International Journal of Scientific Research in Science and Technology
Print ISSN: 2395-6011 | Online ISSN: 2395-602X (www.ijsrst.com)
doi : https://doi.org/10.32628/IJSRST

High Performance ALU Design using Energy Efficient Borrow Select Subtractor

Kuruba Surendra[1], Sri. D. Purushotham Reddy[2]

[1]M·Tech Student, [2]Assistant Professor

[1]Department. of ECE, [2]Wireless Sensor Networks

[1&2]Sri Krishnadevaraya University, Anantapur Andhra Pradesh, India

Article Info
Volume 9, Issue 5
Page Number : 463-470

Publication Issue
September-October-2022

Article History
Accepted : 05 Oct 2022
Published : 19 Oct 2022

ABSTRACT

The fundamental architecture of any processor, or ALU, is a key factor in determining how efficiently it operates. Addition and subtraction is considered to be an important operation in ALU so the circuit which performs these operations has a drastic impact on the processor's performance. The pace of operation in the subtraction process is determined by the sequential borrow bit, which travels from LSB to MSB encompassing all bits in the operands. Two modified borrow choose subtractor designs with improved area efficiency and lower power consumption are used in the proposed ALU circuit. As a result,a lesser number of gates are used for the logical flow of the subtraction process by employing blocks with fewer logic gates, which results in a reduction in the number of devices, area, and power dissipation. The proposed designs were implemented using Xilinx ISE 14.7 tool.

Keywords: ALU, Adders, Subtractors, Borrow select subtractor, Xilinx ISE 14.7

I. INTRODUCTION

The size, functionality, and dependability of the design were historically the three key considerations for VLSI circuit designers. However, as the need for portable gadgets has grown in recent years, designers' top concerns—on par with those of space and speed—are power and battery life. Since portable devices need extended battery lives and non-portable equipment need significant heat removal the low power VLSI designs are significantly more important. The two things which a low power VLSI circuit needs to tackle are its Analysis and optimization. The Analysis is focused on power or energy dissipation during the various designing stages of the circuit. The effectiveness and efficiency of various analysis methodologies varies. Depending on the information used to create a certain design, analysis might vary in accuracy. The process of optimization entails coming up with the best design possible while yet adhering to the design requirements. A subtractor's important work is to produce a binary divider. Power dissipation emerged as the main design constraints in systems since applications are battery-operated gadgets. Processor speed, circuit speed, space, performance, price, and dependability were formerly of utmost

importance. The secondary issue was power usage. Power usage, however, has received equal weight in recent years. The rapid growth of mobile computing devices and wireless communication systems, which require quick computations and intricate functionality with little power usage, is likely to be to blame for this changing trend. High performance CPUs also need a lot of electricity, which drives up the price of cooling and packaging. After that, the dependability is hampered by an increase in the power density of VLSI processors. A number of Silicon failure mechanisms have been found to cause silicon-based components to fail at a rate that is roughly doubled for every 10 degrees higher operating temperature. From an environmental standpoint, less heat will be dissipated in rooms, which will have a good effect on the environment at large. This is because less power is lost by electrical components. It will also use less power. The implementation of three-bit subtraction via a complete subtractor circuit. Minuend is the first bit, subtrahend is the second, and borrowed is the third bit in input. Borrow and difference bits are used for output.

The reference signal is used by the subtractor circuit to examine the data before allowing the signal to be used in the desired operation without altering its original form. The subtractor circuit is a highly helpful tool for signal processing, data processing, quick multiplication, and signal propagation. With the introduction of low-power manufacturing methods, circuit approaches, dynamically programmable power supply, and power-efficient microprocessors, low-power IC design has evolved as an active field of research and development. The subtractor circuit keeps the signal level constant and provides a lossless feed to the DSP processor circuit.

Due to the significance of arithmetic circuits in the operation of signal processing units, the Adder-Subtractor circuit is a need. The Ripple Borrow Subtractor (RBS) has a simple architecture which is commonly used for subtracting unsigned values. The time taken to propagate LSB to MSB limits RBS performance. The size of the binary word influences the RBS delay. To look into limitations of present RBS circuits with different RBS circuits, borrow select subtractor (BSLS) architecture is proposed. The final difference and borrow is selected post multiplexers (MUX). It is shown that the BSLS is an area waster since it makes use of a lot of RBS circuits. Addition of the two's complement is the standard method of subtraction for signed numbers.

The suggested structures are contrasted with the standard two's complement technique used in the literature, which essentially uses an addition. For comparing the two's complement approach with the suggested architectures, the adder has been taken into consideration. A BSLS variation is suggested for signed number subtraction. Our motto in this paper is area reduction and less power consumption which is done by implementing logic blocks having fewer gates that take up less space while keeping the same degree of logic capacity.

II. EARLIER WORK

Peripheral devices with addresses and various communication response times are employed with a data processing device. Values that reflect different address ranges make up the wait status, which is saved in programmable registers that may be addressed. The number of wait states determined by the value kept in one of the addressable programmable registers is the number of wait states that are generated when the digital processor asserts an address to the peripheral devices. Given the various peripheral device communication response times, this correlates to one of the address ranges where the asserted address occurs. The subtractor circuit uses a reference signal to assess the data and permits the signal to be used in the relevant operation without altering its original form. The subtractor circuit is a highly helpful tool for signal propagation, quick multiplication, and

signal processing. As a consequence of the introduction of low-power manufacturing processes, circuit approaches, dynamically programmable power supply, and power-efficient microprocessors, low-power IC design has grown into one of the most active fields of research and development.

Using an XOR gate, a NOT gate, and an AND gate, Fig. 1 depicts the structure of a half subtractor. A maximum delay of two is experienced by the half subtractor structure while creating the output. Difference D has a 1 delay, whereas creation of borrow B has a 2 latency. An XOR gate and an AND gate are used to create a half adder in Fig. 2. This half adder incurs a maximum of one delay in producing sum S and carrying C. Using two XOR gates, two NOT gates, two AND gates, and one OR gate, Fig. 3 depicts the internal construction of a complete subtractor. A maximum delay of 4 is experienced by the entire subtractor understanding of differences D experiences a 2 delay, but borrowing from B results in a 4 latency.

The implementation of an XOR, AND, and NOT gate in a 2-bit Binary to Excess-1 Converter (BEC) is depicted in Fig. 4. Five logic groups comprise the unsigned 16-bit BSLS. Each group divides the total into equal halves and borrows for any number of bits. Every functional group save the first has a (2n+2):(n+1) multiplexer, an n-bit BEC, and two n-bit RBS. In the first group, there is just one RBS with two bits.

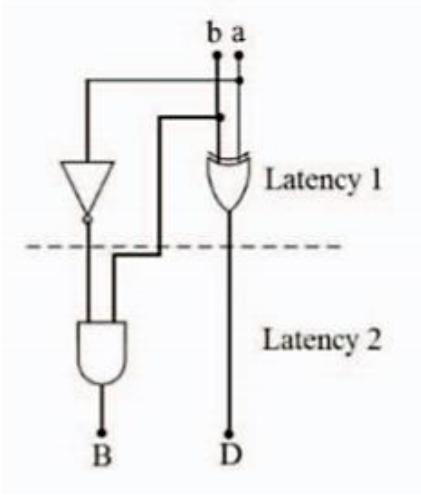

Fig. 1. Half Subtractor

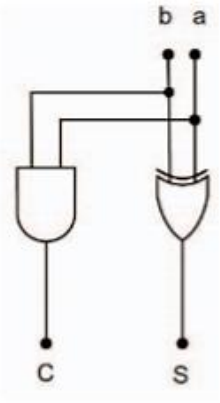

Fig. 2. Half Adder

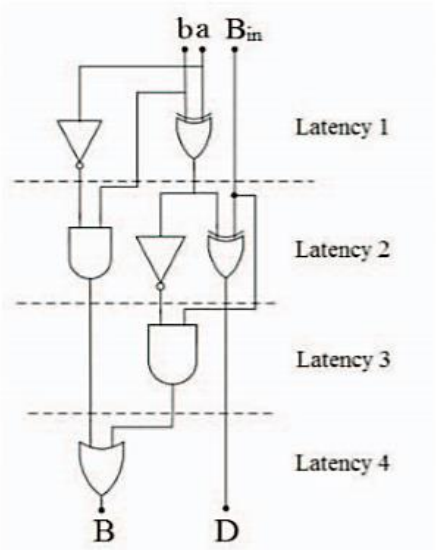

Fig. 3. Full Subtractor Gate Structure

RBS receives inputs from minuend and subtrahend The production of the difference bit is accounted for borrow if the previous step is 1. The borrow bits for difference and borrowing are provided as 0 and 1. Using the MUX, the borrow bit gained from the prior stage is used to determine the appropriate output.

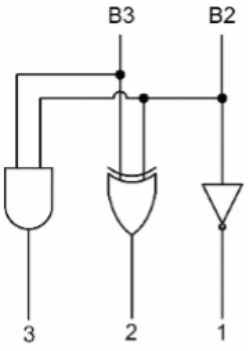

Fig. 4. 2-bit BEC

If the subtrahend is bigger than the minuend, the difference is taken as 2's complement. The number of

blocks required for changing into two's complement form into signed magnitude form are more. Successful results are obtained when the unsigned BSLS's borrow output is zero. As a result, signing the output's magnitude is not necessary. Similar to that, the result is unfavorable as 1 is the borrow output. The borrow-out output is then complemented, and the whole result is then performed using the 2's complement to represent it in signed magnitude form.

$X_0 = \tilde{}B_0$ (1)

$X_1 = B_1 \text{ xnor } B_0$ (2)

$X_2 = B_2 \text{ xor } \tilde{} (B_0 + B_1)$ (3)

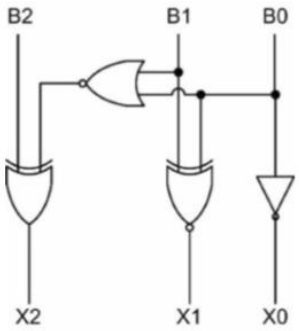

Fig. 6. 3-bit Binary-Less-One (BLO)

Despite the fact that the internal parts are different, this is similar to an adder. When the Bin reaches 0, it borrows after accepting them as input and deducts one from the difference. Whenever Bin is 1, the difference, borrow, and the BLO logic all are the same.

III. PROPOSED WORK

A modified Borrow Select Subtractor circuit using Binary-Less-One logic is proposed which is shown in Fig. 7. Leaving group 1 every other group uses Binary-Less-One logic to compute difference and borrow. The MUX circuit work is to select output difference and borrow based on Bin value from the previous stages. The difference and borrow from first n-bit RBS with Bin=0 and Binary-Less-One (BLO) read as Bin=1. The first group has 1 two-bit RBS and the remaining groups have an n+1-bit BLO, an n-bit RBS, and a (2n+2):(n+1) multiplexer.

One OR gate, two numbers of XOR, NOT, and AND gates, and one of these gates are commonly used to build entire subtractors.

You can build a half subtractor by using one XOR, one NOT, and one AND gate. The 2:1 MUX must be realized using 2 AND gates, 1 OR gate, 1 NOT gate, and 2 NOT gates. In order to process the first three bits using RBS, a half subtractor and two full subtractors are required. This specifies the initial 3-bit RBS's requirement for five XOR gates, five NOT gates, five AND gates, and two OR gates. Four gates—one XNOR, two XOR, two NOT, and one NOR—are used to implement the 4-bit BLO. The 8:4 MUX has four 2:1 MUX with 4 NOT, 8 AND, and 4 OR gates each.

In BSLS-BLO, group 3 requires a total of 40 gates. 7 XOR, 1 XNOR, 11 NOT, 13 AND, 6 OR, and 2 NOR gates were used to implement group 3 of the BSLS-BLO. In each and every group the second n-bit RBS, BEC-1, or n-bit BLO is substituted with an n-bit ripple borrow half subtractor.

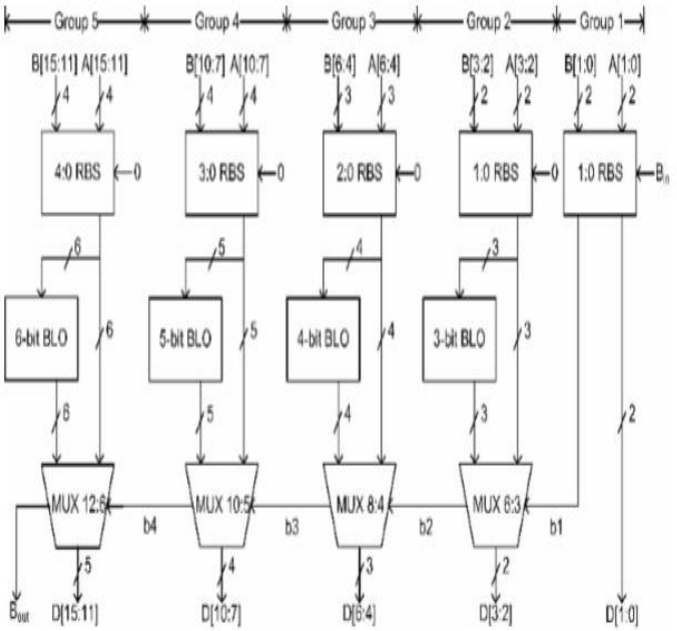

Fig. 7. Modified Borrow Select Subtractor using Binary-Less-One Logic

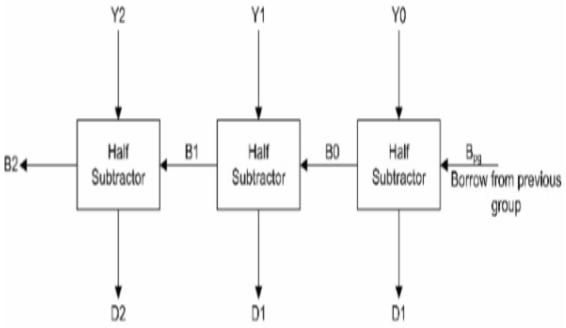

Fig. 8. 3-bit Ripple Borrow Half Subtractor

The received input is sent to the output as-is if borrow is set to 0. Otherwise, output is formed by taking away one from input (Bpg = 1). By using half subtractors, this is accomplished. The n half subtractors in an n-bit RBHS form the bit size. The n-bit RBS with Bin=0 serves as the input Y for each half-subtractor. Borrowing from the prior group or from the prior half subtractor serves as the half subtractor's additional input. The final borrow from the n-bit RBHS is the output of the nth half subtractor (Bout).

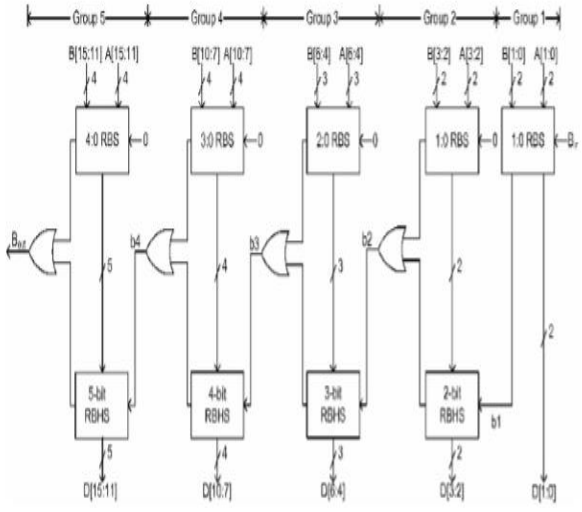

Fig. 9. Modified Borrow Select Subtractor using Ripple Borrow Half Subtractor

The proposed borrow select subtractor circuit is implemented using a ripple borrow half subtractor. By assuming Bin=0, the first n-bit RBS's difference and borrow bits are computed and then taken as input to n-bit RBHS. The difference and borrow are generated by borrow Bin from previous groups. A borrow-out bit called Bout is given to the OR gate along with n-bit RBS and n-bit RBHS.

For the present group and the following group Borrow-out and Borrow-in bits are the outputs of the OR gate respectively. In this architecture each group's (2n+2):(n+1) MUX is replaced by just one OR gate which reduces the gate count. One n-bit RBS, one n-bit RBHS, and one OR gate are present in each functional group, omitting the first group.

A high performance arithmetic hardware with the fewest clock cycles feasible that can execute fundamental arithmetic operations as well as the square, square root, and inverse. In this thesis, the design of the ALU was done with high performance and testability in mind. For the development of a high-speed arithmetic unit, architectures with a high degree of parallelism were investigated. Functional units were created with a 16 bit capacity for simplicity. Increases in operand size would only necessitate hardware duplication in parallel with current circuitry due to architectural parallelism. The ALU features independent hardware that can compute square, square root, and inverse in addition to standard integer arithmetic operations. A combined ADD/SUB unit was used to implement the hardware for addition and subtraction. Usually, two's complement addition is used for subtraction. Implementing high performance arithmetic hardware has always been a desirable design challenge due to the widespread usage of microprocessors and signal processors.

The workhorse of microprocessors and the component that controls how quickly the processor operates is the arithmetic and logic unit (ALU). Basic arithmetic computations can be performed independently by all contemporary CPUs. Processors come with rapid arithmetic hardware as well as on-chip memory (cache), which helps them operate much better by reducing delays brought on by data access from main memory.

The most crucial component in the architecture of a digital system is an arithmetic logical unit. It serves as a combinational logic unit and an essential component of a computer processor, carrying out

both arithmetic and logical operations. In very large-scale integrated circuits (VLSI), which range from CPUs to application-specific integrated circuits, ALUs of different fixed bit-widths and full precision bit widths are usually used (ASICs). Today's ALU is becoming more complicated and smaller to allow for the creation of processors and computers that are more powerful yet smaller.

As a result of computer, digital signal processing, and networking applications, there is a growing need for processors that operate at fast speeds, consume little power, and are interoperable. Multiplication, addition, division, subtraction, and other arithmetic operations are performed utilizing a variety of processor types that are employed in diverse applications.

Effective arithmetic circuits can be implemented to improve the performance of the ALU. A simple ALU that performs 4 separate operations has been created in this case. ALU architecture may be produced that is more effective than traditional designs by implementing an efficient subtractor as one of the operations.

IV. EXPERIMENTAL RESULTS

In the figure 8 & 9 RTL schematic and Technology schematic of ALU_BSL1 are shown below respectively.

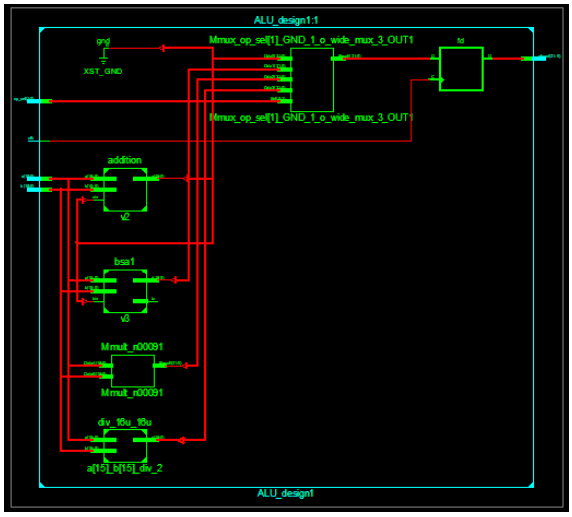

Fig.8: RTL schematic of proposed ALU_BSL1

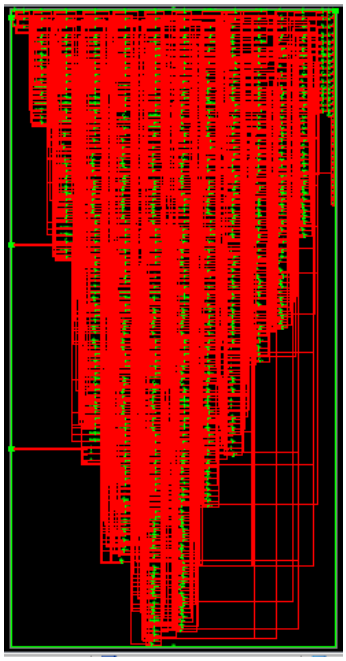

Fig.9: Technology schematic of proposed ALU_BSL1

In the figure 10 & 11 RTL schematic and Technology schematic of ALU_BSL1 are shown below respectively.

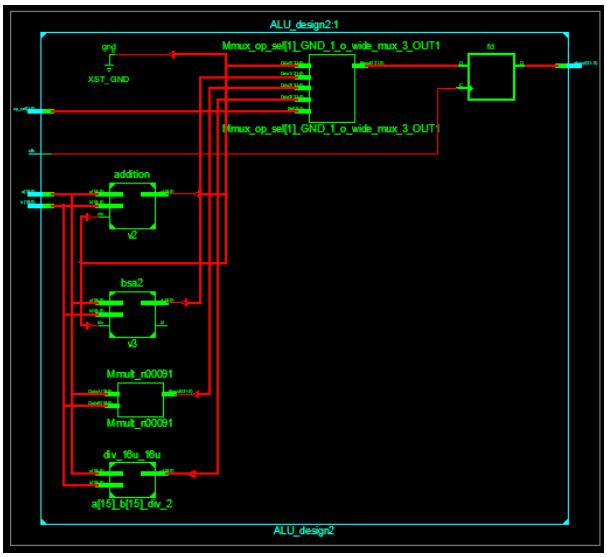

Fig.10: RTL schematic of proposed ALU_BSL2

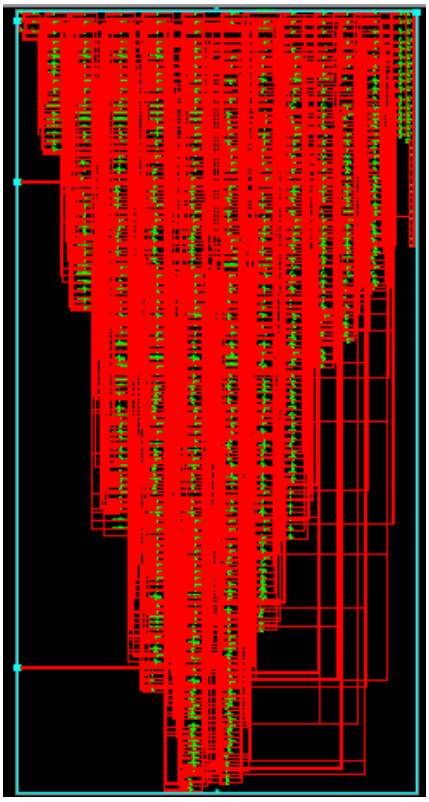

Fig.11: Technology schematic of proposed ALU_BSL2

Figure 12 & 13 shows the simulation results for the proposed ALU_BSL1 & ALU_BSL2 respectively.

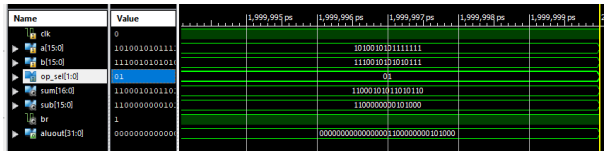

Fig.12: Simulation results of proposed ALU_BSL1

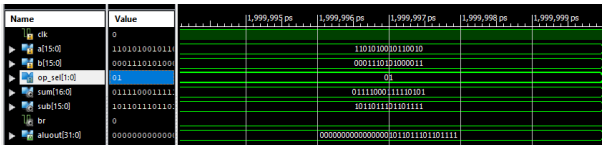

Fig.13: Simulation results of proposed ALU_BSL2

Evaluation Table:

	Area (LUT's)	Delay (ns)
ALU_BSL_1	450	33.657
ALU_BSL_2	443	33.441

V. CONCLUSION

The application to the suggested subtractor designs is what this paper's implementation is all about. We describe an ALU architecture that employs modified borrow select subtractors employing BLO and RHBS, respectively. In comparison to the equivalent designs, fewer gates are used in both modified architectures. Additionally, this design strategy results in a smaller overall footprint, less power consumption, and fewer transistors being used, all of which have an impact on how effective the ALU architecture is. The suggested ALU, which uses a modified borrow select adder and RHBS, takes up less space and has a shorter latency, according to the findings of the simulation.

VI. REFERENCES

[1]. L. Maruti Kunjir and V S Kanchana Bhaaskaran, "A high speed borrow select 16-bit subtractor", The Patent Office Journal Appl. 2868/CHE/2014 A, Jan. 22, 2016.

[2]. B. Ramkumar and Harish M Kittur, "Low-Power and Area-Efficient Carry Select Adder", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 2, pp.371-75, Feb. 2012.

[3]. Kore Sagar Dattatraya and V. S. Kanchana Bhaaskaran, "Modified Carry Select Adder using Binary Adder as a BEC-1,", European Journal of Scientific Research V.103, no.1, pp.156-164, Jan. 2013.

[4]. Y. Kim and L.-S. Kim, "64-Bit Carry-Select Adder with Reduced Area," Electronics Letters, Vol. 37, No. 10, pp. 614–615, May 2001.

[5]. B. Srinivasa Ragavan, B. P. Bhuvana and V. S. Kanchana Bhaaskaran, “Low power 64-bit carry select adder using modified exnor block”, Journal of Engg. and Applied Sciences, Vol.10, pp. 17294, Dec, 2015.

Cite this article as :

SaiKuruba Surendra, Sri. D. Purushotham Reddy, "High Performance ALU Design using Energy Efficient Borrow Select Subtractor", International Journal of Scientific Research in Science and Technology (IJSRST), Online ISSN : 2395-602X, Print ISSN : 2395-6011, Volume 9 Issue 5, pp. 463-470, September-October 2022.
Journal URL : https://ijsrst.com/IJSRST2295202