presents a signed binary recoding scheme [1], this scheme is used to reduce the number of partial products from the multiplier, after some time booth approach is improved by another researcher whose known as Wallace in [2]. Besides integer operations, floating-point arithmetic is also a hot topic for many researchers [3,4-8]. With the emergence of reconfigurable computing, engineers started to look for practical solutions for multiple-precision computations. Constant in ides, and his colleagues had broad explorations [9-11] of bit width assignment and optimization for static multiple precision applications. Wang and Leeser spent years to develop and refine a complete statically-defined, variable-precision fixed- and floating-point ALU library for reconfigurable hardware [12].Since re-synthesizing, re-downloading, and reconfiguring are required for static multiple-precision ALUs whenever precision is changed, this solution is not practical for applications that require frequently changing precision. Thus, multi-mode ALUs become more attractive.

In [13], Tan proposed a 64-bit multiply accumulator (MAC) that can compute one 64x64, two32x32, four 16x16, or eight 8x8 unsigned/signed multiply-accumulations using shared segmentation. On the other hands, Akkas presented architectures for dual mode adders and multipliers in floating-point [14, 15], and Is seven presented a dual-mode floating-point divider [16] that supports two parallel double-precision divisions or one quadruple-precision division. In [17], Huang present a three-mode (32-, 64- and 128-bitmode) floating-point fused multiply-add (FMA) unit with SIMD support. It is clear that all the above multimode multiple-precision structures can only support a few pre-defined precisions. To the best of our knowledge, our proposed architecture is the first true dynamic precision architecture targeting both fixed point and floating-point ALUs. Now a the days in current research there is also a very use full technique is VEDIC mathematics using this logic there is improvement in timing complexity, area power [18].

There is some more latest research on arithmetic logical unit, in [19] author presents a ALU which contains two sub modules lower bound module and upper bound module. This design perform addition, substraction, multiplication and set operation of union, division is performed by shifting Lower and upper bound module is selected by flag generation. Drawback of this approach is Hardware size and

power consumption is increases for division only shifting property is used.

In [20] author proposed a ALU unit for 8 bit microcontroller according to that approach proposed ALU contains three sub modules, ARITHMETIC, LOGIC, and BIT operation. This design performs arithmetic, logical and shifting or rotation operation. This design performs fixed and floating point operation. All sub module are connected with mux by controlling signal mux will select output. Problem with this approach is Cascade connection of Full adder creates a major problem and it will increase latency. In this design arithmetic unit based on cascade connection. Similar in logical unit shifter are connected in cascade form. Limited instruction set only 15 operations are performed.

In [21] author proposed a approach which have two type of ALU structure first one is tree and second one is chain. In tree approach require less area as compare to other design. In chain approach latency of design is fast and controlling on operation is simple. The approach can be easily integrated into a processor design environment. Problem with this approach is there is no any control signal, when any two inputs A & B are generated so that input is computed by every computation unit. Require extra hardware for Chain Structure. Input, Output Pins are increases in chain structure. Tree structure increase latency according to operation.

In [22] author proposed a Bit-width aware for the control (ctrl) and carry-out (carry) signals for the design with multiple dynamic precision (DP) operation. Problem of this approach is that it require extra hardware unit for dynamic precision. There is some another logic is developed in present era which is known as Approximation [23,24] according to that approach there is many application which are error tolerant means the error which is not identify by human eyes. Similar for power reduction there is one useful module which is known as Clock gating [25] according to that it will reduce the dynamic power of the whole system.

So according to previous existing approaches there is lots of problems we can identifies, these problem are related to need of large hardware unit, need of large power which is not suitable for portable devices like mobile, laptop etc. The require speed is also a big challenge for those designs. All proposed architecture are facing a common problem which is input output pins and increment in dynamic power.

In this paper we are basically try to resolve those existing problems using of approximation concept. These are the key points which we are targeting on this paper:

- Devolve architecture which is reduce the timing complexity issues.
- Devolve architecture which is reduce the area and worked on low power.
- Here we are focusing on approximation concept, where we divide 32 bit ALU in four 8 Bit Alu.

The rest of the paper is organized as follows. Section 3 describes research gap. Section 4 represent the future objective. Section 5 demonstrates Implementation details of previous existing design. Section 6 shows Experimental. Section 7 presents application of ALU design. Finally, Section 8 concludes the paper.

3. Research Gap:

There are followings research gaps in previous different architecture of Arithmetical and logical unit:

1. There is need of large hardware unit.
2. There is need of large power which is not suitable for portable devices like mobile, laptop etc.
3. The require speed is also a big challenge for those designs.
4. All these proposed architecture are facing a common problem which is input output pins and increment in dynamic power.

4. Future Objective

There is lots of future objective in this ALU architecture. There is lots of problems which is faced by previous approach. Future Objective of this Research area are followings:

1. To devolve architecture which is reduce the timing complexity issues.

2. To devolve architecture which is reduce the area related problems and which is suitable for all portable device.
3. To devolve architecture which is reduces power and consumption and work on low power.
4. To devolve architecture which is reduce energy consumption and which will make justice with all portable devices which are demands low energy consumption.
5. To devolve a architecture which use some approximation concept and also some power reduction approach.

5. Implementation Details:

Here we are presenting the implementation details of previous existing. Here we are using one hardware descriptive language which is known as Verilog. Using this language we were design existing and proposed approach on FPGA based designing tool and design verification is done on Modelsim. Here we are designing our complete design in 45nm technology based FPGA which is known as Vertix-6.

Here we are present a proposed architecture of our ALU. Which is basically having four different eight bit block. Which is known as Accurate, Semi accurate and approximate. Here all implementation is done on Xilinx 14.2 tool using verilog.

5.1 Implementation of 32 bit Accurate ALU Design:

Here we are designing existing accurate 32 bit alu logic with 16 different instructions set. Here we show Gate level schematic and logic based schematic of accurate ALU:

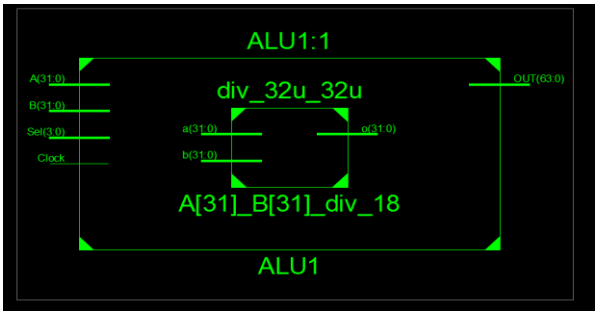

Fig. 1 ALU Top Design

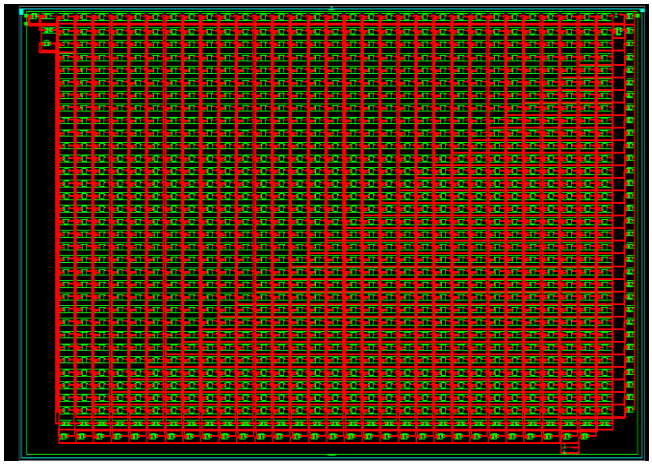

Fig. 2 ALU Schematic Level Design

5.2 Implementation of Pipe based ALU Architecture [19]:

Here we are designing existing pipe line based 32 bit alu logic with 16 different instructions set. Here basically ALU is divide in two different part first one is LSB based and Second one MSB based ALU. Here we show Gate level schematic and logic based schematic of accurate ALU:

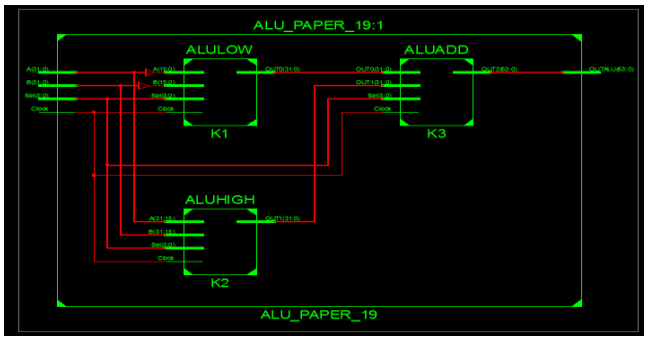

Fig. 3 ALU Pipe Based Top Module

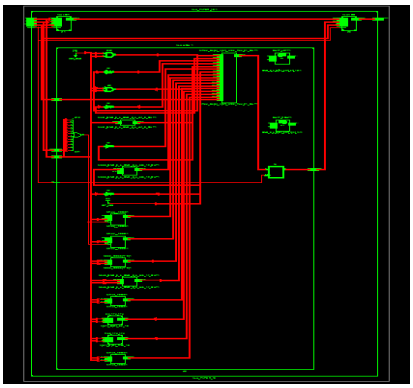

Fig. 4 ALU Pipe Based LUT Design

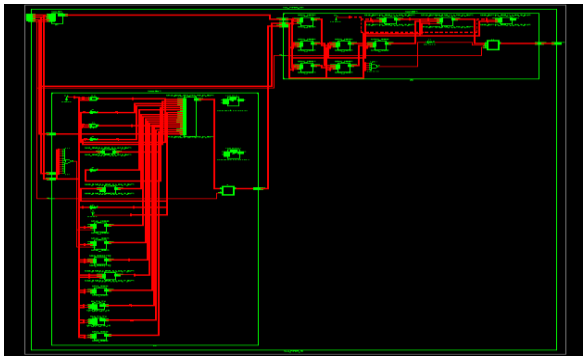

Fig. 5 ALU Pipe Based Schematic Design

5.3 Implementation of Vedic based ALU Architecture [18]:

Here we are designing existing vedic based 32 bit alu logic with 16 different instructions set. Here basically ALU is having one multiplier module which is based on Veidc concept of URDHAVA multiplier. Here we show Gate level schematic and logic based schematic of accurate ALU:

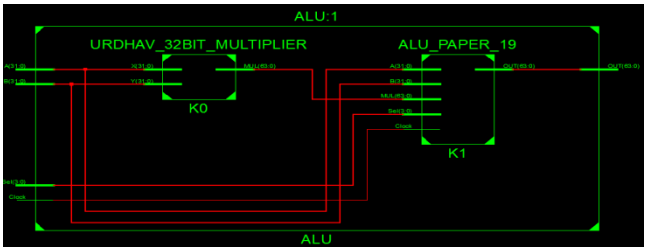

Fig. 6 ALU Vedic Based Top Level Design

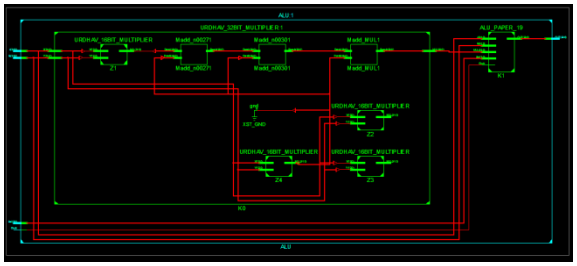

Fig. 7 ALU Vedic Based LUT Design

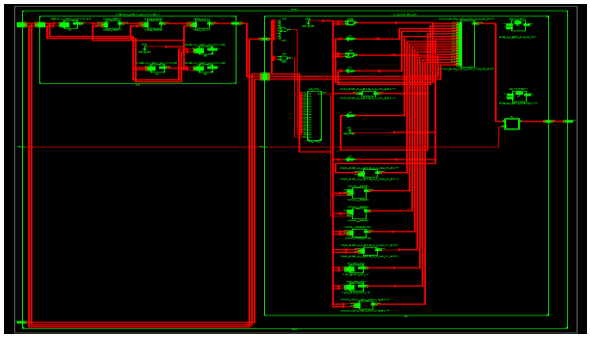

Fig. 8 ALU Vedic Based Schematic Design

5.4 Implementation of ALU Architecture [20]:

Here author proposed a alu architecture where he divides instruction set in three section LOGIC, Arithmetic, Bit Here we show Gate level schematic and logic based schematic of accurate ALU:

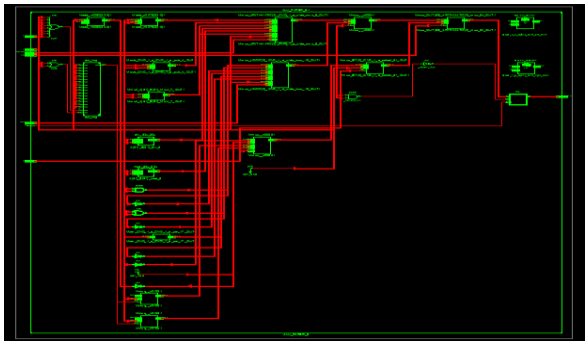

Fig. 9 ALU Based Schematic Design

6. Result & Analysis

In this section we are representing comparative analysis of proposed ALU architecture with existing ALU architecture in terms of power, area(LUT), delay and frequency. The FPGA comparison analysis of proposed and accurate are shown below, here hardware analysis is done on Vertix 6 FPA which is 45nm based technology.

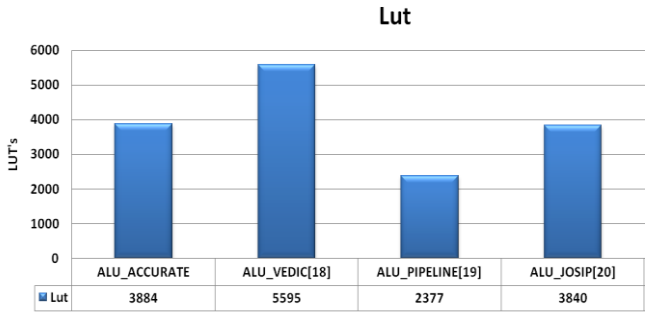

Fig. 10 LUT Based Comparative Analysis

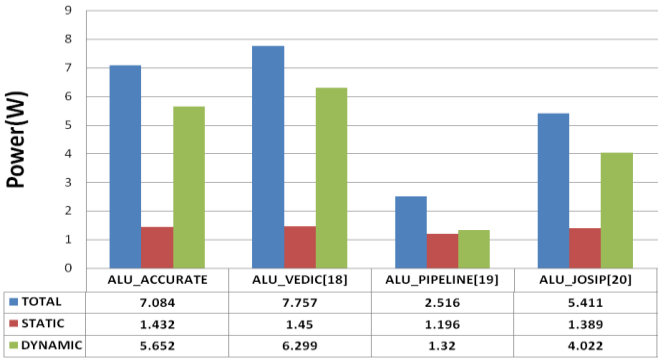

Fig. 11 Power Based Comparative Analysis

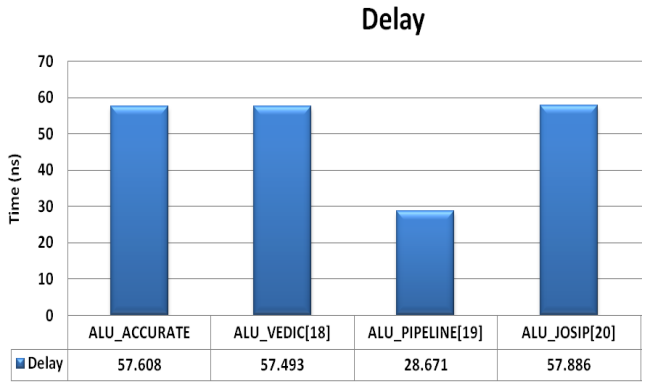

Fig. 12 Delay Based Comparative Analysis

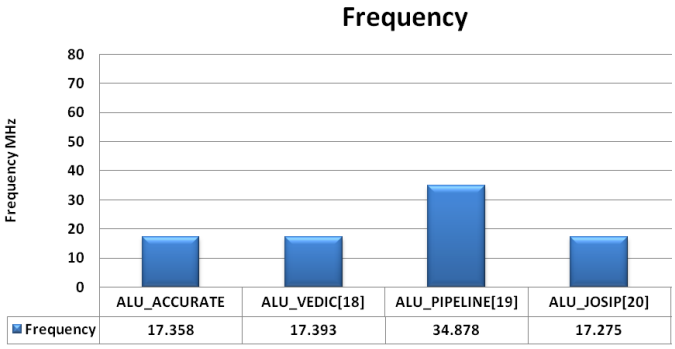

Fig. 13 Frequency Based Comparative Analysis

As we can see in Fig. 10,11,12,13 shows the comparative analysis about the previous existing ALU design in terms of area, power, delay and frequency.

7. Applications:

There are many applications of Arithmetic and Logical unit. These is used in many filed like multimedia, signal processing, general purpose. There is some applications which are demands ALU and those are:

1. MP3 Decoding/Encoding.
2. Video Decoding/Encoding.
3. Image Filtering.
4. Adaptive Difference Pulse Code Modulation.
5. Image and video processing.
6. General Purpose Processor.
7. Application Specific Processor.

So these are the different applications where we use Arithmetic logical unit.

8. Conclusion

In this paper we are presenting a comparative study about the ALU architecture. According to previous existing approach there is lots of issues with ALU architecture, which still require some improvement. As a research area this architecture has lots of future scope. Here we also design all previous existing ALU architecture and make a comparative analysis in terms of Area, power, Delay and frequency. All hardware implementation is done on Xilinx 14.2 and design simulation is done on modelsim. Here we are using 45nm technology based fpga which is known as Vertix-6.

References

1. A Booth, "A signed binary multiplication technique," The Quarterly Journal of Mechanics and Applied, vol. 4, no. 2, pp.236-240, 1951.
2. C. S. Wallace, "A suggestion for a fast multiplier," Electronic Computers, IEEE Transactions on, vol. EC–13, no. 1, pp. 14–17, 1964
3. K. S. Hemmert and K. D. Underwood, "Fast , Efficient Floating-Point Adders and Multipliers for FPGAs," Technology, vol. 3, no. 3, 2010.
4. S. Anderson and J. Earle, "The IBM system/360 model 91:floating-point execution unit," IBM Journal of, no. January,1967.
5. P. Seidel and G. Even, "Delay-optimized implementation of IEEE floating-point addition," IEEE Transactions on Computers, vol. 53, no. 2, pp. 97-113, Feb. 2004.
6. R. M. Jessani and M. Putrino, "Comparison of single- and dual-pass multiply-add fused floating-point units," IEEE Transactions on Computers, vol. 47, no. 9, pp. 927-937, 1998.
7. Division algorithms for an x86 microprocessor with a rectangular multiplier," in 2007 25th International Conference on Computer Design, 2007, pp. 304-310.
8. G. Even and P.-M. Seidel, "A comparison of three rounding algorithms for IEEE floating-point multiplication," IEEE Transactions on Computers, vol. 49, no. 7, pp. 638-650, Jul.2000.

9. G. A. Constantinides, P. Y. K. Cheung, and W. Luk,"Wordlength optimization for linear digital signal processing," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 22, no. 10, pp. 1432-1442, Oct. 2003.
10. G. a. Constantinides, P. Y. K. Cheung, and W. Luk,"Optimum and heuristic synthesis of multiple word-length architectures," IEEE Transactions on Very Large ScaleIntegration (VLSI) Systems, vol. 13, no. 1, pp. 39-57, Jan.2005.
11. D.-U. Lee, A. A. Gaffar, R. C. C. Cheung, O. Mencer, W.Luk, and G. A. Constantinides, "Accuracy-Guaranteed Bit-Width Optimization, " IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 25, no. 10, pp.1990-2000, Oct. 2006.
12. X. Wang, "VFloat: A Variable Precision Fixed- and Floating-Point Library for Reconfigurable Hardware," ACM Transactions on Reconfigurable Technology, vol. 3, no. 3, pp.1-34, 2010.
13. D. Tan, A. Danysh, and M. Liebelt, "Multiple-precision fixed point vector multiply-accumulator using shared segmentation," in 16th IEEE Symposium on Computer Arithmetic, 2003. Proceedings,2003, vol. 00, no C, pp 12-19.
14. A. Akkas, "Dual-mode floating-point adder architectures ,"Journal of Systems Architecture, vol. 54, no. 12, pp. 1129-1142, Dec. 2008.
15. A. Akkas and M. Schulte, "Dual-mode floating-point multiplier architectures with parallel operations," Journal of Systems Architecture, vol. 52, no. 10, pp. 549-562, Oct. 2006.
16. A. Isseven and A. Akkas, "A Dual-Mode Quadruple Precision Floating-Point Divider," in Signals, Systems and Computers,2006. ACSSC
17. L. Huang, S. Ma, L. Shen, Z. Wang, and N. Xiao, "Low CostBinary128 Floating-Point FMA Unit Design with SIMD Support," IEEE Transactions on Computers, vol. PP, no. 99,pp. 1-8, 2011.
18. Gupta, A., U. Malviya, and Vinod Kapse. "Design of speed, energy and power efficient reversible logic based vedic ALU for digital processors." Engineering (NUiCONE), 2012 Nirma University International Conference on. IEEE, 2012.
19. Gupte, Ruchir, et al. "Pipelined alu for signal processing to implement interval arithmetic." Signal Processing Systems Design and Implementation, 2006. SIPS'06. IEEE Workshop on. IEEE, 2006.
20. Divic, Josip, and Marino Debeljuh. "Model of 8-bit microprocessor intended for lecturing." MIPRO, 2012 Proceedings of the 35th International Convention. IEEE, 2012.
21. Zhou, Yu, and Hui Guo. "Application specific low power ALU design." Embedded and Ubiquitous Computing, 2008. EUC'08. IEEE/IFIP International Conference on. Vol. 1. IEEE, 2008.
22. Liang, Getao, JunKyu Lee, and Gregory D. Peterson. "ALU Architecture with Dynamic Precision Support." Application Accelerators in High Performance Computing (SAAHPC), 2012 Symposium on. IEEE, 2012.
23. Kyaw, Khaing Yin, Wang Ling Goh, and Kiat Seng Yeo. "Low-power high-speed multiplier for error-tolerant application." 2010 IEEE International Conference of Electron Devices and Solid-State Circuits (EDSSC). 2010.
24. Lee, Kyoungwoo, et al. "Error-exploiting video encoder to extend energy/qos tradeoffs for mobile embedded systems." Distributed Embedded Systems: Design, Middleware and Resources. Springer US, 2008. 23-34.
25. Kathuria, Jagrit, M. Ayoubkhan, and Arti Noor. "A review of clock gating techniques." MIT International Journal of Electronics and Communication Engineering 1.2 (2011): 106-114.