International Journal for Innovative Engineering and Management Research

PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL

www.ijiemr.org

IJIEMR Transactions, online available on 29th May 2021. Link :http://www.ijiemr.org/downloads.php?vol=Volume-10&issue=Issue05

10.48047/IJIEMR/V10/ISSUE 05/62

TITLE: Low Power and Area Efficient 64 Bit Arithmetic Logical Unit Design

Volume 11, ISSUE 05, Pages: 277-293

Paper Authors **P S. Surekha1, Amandu Swathi2, K. Shravani2, A. Srinivas2, P. Aravind2**

International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

Low Power and Area Efficient 64 Bit Arithmetic Logical Unit Design

P S. Surekha[1], Amandu Swathi[2], K. Shravani[2], A. Srinivas[2], P. Aravind[2]

[1]Assistant professor,[2]UG Student, Department of Electronics and Communication Engineering

[1,2]Malla Reddy Engineering College and Management Sciences, Kistapur, Medchal-50140l, Hyderabad, Telangana, India.

ABSTRACT: The adders, multipliers are the essential building blocks for every integrated circuit (IC) and especially for Arithmetic and Logical units (ALU). Thus, the design of adders and multipliers must inhibit the area, delay and power efficient properties. But most of the conventional adders are failed to provide these properties in multipliers andALUs implementation. To solve this problem Hybrid Koggestoneadder (HKSA) has been developed at nano technology level using the parallel prefix addition (PPA) properties. The quantum cost for this HKSA isvery low, thus in this paper HKSA based N-bit adder, N-bit subtractor, N-bit multiplier and N-bit ALU developed with reconfigurable properties. The effective utilization of the HKSA provides more flexible nature for ICs. The implementations are conducted in Xilinx ISE environment, the simulation results shows that proposed method is area, power and delay efficient compared to the conventional approaches.

Keywords–HKSA, ALU, PPA

1. INTRODUCTION:

The digital computer performs operations that seem to discard data in computer's history. In this, the machine state will be ambiguous [1]. The operations of computers incorporated to overwrite/erase the data and also consists a section which addresses a bit of data addressed at distinctive transfer instructions. Hence, the system is logically irreversible - its transition work lacks a single-esteemed inverse [2]. In development of nanotechnology, which is speed enhanced, less sized, and composed of highly convoluted engineering design than existing systems. The improvement in the technology has introduced a system considering the parameters like power and heat dissipation and clock frequency [3]. The highly enhancement in the clock frequency to get the better speed and increase the total transistors stuffed in a chip to accomplish required system results more power consumption. Almost in all the total logic gates for logical operations in old computer are irreversible. Hence, in every time a logical operation is performed to know input lost and it is dissipated as heat. For any digital design, the power loss must be considered for desired parameter [4]. The current technology improvement in the computer design are increased and also the power utilization also optimized by using the CMOS logic. As per literature [5] with exponential development of heat produced because of information loss is major issue. The heat dissipation causes the reduction in the circuit's execution time and lifetime. Thus, the use of reversibility can give low power consumption and heat dissipation system. In the work of literature [6] indicated how the reversibility can be achieved with zero power dissipation. Also, reversibility cannot cause information loss as like in irreversible. For reversibility circuit design we need set of gates and these kinds of gates are available since last decades. As per literature [7] focused on logical irreversibility which causes more heat losses. Accordingly, a computer must dissipate entropy (kTln 2) of energy for each data loss. An irreversible computer can made reversible by conserving the information. The reversible machine additional unit is used to store the every operation performed. In this machines the controls both the input and output information. Thus, as discussed in [7], this will prevent the information loss as it can be reused. Thus reversible computer halts the information, can be erased in middle results, the output can be reused as input. The research starts with the concept of reversibility [7] where at the ending of computation and first inverse of the transition work, the system can perform reverse computation.

he work of Yugandhar, K., et al [8] gives a new design method for array multiplier which uses more garbage outputs. Authors considered the 4-bit reversible high performance array multiplier (RHPAM) with reversible high performance adder (RHPA) synthesizing by utilizing the advanced bi-directional synthesis mechanism. Hence, for multiplier synthesis, the optimization of input bits and also the delay are not yet addressed except in the recent works which discusses about the post synthesis mechanism to reduce the quantum bits in the reversible multiplier.

Kamaraj, A., and P. Marichamy [9] introduced the fault tolerant ALU (FTALU) with no input carry with one ancillary input bit. Authors have examined new QR carry adder designs with no ancillary input bit gives improved delay. The reversible ALU in the existing literature is evaluated by garbage outputs, total RL used, QC and delay.

Amirthalakshmi, T. M., and S. Selvakumar Raja [10] have described concepts of 8-bit Reversible ALU (RALU). Also, designed and implemented high cost, efficient, fault tolerant, reversible ALU.

In this more garbage outputs were compensated with fewer operations. The author concluded that the ALU performs all the logical operations better than existing methods not arithmetic operations.

Dasharatha, M., et al [11] addressed a concept that a function can be reversible if every vector produces equal number of outputs. In this the High speed ALU (HSALU) design is presented by making use of control signals and vedic multiplier units. With this design author has found that the proposed design is more effective as per garbage outputs and constant inputs are considered.

Rahim, B. Abdul, et al. [12] described a novel nonprogrammable logic gate and verified its implementation in ALU design using reversible multipliers. With this work author has found that the delay and are of ALU using RG should be low.

A new QCA reversible based design technique is suggested by Oskouei, Saeed Mirzajani, and Ali Ghaffari [13] focused on the design for 8 bit reversible QCA ALU and was utilized Xilinx as Synthesis tool. The ALU performs the reversible QCA addition(RQCAA), but it consumes more area and delay . Through this author achieved high propagation delay and high power dissipation by implementing 8bit arithmetical operations.

Shukla, Vandana, et al [14] described a design and implementation of reversible ALU of N-bit through low power addition (LPA). The ALU reduces the propagation delay and the power dissipation also reduced through clock gating but cost ineffective. Table 1 summarizes the existing methods problems addressed and its outcome.

Table 1: Summary of Literature survey

Authors	Problem addressed	Outcome
Yugandhar, K[8]	Synthesis of reversible gates, Adder and multiplier design issues	ALU design with Low QC, delay and garbage output
Kamaraj, A., and P. Marichamy [9]	RL design Reduced minimum outputs	Reduced minimum outputs
Amirthalakshmi, T. M., and S. Selvakumar Raja [10]	Reversible design with sequential circuits	Better design than traditional RL design
Dasharatha, M., et al [11]	Adder design	ALU design with Low QC, delay, garbage output

Rahim, B. Abdul, et al. [12]	Design aspects of ALU using RL	Efficient ALU design for all mathematic operations
Oskouei, Saeed Mirzajani, and Ali Ghaffari [13]	Issues of ALU design process	Fault tolerant ALU
Shukla, Vandana, et al [14]	Off-chip biasing issues	Low power Nbit ALU design

To solve these problems, the paper is contributed as follows

- A64-bit HKSA has developed utilizing the reducing the parallel prefix operations.
- An N-bit multiplier has been developed by utilizing the 64-bit HKSA.
- Using this adder, subtractor and multiplier and64-bit ALU was developed.
- A detailed analysis of results has been presented with comparison to the existing method.Theresults show that proposed method is area, power and delay efficient.

Rest of the paper as follows; section 2gives the detailed analysis over proposed HKSA. Section 3gives the detailed analysis of N-bit multiplier.Section 4gives the detailed analysis of 64-bit ALU.Section 5gives the detailed analysis on results with respect to both simulation and synthesis outcome and comparative analysis also performed with various convention approaches. Section 6deals about conclusion and future works of proposed methods.

2. PROPOSED HYBRID KOGGE STONE ADDER:

Let$A = a_{n-1}, a_{n-2}, a_{n-3} \ldots . a_1, a_0$and $B = b_{n-1}, b_{n-2}, b_{n-3} \ldots . b_1, b_0$be the n-bit augend and n-bit addend, $c_i$be the carry input respectively, then binary addition is defined by the equations (1) and (2)

$$S_i = a_i \oplus b_i \oplus c_{i-1} \tag{1}$$

$$c_i = a_i b_i + b_i c_{i-1} + a_i c_{i-1} \tag{2}$$

Where i be the bit position.

By utilizing this operation simple ripple carry adder (RCA) can be implemented, but it consumes more area, power and delay respectively. So to overcome those problems, a new HKSA is designed with PPA methodology as follows:

Step 1 Bit Preprocessing: this step involves creation of generate and propagate signals. According to prefix computation g_i (generate) and p_i (propagate) signals are defined by the following equations (3) and (4) respectively.

$$p_i = a_i \oplus b_i \qquad (3)$$

$$g_i = a_i b_i \qquad (4)$$

Propagation is used to generate the next stage sum outputs and generation is used for next stage carry output generation.

Step 2 Prefix Computation: HKSA construction depends on the notion of group carry propagate$P_{[i:k]}$ and group generate signals $G_{[i:k]}$. Group generate and group propagate signals are defined by the equations (5) and (6) respectively

$$G_{[i:k]} = \begin{cases} g_i & if\ i = k \\ G_{[i:j]} + P_{[i:j]}G_{[j-1:k]} & otherwise \end{cases} \qquad (5)$$

$$P_{[i:k]} = \begin{cases} p_i & if\ i = k \\ P_{[i:j]}P_{[j-1:k]} & otherwise \end{cases} \qquad (6)$$

To simplify the representation of G and P , an operator called as dotoperator represented by ' *' is introduced to create group generate and group propagate, and is defined by the equation (7) as

$$(G,P)_{[i:k]} = (G,P)_{[i:j]} * (G,P)_{[j-1:k]} \qquad (7)$$

Step 3: Post-processing: Post-processing step involves formation of carry and sum bits for each individual operand bit. The equations for $c_i$andS_i, are defined as per equations (8) and (9) respectively.

$$c_i = G_{[i:0]} \qquad (8)$$

$$S_i = p_i \oplus c_{i-1} \qquad (9)$$

The step 1 andstep 3 are fundamentallyperforms speed of operation because the simple Boolean operations are enough to implement them for each bit. But the intermediate step has much complex operations, so the performance of HKSA is more dependent on the step 2 respectively. To optimize this operation two properties are introduced to reduce the area,power and delay respectively.

1) The following equation(10) explains the associative property respectively.

$$(G,P)_{[h:j]} * (G,P)_{[j:k]} = (G,P)_{[h:i]} * (G,P)_{[i:k]} \qquad (10)$$

Where $h > i \geq j \geq k$

2) Idempotent property can be explained as given in equations (11) respectively

$$(G,P)_{[h:j]} * (G,P)_{[i:k]} = (G,P)_{[h:k]} \qquad (11)$$

The sub-terms can be easily calculatedby implementing the associative property at Prefix Computation step. By this property, the prefix operations performed in parallel manner. Idempotent property allows the overlapping of sub-terms, thus the sub-terms parallelization can be performed flexibly. Using these properties, by utilizing Directed Acyclic Graphs (DAG's) HKSA was designed.HKSA consist of three nodes namely whitecell, blackcell and gray cell as presented in figure 1.

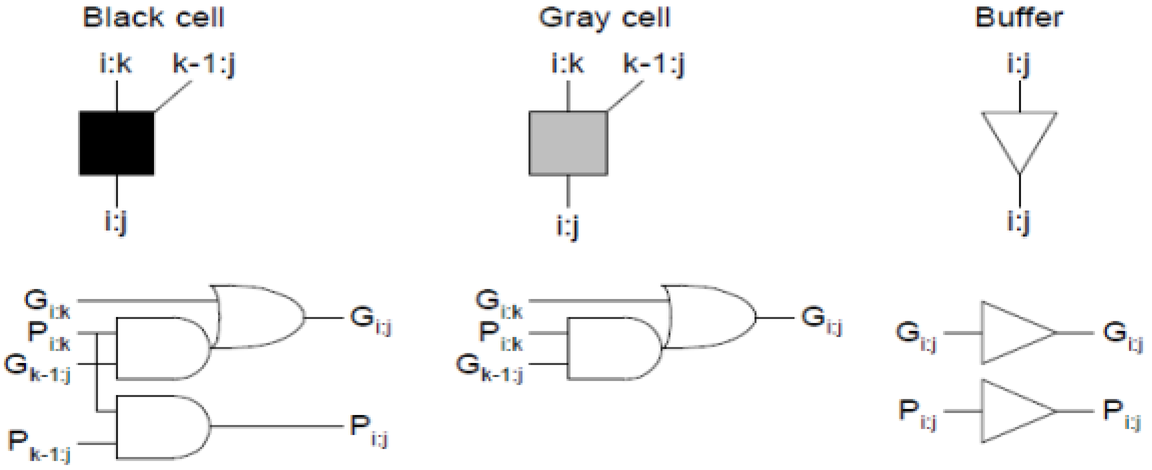


Figure 1: schematics of blackcell, gray cell and white cell

The operation of white cell simply acts as buffer. The gray cell operation given by equation 5 and gray cell operation given by equation 7 respectively, utilizing these three cells the HKSA is developed as shown in Figure 2.

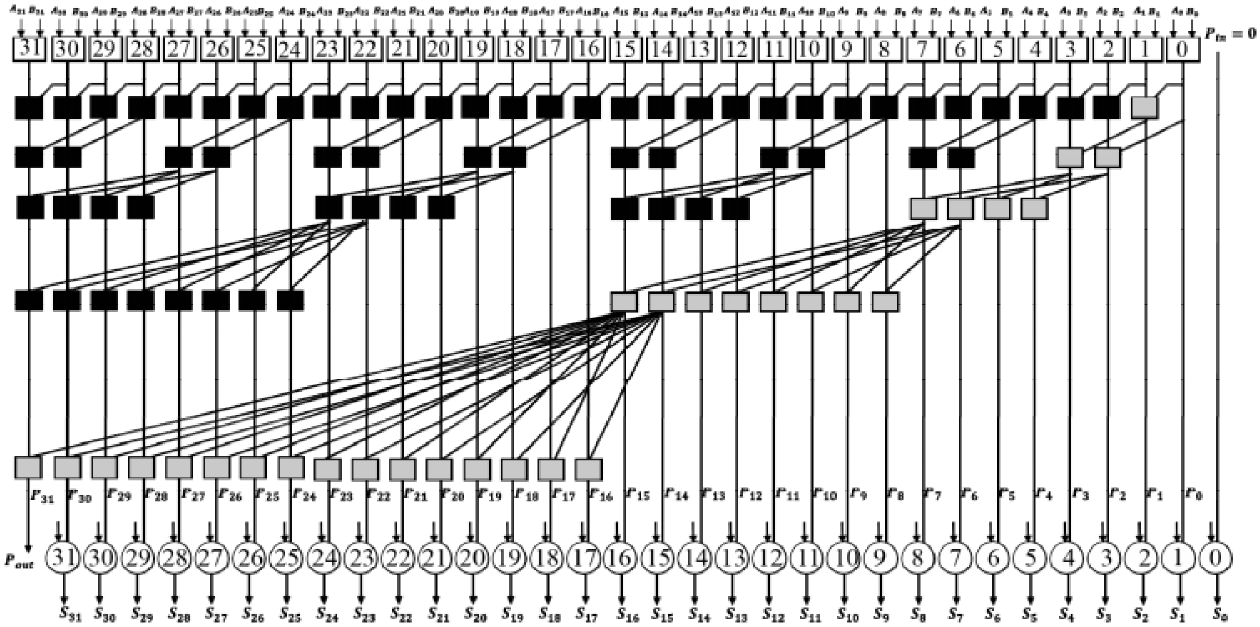

$P_{in} = 0$
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
r_{31} r_{30} r_{29} r_{28} r_{27} r_{26} r_{25} r_{24} r_{23} r_{22} r_{21} r_{20} r_{19} r_{18} r_{17} r_{16} r_{15} r_{14} r_{13} r_{12} r_{11} r_{10} r_9 r_8 r_7 r_6 r_5 r_4 r_3 r_2 r_1 r_0
P_{out}
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
S_{31} S_{30} S_{29} S_{28} S_{27} S_{26} S_{25} S_{24} S_{23} S_{22} S_{21} S_{20} S_{19} S_{18} S_{17} S_{16} S_{15} S_{14} S_{13} S_{12} S_{11} S_{10} S_9 S_8 S_7 S_6 S_5 S_4 S_3 S_2 S_1 S_0

Fig 2: architecture of 32-bit HKSA

HKSA performs initially data bit preprocessing operation, so g_i (generate) and p_i (propagate) signals for every bits. Further parallel prefixes are computed using prefix computation stage, there are five levels. The blackcells count will be reducedand gray cells count will be increased by square manner.HKSAoperation performed using the divide and conquer concept. In this adder design, group of two PG signals compute the partial-Carry/Carry signal, and this is further used to find the prefix of the 4-bit group and so forth until the last bit is reached, i.e., log2 n stage. These prefixes are then fed back into the adder to compute the carry of each bit. The advantage of HKSA is that the fan-out of each dot is two. The quine computational complexity $Q_{HKSA}$and gate count $C_{HKSA}$for the proposed HKSA is given by equation 12 and 13 respectively.

$$C_{HKSA} = \left[\frac{3n}{2}(log_2 n + 1)\right] + 2n - 2 \qquad (12)$$

$$Q_{HKSA} = 3n(log_2 n + 1) + 4n - 4 \qquad (13)$$

An n bit BKA takes a time (T_{HBKA}), as the summation of time taken by the Prefix Network (T_{PG}), $(2(\log_2 n) - 1)$stages of the Carry Network(T_{carry}), and the Sum Network (T_{sum}).

$$T_{HKSA} = T_{PG} + (2(log_2 n) - 1)T_{carry} + T_{sum} \qquad (14)$$

A32-bit HKSA needs, 2 units delay for the PG Network, 10 units delay for the Carry Network and the 2 units delay for the Sum Network. In total, it takes $14\Delta g$. Table 1 compares the different parameters with respect to the conventional KSA to the proposed HKSA. Similarly by extending the cells the bit length of adders can be extended to 64-bit, 128-bit….and so on. From the table 2it confirms that the proposed HKSA consumes the low area and delay requirements.

Table 2: comparison over HKSA over KSA

parameter	Conventional KSA	HKSA
Gate count	$3n(log_2 n) + 4$	$\left[\frac{3n}{2}(log_2 n + 1)\right] + 2n - 2$
Quine complexity	$6n(log_2 n) - 2n + 8$	$3n(log_2 n + 1) + 4n - 4$
Time complexity	$T_{PG} + (log_2 n)T_{carry} + T_{sum}$	$T_{PG} + (2(log_2 n) - 1)T_{carry} + T_{sum}$

3. MULTIPLIER USING HKSA

The multiplier is the essential building block in the DSP processor DIP processor, ALUs and all the types of integrated circuits. Thus the efficient design of multiplier makes the design to area, delay and power efficient. The design of N bit multiplier presented in the figure 3.

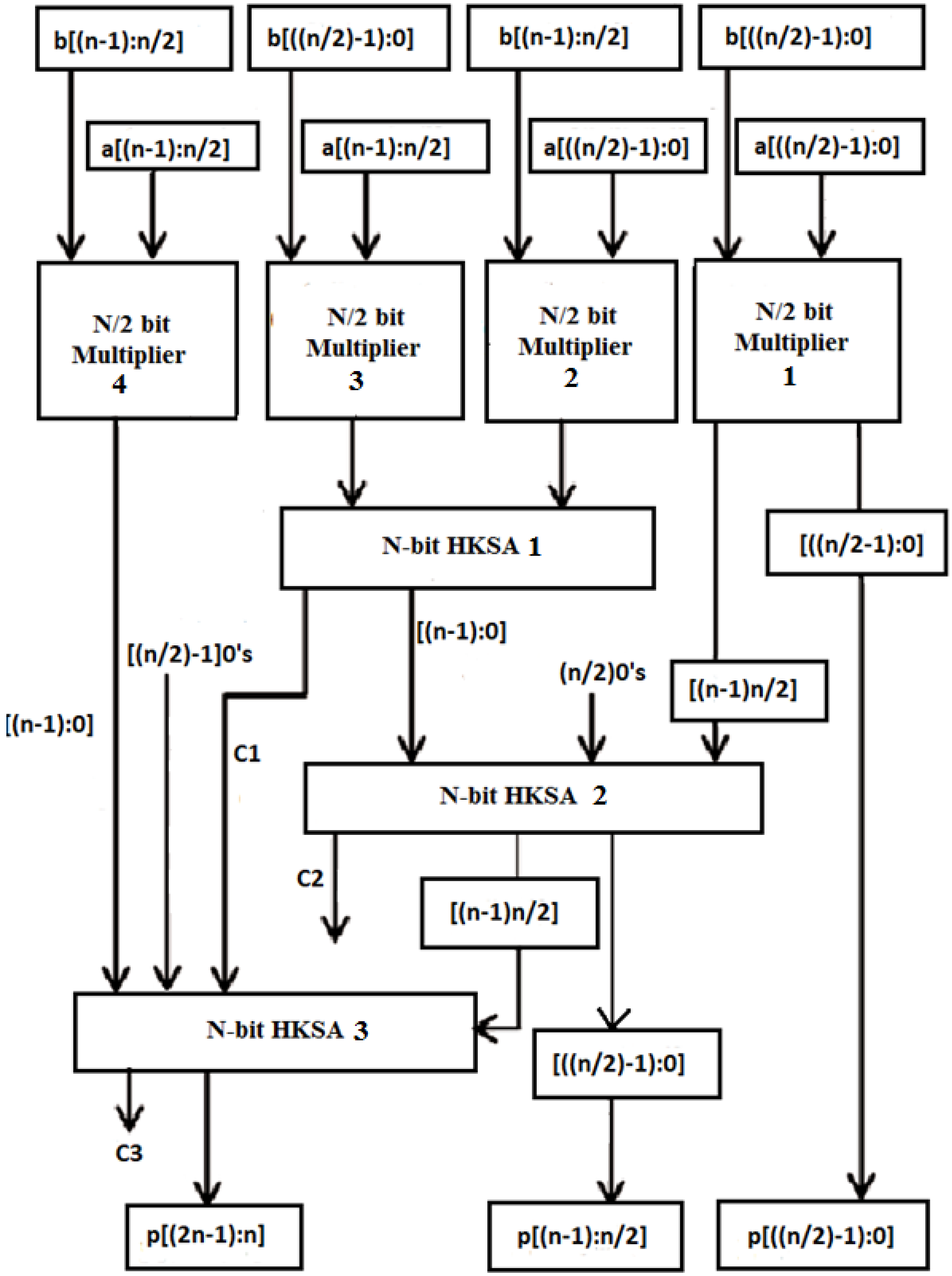

b[(n-1):n/2]
b[((n/2)-1):0]
b[(n-1):n/2]
b[((n/2)-1):0]
a[(n-1):n/2]
a[(n-1):n/2]
a[((n/2)-1):0]
a[((n/2)-1):0]
N/2 bit Multiplier 4
N/2 bit Multiplier 3
N/2 bit Multiplier 2
N/2 bit Multiplier 1
N-bit HKSA 1
[((n/2-1):0]
[(n-1):0]
[(n/2)-1]0's
[(n-1):0]
(n/2)0's
[(n-1)n/2]
C1
N-bit HKSA 2
C2
[(n-1)n/2]
N-bit HKSA 3
[((n/2)-1):0]
C3
p[(2n-1):n]
p[(n-1):n/2]
p[((n/2)-1):0]

Fig 3: Proposed N-bit Multiplier using HKSA

The operation of N-bit multiplier as follows

Step 1: Divide the inputs$A$ and B into four groups as mentioned in figure.

Step 2: perform the multiplication operation on each group; here the proposed multiplier will be iteratively utilized. For an instance N-bit multiplier contains$N/2$ bit multiplier and $N/2$ bit multiplier contains$N/4$ bit multiplier and it goes on respectively.

Step 3:for first HKSA 1 inputs will be outputs generated from $N/2$ bit multiplier 2 and $N/2$ bit multiplier3 respectively.

Step 4:the sum output from HKSA 1 andsecond half output bits of $N/2$ bit multiplier 1 applied as inputs to the HKSA 2 respectively.

Step 5: the second half sum outputs from HKSA 2, output bits of $N/2$ bit multiplier 4 and caary out of HKSA 3 applied as inputs to the HKSA 3 respectively.

Step 6: first half output bits of $N/2$ bit multiplier 1, first half sum output bits of HKSA2 and all the sum bits of HKSA 3 directly reflected as the product output respectively.

4. PROPOSED HKSA BASED ALU

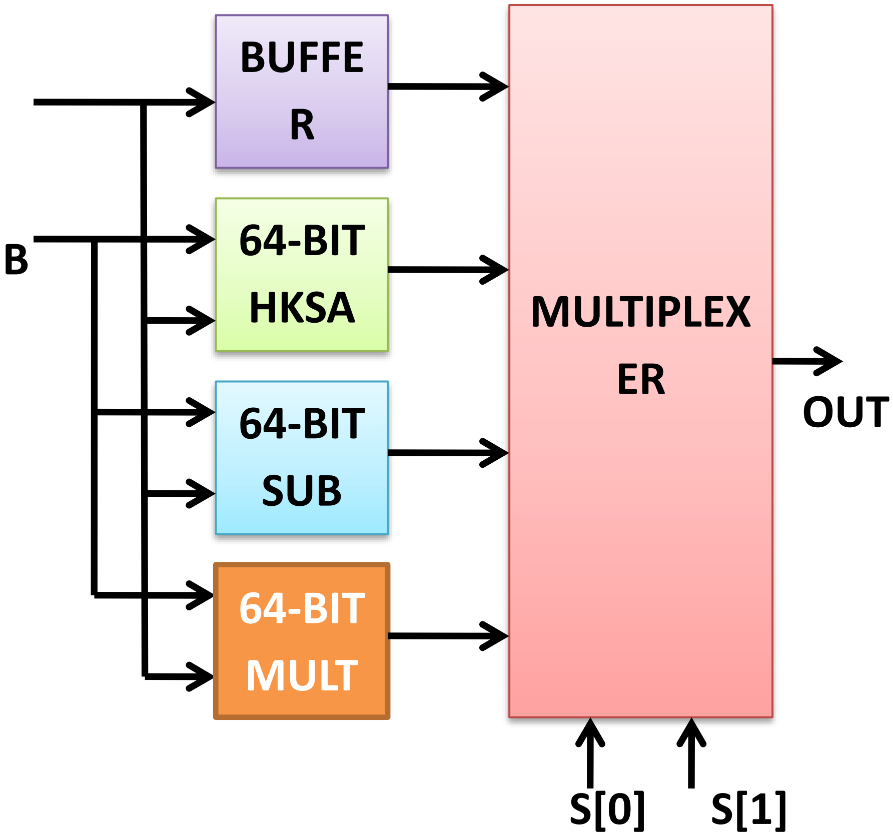


Fig 4: architecture of N-bit RALU

The ALU is the essential building block in the every DSP processor DIP processor, Intel processors and all the types of integrated circuits. Thus the efficient design of ALU makes the design to area, delay and power efficient. In the proposed method majorly focuses on design the N-bit ALU with combinations HKSA, The 64-bit ALU also utilizes the 64-bit HKSA, N-bit subtractor and 64-bit multiplier, thus the operation of each arithmetic unit explained in detail as shown in figure 4.

Table 3: ALU operation

S[1]	S[0]	OPERATION
0	0	BUFFER
0	1	ADDITION
1	0	SUBTRACTION

1	1	MULTIPLICATION

If selection lines of multiplexer is 2’b00, and then multiplexer selects output as buffered A. if selection lines of multiplexer is 2’b01, multiplexer selectsthe HKSA addition asits output. The operation of adder explained detailed in section2.If selection lines of multiplexer is 2’b10, multiplexer selects the subtractor output respectively. Here for implementing the subtraction operation, again the HKSA utilized with twos complement operation. By making the B input to complement and by apply the cin as one to HKSA, it will acts HKSA based subtractor respectively.

$$diff = A - B = A + \bar{B} + 1 \qquad (15)$$

When selection lines of multiplexer is 2’b11, and then multiplexer selects the output as multiplication operation. The operation of adder explained detailed in section 3.

5. SIMULATION RESULTS:

All the proposed designs have been programmed and designed using Xilinx ISE software this software tool provides the two categories of outputs named as simulation and synthesis. The simulation results give the detailed analysis of proposed design with respect to inputs, output byte level combinations. Through simulation analysis of accuracy of the addition, multiplication process estimated easily by applying the different combination inputs and by monitoring various outputs. Through the synthesis results the utilization of area with respect to the programmable logic blocks (PLBs), look up tables (LUT) will be achieved. And also time summary with respect to various path delays will be obtained and power summary generated using the static and dynamic power consumed.

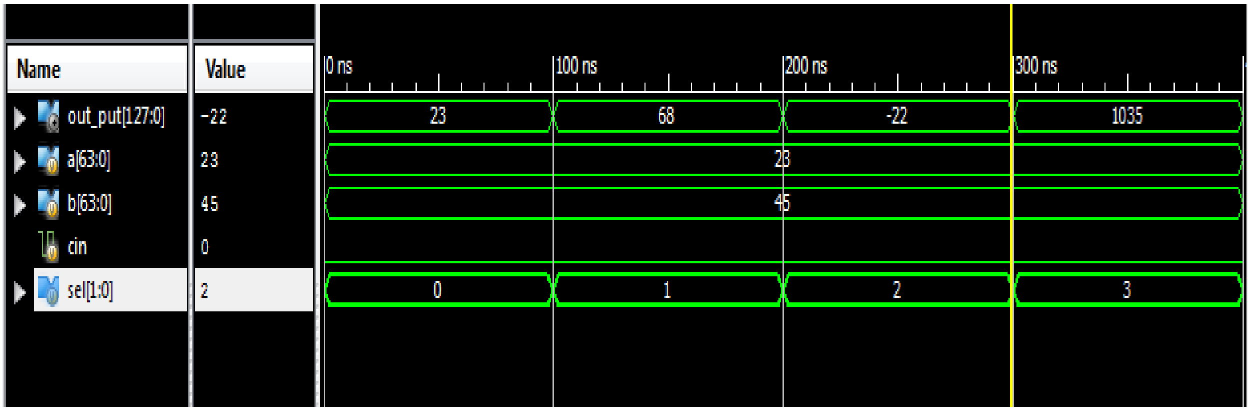
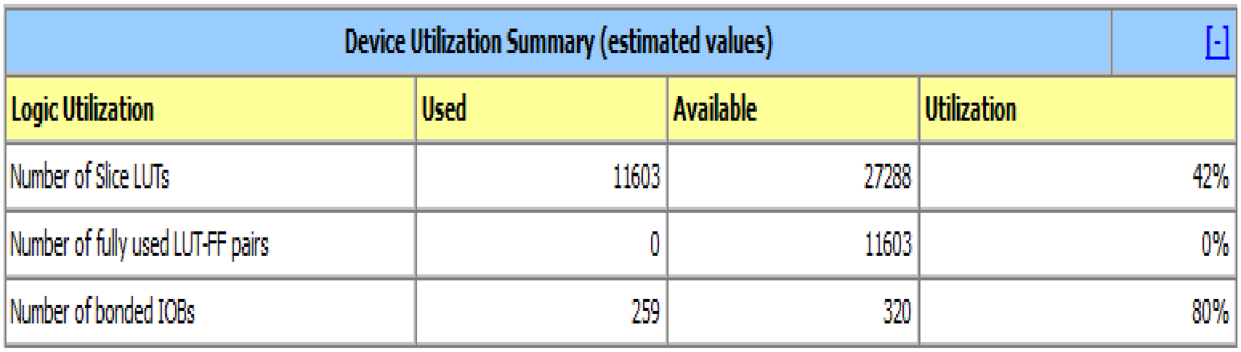


Fig 5: simulation output

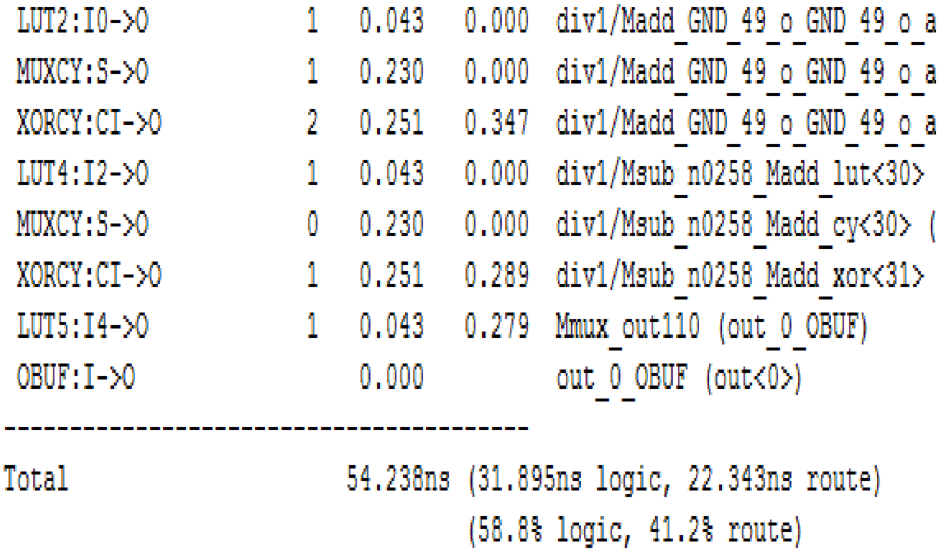
The above result represents the simulation waveform by using the Xilinx ISE software. Where sel is the selection line, if sel is 0 then a {23}buffered and generates the output as 23, s is 1 then a,b {23,45}added together and generates the output as 68, s is 2 then a,b {23,45}subtracted and generates the output as -22 and s is 3 then a,b {23,43} multiplied together and generates the output as 1035 respectively.



Device Utilization Summary (estimated values)			[-]
Logic Utilization	**Used**	**Available**	**Utilization**
Number of Slice LUTs	11603	27288	42%
Number of fully used LUT-FF pairs	0	11603	0%
Number of bonded IOBs	259	320	80%

Fig 6: design summary

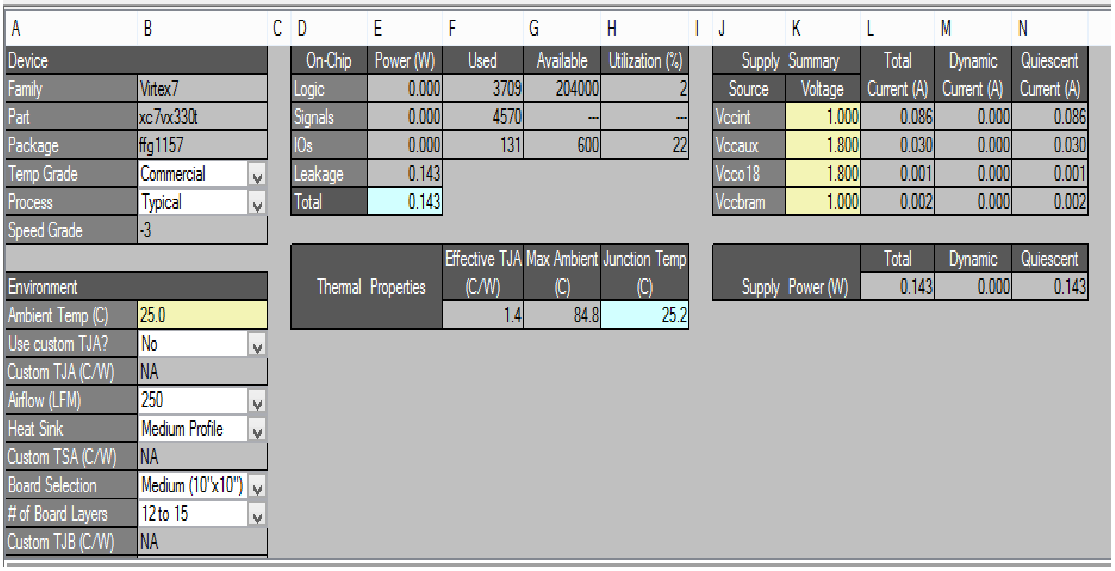
The above result represents the synthesis implementation by using the Xilinx ISE software. From the above table, it is observed that only 11603 look up tables are used out of available 27288. It indicates less areawas used for the proposed design.



```
 LUT2:I0->O           1   0.043   0.000  div1/Madd_GND_49_o_GND_49_o_a
 MUXCY:S->O           1   0.230   0.000  div1/Madd_GND_49_o_GND_49_o_a
 XORCY:CI->O          2   0.251   0.347  div1/Madd_GND_49_o_GND_49_o_a
 LUT4:I2->O           1   0.043   0.000  div1/Msub_n0258_Madd_lut<30>
 MUXCY:S->O           0   0.230   0.000  div1/Msub_n0258_Madd_cy<30> (
 XORCY:CI->O          1   0.251   0.289  div1/Msub_n0258_Madd_xor<31>
 LUT5:I4->O           1   0.043   0.279  Mmux_out110 (out_0_OBUF)
 OBUF:I->O                0.000          out_0_OBUF (out<0>)
----------------------------------------
Total                    54.238ns (31.895ns logic, 22.343ns route)
                                  (58.8% logic, 41.2% route)
```

Fig 7: Time summary

The above result represents the time consumed such as path delays by using the Xilinx ISE software.the consumed path delay is 54.238ns.



Device	
Family	Virtex7
Part	xc7vx330t
Package	ffg1157
Temp Grade	Commercial
Process	Typical
Speed Grade	-3

Environment	
Ambient Temp (C)	25.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250
Heat Sink	Medium Profile
Custom TSA (C/W)	NA
Board Selection	Medium (10"x10")
# of Board Layers	12 to 15
Custom TJB (C/W)	NA

On-Chip	Power (W)	Used	Available	Utilization (%)
Logic	0.000	3709	204000	2
Signals	0.000	4570	---	---
IOs	0.000	131	600	22
Leakage	0.143			
Total	0.143			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	1.4	84.8	25.2

Supply Summary Source	Voltage	Total Current (A)	Dynamic Current (A)	Quiescent Current (A)
Vccint	1.000	0.086	0.000	0.086
Vccaux	1.800	0.030	0.000	0.030
Vcco18	1.800	0.001	0.000	0.001
Vccbram	1.000	0.002	0.000	0.002

	Total	Dynamic	Quiescent
Supply Power (W)	0.143	0.000	0.143

Fig 8: Power summary

The above result represents the power consumed by using the Xilinx ISE software. The consumed power is 0.143uw.From the table 4, it is observed that the proposed method consumes very less area compared to the conventional approaches such as HSALU [11], FTALU [9] and RALU [10] respectively.

Table 4: comparison table

parameter	HSALU [11]	FTALU [9]	RALU [10]	**PROPOSED METHOD**
Time delay	63.13 ns	66.10 ns	59.110 ns	**54.238 ns**
Power utilized	2.364 uw	2.12uw	1.293uw	**0.143 uw**
Look up tables	28108	26037	21029	**11603**
Area utilized	92%	93%	87%	**42%**

CONCLUSION

In this paper an area efficient PPA was designed, utilizing this PPAand64-bit HKSA was developed. And utilizing these HKSAs an N-bit multiplier was developed. Finally 64-bit ALU developed by utilizing the architectures of adder, subtractor and multiplier. The64-bit ALU has design with low hardware resources utilization. The simulation and synthesis results shows the proposed method has area, power and delay efficient compared to this state of art Approaches and many literatures. This work can be extended to implement the N-bit sequential logical units such as shift register, counters and so on thus they can be effectively used in every DSP processor and design of any IC achieved through low power, low delay and area efficient .

REFERENCES:

[1]Zulehner, Alwin, and Robert Wille. "One-pass design of reversible circuits: Combining embedding and synthesis for reversible logic." IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 37.5 (2017): 996-1008.

[2]Raveendran, Sithara, et al. "Design and Implementation of Reversible Logic based RGB to Gray scale Color Space Converter." *TENCON 2018-2018 IEEE Region 10 Conference*. IEEE, 2018.

[3]Qian, Junzhou, and Junchao Wang. "A 4-bit array multiplier design by reversible logic." *Information Technology*. CRC Press, 2015. 21-24.

[4]Ghosh, Arpita, and S. K. Sarkar. "Performance Investigation of Nanoscale Reversible Logic gates designed with SE-TLG Approach." *International Journal of Electronics* just-accepted (2020).

[5]Gaur, Hari Mohan, Ashutosh Kumar Singh, and Umesh Ghanekar. "In-depth comparative analysis of reversible gates for designing logic circuits." Procedia Computer Science 125 (2018): 810-817.

[6]Sasamal, Trailokya Nath, Ashutosh Kumar Singh, and Anand Mohan. "Reversible logic circuit synthesis and optimization using adaptive genetic algorithm." *Procedia Computer Science* 70 (2015): 407-413.

[7]Ahmad, Peer Zahoor, et al. "A novel reversible logic gate and its systematic approach to implement cost-efficient arithmetic logic circuits using QCA." *Data in brief* 15 (2017): 701-708.

[8].Yugandhar, K., et al. "High Performance Array Multiplier using Reversible Logic Structure." *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*. IEEE, 2018.

[9]Kamaraj, A., and P. Marichamy. "Design of integrated reversible fault-tolerant arithmetic and logic unit." *Microprocessors and Microsystems* 69 (2019): 16-23.

[10]Amirthalakshmi, T. M., and S. Selvakumar Raja. "Design and analysis of low power 8-bit ALU on reversible logic for nanoprocessors." *Journal of Ambient Intelligence and Humanized Computing* (2018): 1-19.

[11]Dasharatha, M., et al. "VLSI Design and Synthesis of Reduced Power and High Speed ALU Using Reversible Gates and Vedic Multiplier." *Advances in Decision Sciences, Image Processing, Security and Computer Vision*. Springer, Cham, 2020. 272-280.

[12] Rahim, B. Abdul, et al. "Design of a Power Efficient ALU Using Reversible Logic Gates." International Conference on Communications and Cyber Physical Engineering 2018. Springer, Singapore, 2018.

[13]Oskouei, Saeed Mirzajani, and Ali Ghaffari. "Designing a new reversible ALU by QCA for reducing occupation area." *The Journal of Supercomputing* 75.8 (2019): 5118-5144.

[14]Shukla, Vandana, et al. "Reversible Realization of N-bit arithmetic circuit for low power loss ALU applications." *Procedia Computer Science* 125 (2018): 847-854.

[15]Ahmad, Peer Zahoor, et al. "A novel reversible logic gate and its systematic approach to implement cost-efficient arithmetic logic circuits using QCA." *Data in brief* 15 (2017): 701-708.

[16]Schumann, Andrew. "Reversible logic gates on Physarum polycephalum." AIP Conference Proceedings. Vol. 1648. No. 1. AIP Publishing LLC, 2015.

[17]Das, Biplab, Avijit Kumar Paul, and Debashis De. "An unconventional Arithmetic Logic Unit design and computing in Actin Quantum Cellular Automata." *Microsystem Technologies* (2019): 1-14.

[18]Sadhu, Tapatosh, et al. "XOR transform with actin quantum automata for better security against cryptanalysis." *Microsystem Technologies* 25.5 (2019): 1705-1717.

[19]Nafees, Naira, et al. "Modeling and Logic Synthesis of Multifunctional and Universal 3× 3 Reversible Gate for Nanoscale Applications." *International Conference on Intelligent Computing and Smart Communication 2019*. Springer, Singapore, 2020.

[20]Singh, Snigdha, Abhinay Choudhary, and Manoj Kumar Jain. "A Brief Overview of Reversible Logic gate and Reversible Circuits." *International journal of Electronics Engineering* 11.2 (2019): 86-104.