Design and implementation of a Microcontroller Arithmetic and Logical Unit with Optimized Area, Power, and Performance on FPGA

Mrs.M.Haritha[(1)] Mr.N.B.Jilani[(2)] Puvvada Manideepika[(3)] Yamanuri Lavanya[(4)] Basani Gnana Vishnu Priya[(5)] Solleti Deepika[(6)]

[1,2] *Krishna Chaithanya Institute Of Technology & Sciences, Ece Department, Markapur, Andhra Pradesh.*
[3,4,5,6,] *Krishna Chaithanya Institute Of Technology & Sciences, UG Student-ECE, Markapur, Andhra Pradesh.*

Abstract In this paper, compact expressions for the jitter in clock and data recovery (CDR) circuits based on bang-bang phase detectors are proposed. These expressions consider the quantization noise related to the finite number of phases of the phase interpolator (PI), which aligns the receiver clock to the incoming data, as well as the phase noise of the transmitter and receiver oscillators. Various methods for detecting edges and deserialized data using Early/Late detection are compared. Majority voting reduces CDR bandwidth and amplifies the effect of clock jitter on CDR jitter; counting individual Early/Late occurrences, on the other hand, does not reduce bandwidth but does increase noise associated with the PI's finite phases. Validation of the suggested analytical results are simulated in Xilinx Vivado verilog Hdl.

Keywords: ALU-Arithmatic,5G Communications, FPGA, Verilog HDL, PPA-Power Performance and Area.

I.INTRODUCTION

The power of any embedded system is assessed based on how powerful the processor is, and the capability of processor is adjudicated by the ALU which invariably boils-down to the instruction set. ALU is a core part of a processor, and its design will play a vital role in execution of every instruction.

The three primary parameters under consideration for ALU design are always power dissipation, speed of operation and chip area. As the designer tries to minimize one parameter, at least one of the other two parameters increases, thereby creating a challenge. Hence, optimizing all the three parameters in a proper balance is crucial. The circuit designers and researchers have tried hard to explore and exploit every avenue in the chip designing to achieve less power consumption, increased speed of operations and optimize area on the chip. The advancement in technology and ever rising demand for hand-held devices have compelled chip designers to focus on area reduction/ optimization.

The miniaturization of the digital circuits can be achieved either by reducing the size of transistor or by minimizing the gate count of the design. The first approach tries to keep up with Moore's law and has reached sub-nanometer technology leading to advancements in CMOS VLSI. The latter approach has less buyers due to the need of sound knowledge in logic design which demands more human efforts. It deals with application of Boolean rules to simplify the equations thereby reducing the complexity of the circuits. The very careful design and testing of ALU are crucial in the process of area optimization.

The Central Processing Unit (CPU) is the core of any embedded system. The CPU's are categorized based on the processing devices as Application Specific Integrated Circuits (ASIC),

Field Programmable Gate Arrays (FPGA), Microprocessor, Digital Signal Processor (DSP), Microcontrollers. Each of these processing devices are targeted for specific application. The major feature that distinguishes all the above specified processing devices is primarily it ALU. The ALU is generally an asynchronous logic circuit which acts as the core of any processing device. The ALU of the processing devices vary in its functionality, decoding logic, size of the instruction set based on the device it's going to be incorporated.

ASIC's are extremely fast as they are designed for a dedicated application and occupy the least amount of area on the chip. The design and fabrication of an ASIC consumes significant amount of time and man hours making them expensive at a smaller scale but would turn out to be cost effective at large scale. FPGA designs are slower than the other counterparts and occupies a significantly larger area on chip but dominate over the others due to their hardware reconfigurable architecture, and shorter design and testing time.

The Microprocessors find their application in the highspeed fixed-point arithmetic, logical and data transfer operations. They are suitable for the more of a general-purpose application ranging from a personal computer to a supercomputer. They are designed generally using Van Neumann Architecture along with the complex instruction set (CISC) which makes the decoding logic bulky at the same time making it programmer friendly.

Digital Signal Processors operate on floating-point arithmetic and is supported by a Multiply and Accumulate (MAC) unit to perform faster multiplication and addition; and a Barrel Shifter to perform faster logical operation. This increases the area occupied on the chip, hence increasing the cost. The DSP’s operate at lesser clock frequencies compared to Microprocessors, which reduces the power dissipation on the chip. The DSP’s find their application in all multimedia applications due to the presence of MAC unit, Barrel shifter and a floating point ALU.

Microcontrollers are primarily system-on-chip (SoC), which works on fixed-point arithmetic at even lower clock frequencies compared to DSP’s. The lesser external interface is required for the system to operate for a similar functionality when compared to Microprocessors. Microcontrollers operating at very low speeds that is suitable for the rate of change of physical parameters makes them handy to interface with the external world. The lower prices of microcontrollers add on to the demand for them in the industry. Both DSP’s and Microcontrollers are designed using Harvard Architecture along with the reduced instruction set (RISC) leads to the smaller decoding logic thus reducing the chip area compared to microprocessors. The multiply/divide operations are performed using microcode architecture in Microprocessor/ Microcontroller, whereas DSP's use hard-wired architecture.

Several architectures have been investigated previously with an intention to achieve trade-off between power and performance. Little has been done in the area optimization, which will lead to a win-win situation for both the cases: as lesser area requires lesser power and reduced number of gates will lower the complexity thereby increasing performance.

The aim is to achieve lower area and decent performance with our proposed architecture. The work focuses on optimizing the circuit by reducing the gate count and designing a more powerful device. The power of any embedded system is assessed based on how powerful the processor is; and the capability of the processor is adjudicated by ALU, which invariably boils down to the instruction set. The very careful design and testing of the ALU are crucial in the process of area optimization. Verilog is used for the design and verification of the IP core. The Verilog designs are said to occupy lesser area if the number of Look-Up-Tables (LUT) and the number of slices is reduced in the design. Thus, conclude that the chip area can be reduced significantly by reducing the gate count of the design for the same operation.

An Arithmetic Logic Unit (ALU) is a fundamental component of the central processing unit (CPU) in a computer. It is responsible for performing arithmetic and logic operations on binary numbers. The ALU is a critical part of the processor that executes instructions and manipulates data during computation. Here are key aspects of an ALU:

1. Basic Functions:

Arithmetic Operations: Addition, subtraction, multiplication, and division of binary numbers.

Logic Operations: Bitwise AND, OR, XOR, and NOT operations.

2. Inputs:

An ALU typically has two input operands, often labeled A and B, for binary arithmetic and logic operations.

3. Output:

The result of the operation is produced as the output. For arithmetic operations, this is the arithmetic result; for logic operations, it is the logical result.

4. Control Inputs:

Control lines or bits are used to specify the type of operation the ALU should perform. These control inputs determine whether the ALU should perform addition, subtraction, logical AND, OR, XOR, or other operations.

5. Flags:

ALUs often set flags based on the result of an operation. Common flags include:

Zero Flag (Z): Indicates if the result is zero.

Carry Flag (C): Indicates if there is a carry-out in arithmetic operations.

Overflow Flag (V): Indicates if there is overflow in signed arithmetic operations.

Negative Flag (N): Indicates if the result is negative.

6. Multiplexer (MUX):

A multiplexer is used to select the appropriate output based on the control inputs. It routes the result of the desired operation to the output.

7. Combinational Logic:

The ALU is implemented using combinational logic circuits, such as adders and logic gates. Combinational circuits generate outputs solely based on their current inputs, without any memory of past inputs.

8. Bit Width:

The bit width of an ALU determines the size of the binary numbers it can operate on in a single operation. Common bit widths include 4-bit, 8-bit, 16-bit, 32-bit, and 64-bit ALUs.

9. Speed and Pipelining:

ALUs are designed for high-speed operation. In some architectures, multiple ALUs can be pipelined to further enhance processing speed.

10. Application:

ALUs are used in a variety of computing devices, from microcontrollers to powerful CPUs in desktop computers. They are an essential part of the processor's execution unit.

11. Functionality Extension:

Some ALUs include additional functionality such as shifting, rotation, and other specialized operations to enhance their versatility.

II.EXISTING SYSTEM AND PROPOSED SYSTEM

PROPOSED ALU SYSTEM

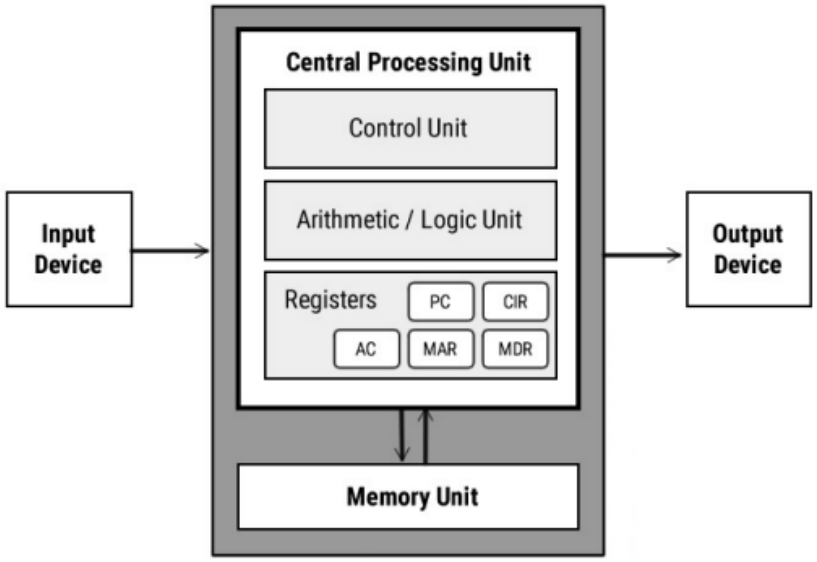

FIG.1.proposed ALU architecture

The ALU operations are primarily divided into Arithmetic operations and Logical operations. The Arithmetic operations like Addition, Subtraction, Increment, Decrement and so on are performed by Arithmetic unit, while Logical unit performs AND, OR, XOR, complement, rotate, shift operations etc. The multiplexers and de-multiplexers play a vital role in selecting the respective results generated by the ALU. The Microcontrollers need to handle the individual port lines which necessitate the need for bit handling instructions. Thus, an ALU of a Microcontroller needs to cater to both the bitwise and the byte-wise Logical operations, arithmetic and data transfer operations. In spite of incorporating all the ALU operations, the challenges lie in optimizing the area occupied on the chip.

The basic circuit for the design of an arithmetic unit is a full adder. There are multiple ways in which various types of adders are designed like Ripple Carry Adder (RCA), Carry Look Ahead Adder (CLA), Carry Save Adder (CSA), Carry Select Adder (CSeA) etc. The respective adders are selected based on the parameter to be minimized. If the focus is on reducing the delay, then CSeA would be a better choice but penalizes with a large chip area and more power-hungry. While CLA would be relatively slower than the CSeA but occupies significantly lesser area. The CSA would be a better choice as the number of bits needed to be added increases as in case of multipliers. The RCA occupies the least possible area but is slower as the number of bits to be added increases.

The proposed architecture focuses on the area optimization and hence RCA has been used in the entire design. The ALU is designed to be incorporated in a Microcontroller along with a barrel shifter to make logical operation yield quicker results. Since the clock frequency targeted for the microcontroller is 20MHz, the speed of operation has not been considered. The complete instruction set of the ALU has been designed.

2.INSTRUCTION SET

There are 8 bit-handling instructions designed as described below: MOV ci, bit: Transfers the content of bit location identified by the Op-code (lower bits) in input 'a' to carry flag (represented by co output in the waveform). XOR ci, bit: Performs XOR operation between the content of bit location identified by the Op-code (lower bits) in 'a' input with carry flag (represented by ci input in the waveform) and store it in the carry flag (represented by co output in the waveform). AND ci, bit: Performs AND operation between the content of the bit location identified by the Op-code (lower bits) in 'a' input with carry flag (represented by ci input in the waveform) and store it in the carry flag (represented by co output in the waveform). OR ci, bit: Performs OR operation between the content of the bit location identified by the Op-code (lower bits) in 'a' input with carry flag (represented by ci input in the waveform) and store it in the carry flag (represented by co output in the waveform). MOV bit, ci: Transfers the carry flag (represented by ci input in the waveform) to the bit location identified by the Op-code (lower bits) in output y. INV bit: Compliments the content of the bit location identified by the Op-code (lower bits) on input 'a' and stores in the same bit position on output y. CLR bit: Clears the content of the bit location identified by the Op-code (lower bits) on input a and stores in the same bit position on output y. SET bit: Sets the content of the bit location identified by Op-code (lower bits) on input 'a' and stores in the same bit position on output y accumulator.

The results of multiplication and division would be stored in exclusive memory locations meant for multiplication and division registers. The NOP instruction, data transfer and exchange instructions are part of this Opcode range. The RLC and RRC instructions perform left and right rotation of the data through carry flag. The waveforms are shown in Fig 3b and Fig 3c. rsrl: This block performs rotate and shift operations left or right by the number of bits specified in the Op-Code (lower bits). These operations are carried out when the Op-Code (higher bits) range from 0100 to 0111. The barrel shifter is designed in this block. The waveforms are shown in Fig 3d. md16_8: This block performs 8-bit * 8-bit multiplication of a and b inputs to yield 16-bit result on res_h and res_l output lines and division of 16-bit (a and b inputs put together) by 8- bit number on d input to yield 16-bit quotient on res_u and res_h and 8-bit reminder on res_l output lines. The waveforms are shown in Fig 3e alu_final: This block integrates anoxcsmm_1, alu, rsrl, and md16_8 blocks.

III. RESULTS AND ANALYSIS DISCUSSION

FIG.1. ALU OUTPUT

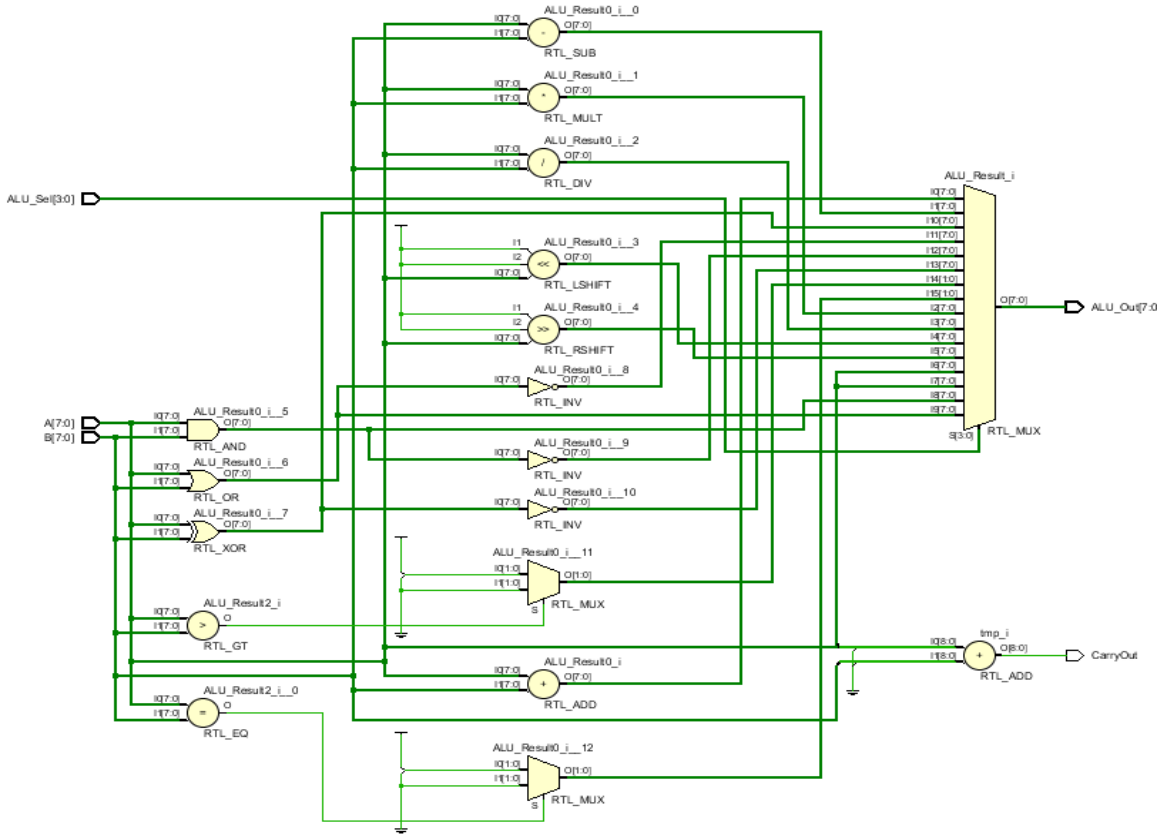

FIG.2.ALU RTL SCHEMATIC DIAGRAM

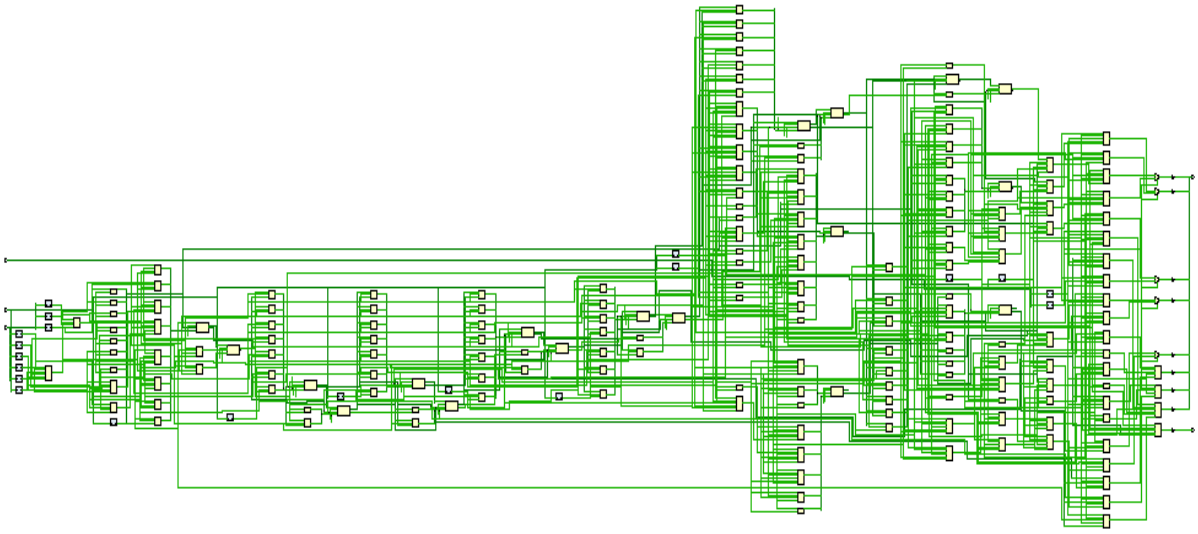

FIG.3.ALU SYNTHESIZED DIAGRAM

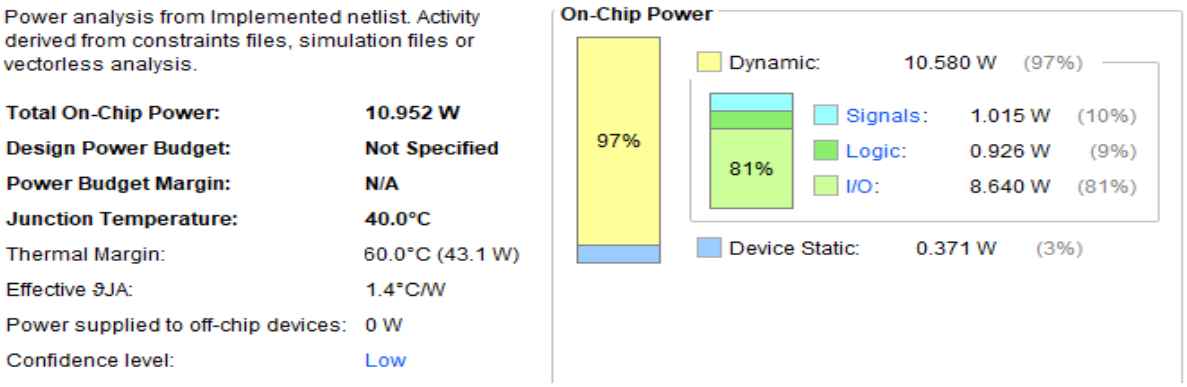

FIG.4.ALU ESTIMATED POWER

Resource	Utilization	Available	Utilization %
LUT	146	117120	0.12
IO	29	204	14.22

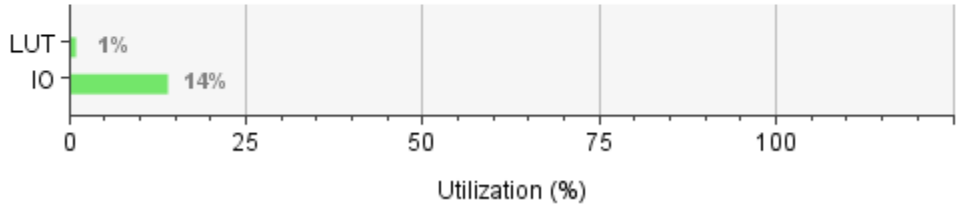

FIG5.ALU AREA LUTS UTILIZATIONS

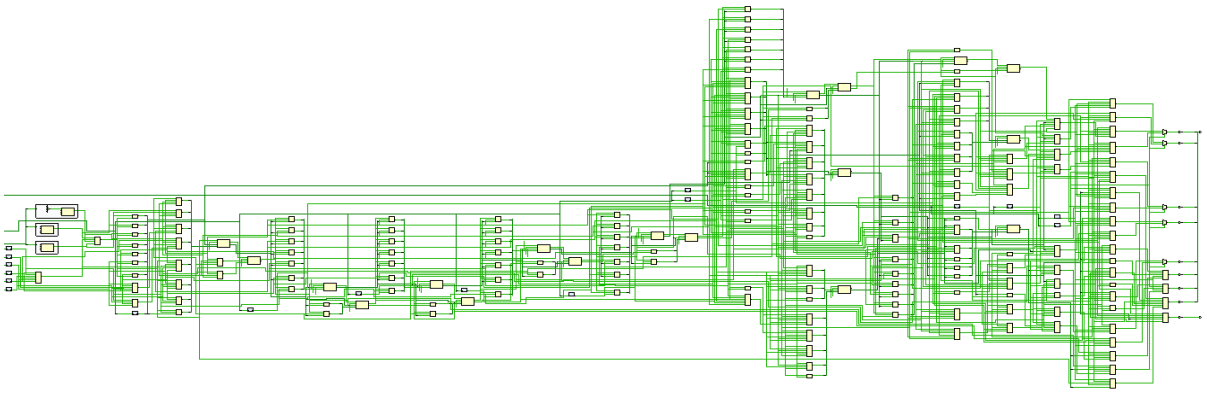

FIG6 ALU OPTIMIZED OUTPUT

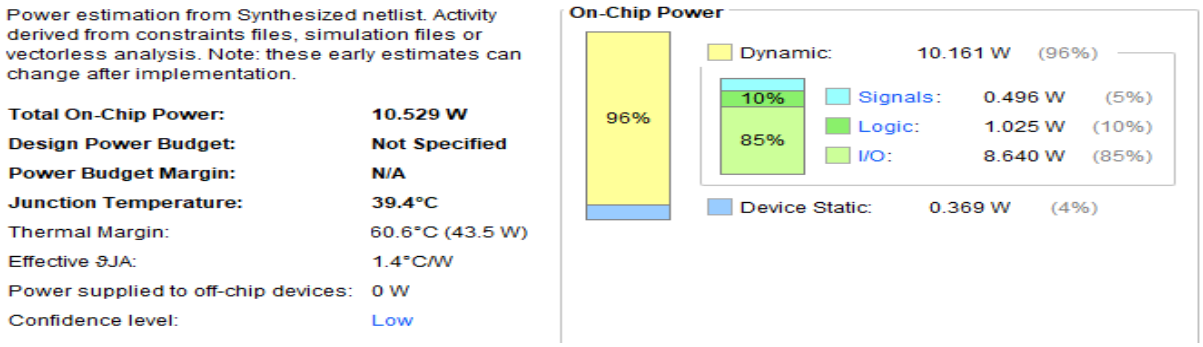

FIG7.ALU OPTIMIZED POWER REPORT.

Resource	Utilization	Available	Utilization %
LUT	146	117120	0.12
IO	29	204	14.22

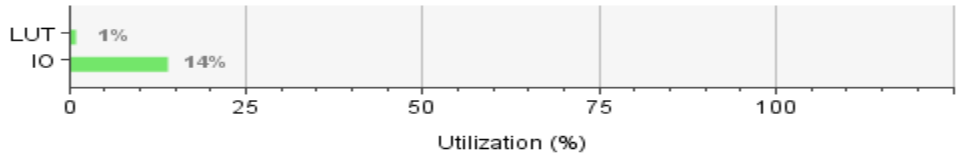

FIG8.ALU OPTIMIZED AREA UTILIZATION

CONCLUSION

An asynchronous ALU has been designed with 4 modules namely: bit handling ALU block, byte handling ALU, Shifter block, and hard-wired Multiplier and Divider block. The proposed architecture is area efficient and is optimized at gate level. The novel architecture incorporating hard-wired multiplier and divider along with barrel shifter have shown good results. Thus, the chip area was reduced significantly by reducing the gate count of the design for same operations. As a part of future enhancement, we can focus on integrating the proposed ALU on to a microcontroller and further extend it to ASIC to reduce the delay.

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)
[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
[8] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)
[9] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[10] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[11] K. Elissa, “Title of paper if known,” unpublished.
[12] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[13] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[14] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
[15] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
[16] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)
[17] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[18] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[19] K. Elissa, “Title of paper if known,” unpublished.
[20] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
[21] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[22] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.
[23] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
[24] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.