Open access Journal International Journal of Emerging Trends in Science and Technology
DOI: http://dx.doi.org/10.18535/ijetst/v3i01.10

# Energy Aware 32 Bit Arithmetic and Logical Unit for Error Tolerant Applications

**Authors**
**Mamta Ratre[1], Jitendra Singh[2]**
[1]Deparment of Electronics and Telecommunication, Takshshila Institute of Engg. & Technology Jabalpur,M.P,India
Email: *ratre.mamta@gmail.com*
[2]Deparment 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 proposed an architecture of 32 bit General Purpose ALU .The critical power dissipation can be avoided by the application of clock gating of the hardware required and improving architectural approach for this in which we divide ALU is four sub block of 8-8 bit. These 32 bit ALU is identify input bit and according to that it will perform operation. Proposed architecture is combination of four 8 bit ALU which is accurate, semi accurate and approximate. Due to these logic we can save power consumption. 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 [23,24]present era which is known as Approximation 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 proposed methodology. Section 4 demonstrates Implementation details of proposed approach. Experimental results and its analysis are given in Section 5. Finally, Section 6 concludes the paper.

## 3. Proposed Methodology

As we know in present era every multimedia and general purpose application demands fast and ultra low power system. In current stage every device is operated on battery power supply and as we know for battery there is some kind of limit ion with their power issues and battery size. We also know for any processing unit there is most important part is Arithmetical logical unit (ALU). Due to heavy arithmetic and logical operation generally ALU require heavy amount of power as compare to complete system. Due to these reason we think for generation of approximate ALU unit. Which is combination of four ALU which are:

**1. Eight Bit Accurate ALU Unit**
**2. Eight Bit Accurate ALU unit**
**3. Eight Bit Semi Accurate ALU Unit**
**4. Eight Bit Approximate ALU Unit.**

In our proposed approach we also use logic of clock gating to reduce the clock power and dynamic power. Here our proposed design will work on total 16 instruction those are followings:

**Table 1:** Operation Details

| Arithmetical Operation | Logical Operation |
|---|---|
| 1. Addition | 1. And Operation |
| 2. Subtraction | 2. Or Operation |
| 3. Multiplication | 3. Nor Operation |
| 4. Division | 4. Nand operation |
| 5. Square | 5. Xor Operation |
| 6. Modules | 6. Xnor Operation |
| | 7. 1's Compliment |
| | 8. 2's Compliment |
| | 9. Right Shifting |
| | 10. Left Shifting |

In this table we represent instruction sets which we are use in our proposed ALU unit.

### 3.1 Proposed Architecture

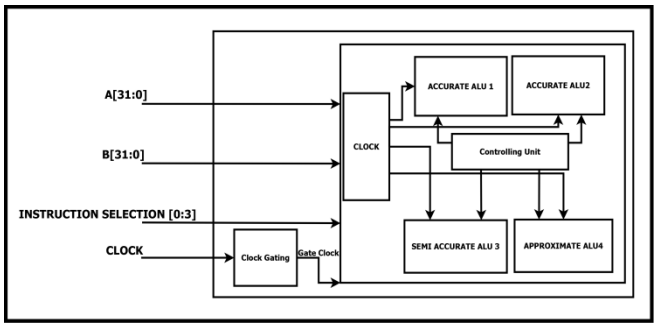



**Fig. 1** Proposed Architecture of 32 Bit ALU Unit

Here fig. 1 shows proposed architecture of 2 bit ALU unit. According to this generalized proposed architecture we are using divide and concurrent rule for ALU design. Basically here we are using four different eight bit ALU where 2 are accurate and one is Semi Accurate and one is Approximate. So according to Accurate ALU it will work like regular ALU architecture means it will create 100% accurate output. In semi accurate architecture we propose a new architecture where all logical operation is 100% accurate but in arithmetic operation accuracy level is between 95 to 99%. In pure approximate ALU architecture we are proposed approximate logical operations like 2's compliment and 80 to 90% accurate arithmetic operation.

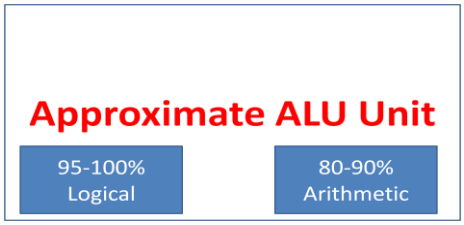
Accuracy Details of all four 8 bit blocks:

1. First Accurate 8 Bit


2. Second Accurate 8 Bit


3. Semi Accurate 8 Bit


4. Approximate 8 Bit:




As we can see above there is four blocks which are sub part of our proposed design. Here we are performing total 16 instruction according to accuracy level it is distribute in different ALU. Detailed Description about Proposed architecture.

1. Accurate ALU: In this architecture we are using all accurate logic and also this architecture is follows the previous existing architecture of ALU. Here it follows total 16 instructions set which is combination of arithmetical and logical.

2. Semi-Accurate ALU: In this architecture we are using some approximate and some accurate instruction set. In this block all logic is accurate and some arithmetic is accurate. Here for every approximate operation we will truncate 4 lsb bit of 8 bit ALU. Here it follows total 16 instructions set which is combination of arithmetical and logical.

3. Approximate ALU: In this architecture we are using all arithmetic instruction as a approximate and some logical as a approximate which is two's compliment. Here for every approximate arithmetic operation we will truncate 4 lsb bit of 8 bit ALU.

Here it follows total 16 instructions set which is combination of arithmetical and logical.

In our proposed approach we also use exsisting And gate based clock gating[25] technique for reduction of power consumtion. According to AND based clock gating in sequential circuit one two-input AND gate is inserted in logic for clock gating. One input to AND gate is clock and while the second input is a signal used to control the output (means it will control the sequential circuit's clock).

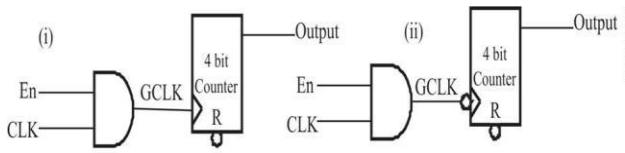



Fig. 2 AND based Clock Gating[25]

**4. Implementation Details:**

In this we are presenting the implementation details of previous existing and proposed approach. 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. Basically proposed design is work for 16 different instruction set which is combination of arithmetic and logical. Fig. 3, 4, 5 shows top level, gate level and Lut based proposed architecture.

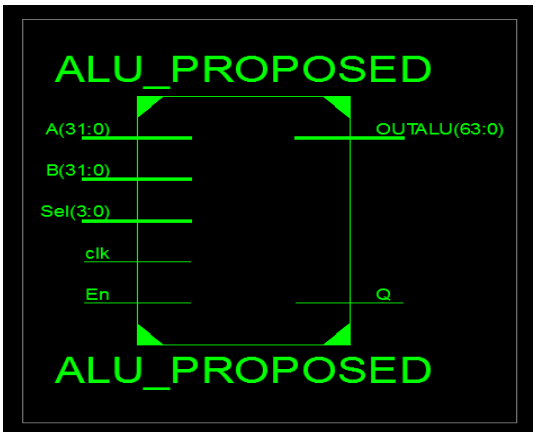



**Fig. 3** Top Level of proposed ALU

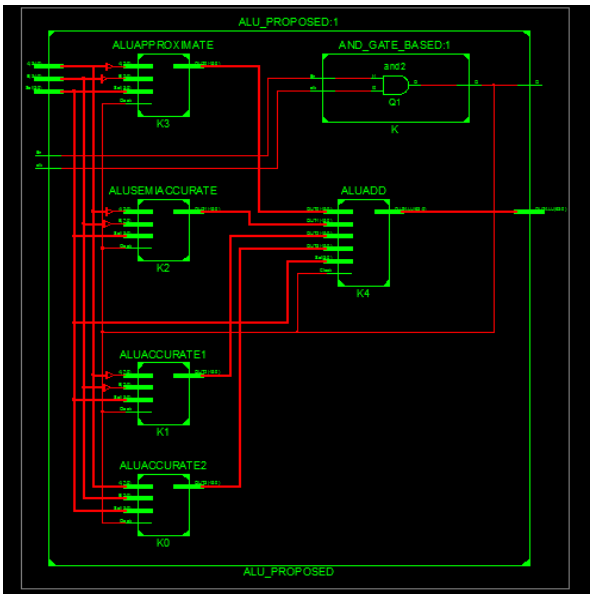



**Fig. 4** Gate Level Schematic of Proposed ALU

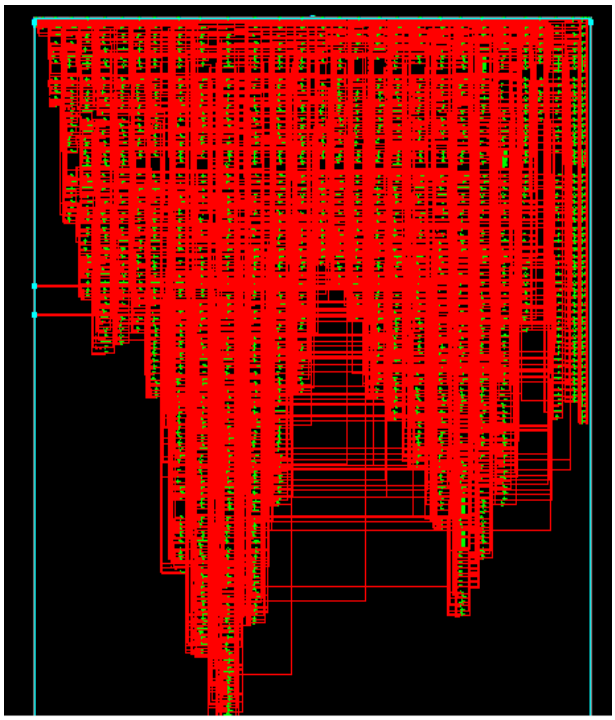


**Fig. 5** LUT Level Schematic of Proposed ALU

## 5. 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.

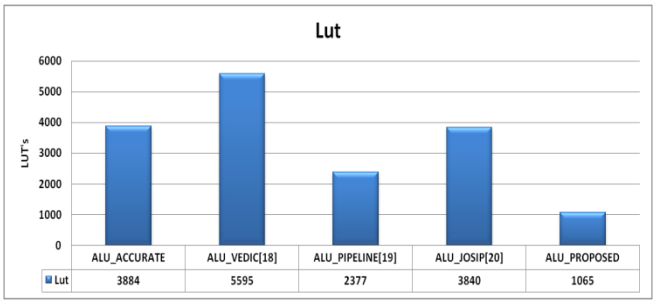



**Fig. 6** LUT Based Comparative Analysis

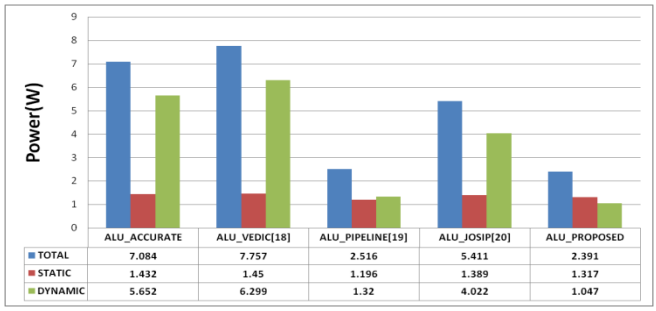



**Fig. 7** Power Based Comparative Analysis

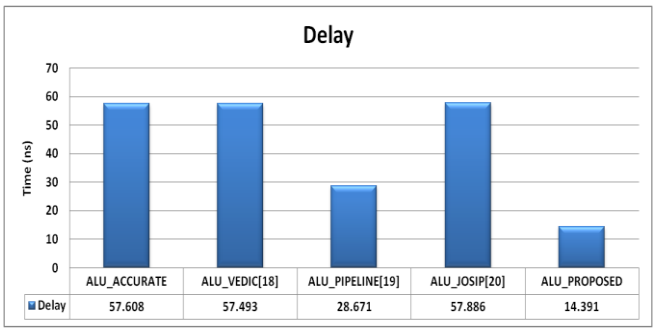



**Fig. 8** Delay Based Comparative Analysis

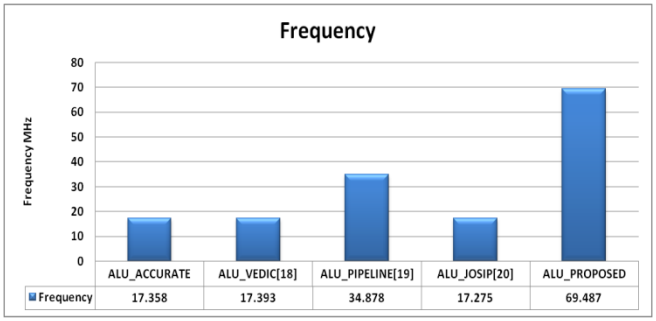



**Fig. 9** Frequency Based Comparative Analysis

As we can see in Fig. 6,7,8,9 there is 30-70% improvement in terms of area, power, delay and frequency as compare to previous existing approaches.

## 6. Conclusion

In this paper we are presenting a novel approach for 32 bit ALU architecture which is combination of four sub ALU block of eight bit. Where two blocks are 100% accurate , third block is semi accurate and fourth block is pure approximate. Here for power reduction we are using clock gating technique. Here and based clock gating technique is used which is well know clock gating technique. 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. Here simulation results shows that there is 75% improvement over delay and frequency. For power there is 50% improvement is done at last as for area there is 72% improvement is done as compare to previous existing approach.

## 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.