Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Exploring the impact of initial design techniques on area, timing, and power in technology mapped designs: A case study on 32-bit arithmetic logic unit

Hammad H. Alshortan, Yasser Almalaq, Muhammad Imran Khan *

Department of Electrical Engineering, College of Engineering, University of Hail, Hail, Saudi Arabia

ARTICLE INFO

Article history:
Received 27 March 2023
Received in revised form
31 July 2023
Accepted 3 August 2023

Keywords:
Initial design techniques
Technology mapped designs
Arithmetic logic unit
Ripple carry method
Sklansky method

ABSTRACT

This research paper investigates the influence of different initial design techniques on the area, timing, and power aspects of technology-mapped designs. As a practical case study, we undertake the design and analysis of a 32-bit arithmetic logic unit (ALU) utilizing two distinct adder approaches. The ALU, a fundamental component of all processors, comprises three major units: the Adder responsible for signed and unsigned number addition and subtraction, the Logic unit which handles bitwise logical operations, and the Shifter unit facilitates arithmetic and logical shift operations. The two adder designs are based on the ripple carry method (ALU_RCA) and the Sklansky method (ALU_SKL), respectively. The design and analysis process involved utilizing established toolsets from Cadence, including NCSIM for simulation and verification, RTL Compiler for logic synthesis, static timing analysis and power estimation, and SOC encounter tool for floorplanning and layout. Through this investigation, we aim to shed light on the varying performance implications of different initial design approaches in technology-mapped designs.

1. Introduction

The fundamental objective of this scholarly work is to deliberate upon the ramifications of initial design choices on the dimensions of area, timing, and power consumption in the context of Application-Specific Integrated Circuit (ASIC) designs. Furthermore, this exposition aims to provide a comprehensive overview of the ASIC design workflow, spanning from its inception with Register-Transfer Level (RTL) design to the conclusive phases of floorplanning and layout synthesis. Of paramount significance in the realm of processor architecture, the Arithmetic Logic Unit (ALU) assumes a pivotal role. In this study, meticulous attention has been devoted to the development and VHDL coding of a 32-bit ALU. The research endeavor was inaugurated with the conceptualization and realization of the 32-bit ALU, tailored for integration within a processor architecture, as illustrated in Figs. 1 and 2 (Alshortan et al., 2021; Alrashdi and Khan, 2022; Durrani et al., 2016; Dossis, 2015).

Fig. 2 presents an intricate exposition encompassing both the detailed specification and the design schematic of a 32-bit ALU, which has been developed in accordance with the provided specifications. The design encapsulated within Fig. 2 is composed of three discrete functional blocks: firstly, an adder/subtractor module; secondly, a bitwise operations module; and lastly, a shifter module. The adder module itself incorporates two distinct implementations, namely the Ripple carry and Sklansky configurations. This ALU design conforms to the conventional architecture requisite for integration within a 32-bit processor. It encompasses two 32-bit input data signals, denoted as 'a' and 'b,' an additional 32-bit output data signal ('out'), a 4-bit operation code ('op') signal, as well as the imperative reset and clock signals governing dynamic timing. It is essential to underscore that the ALU inputs and outputs adhere strictly to positive edge-triggered flip-flop behavior. The ALU, as envisioned, comprises four primary functional blocks: arithmetic, logic, shift, and multiplexer. The transition of data through these functional stages occurs on each positive edge of the clock signal (CLK). The operands 'A' and 'B,' together with the operation code, initially populate the input flip-flop

* Corresponding Author.
Email Address: mu.khan@uoh.edu.sa (M. I. Khan)

https://doi.org/10.21833/ijaas.2023.09.008

Corresponding author's ORCID profile:
https://orcid.org/0000-0002-7844-9567

unit before proceeding to engage with the three functional blocks. It merits mention that the ALU design incorporates a deliberate two-clock cycle delay to optimize its functionality. To validate the VHDL code's adherence to the anticipated behavior of the 32-bit ALU, rigorous simulations and verification procedures were conducted employing the Cadence simulation tool, specifically NCSIM (Gan et al., 2015; Huang et al., 2018; Han, 2013; Iannacci, 2021; Jha et al., 2014; Khan, 2023; Khan et al., 2021).

2. Synthesis

The essence of the synthesis process lies in the transformation of VHDL code into an RTL netlist, which assumes the format of standard cells. It is imperative to note that all the simulations showcased within this scholarly work were executed using the Cadence platform, employing the 130nm process design kit as the foundation. In the context of our 32-bit ALU design, the initial code was expressed in VHDL. However, it is noteworthy that Cadence predominantly operates with Verilog. Consequently, the primary step executed by the RTL compiler entailed the translation of the VHDL code into a Verilog netlist. Concurrently, Cadence undertook a critical verification process to ascertain the synthesizability of the code. This verification step holds paramount importance, as only synthesizable code can be effectively mapped onto the designated process technology. Subsequent to this pivotal phase, a static timing analysis (STA) was conducted to assess the temporal performance characteristics of the design. It is crucial to underscore that static timing analysis delves into the timing performance of the design without necessitating simulation, providing invaluable insights into the design's temporal behavior (Khan and Lin, 2014a; 2014b; Khan et al., 2018a; Li et al., 2023; Marculescu et al., 1998).

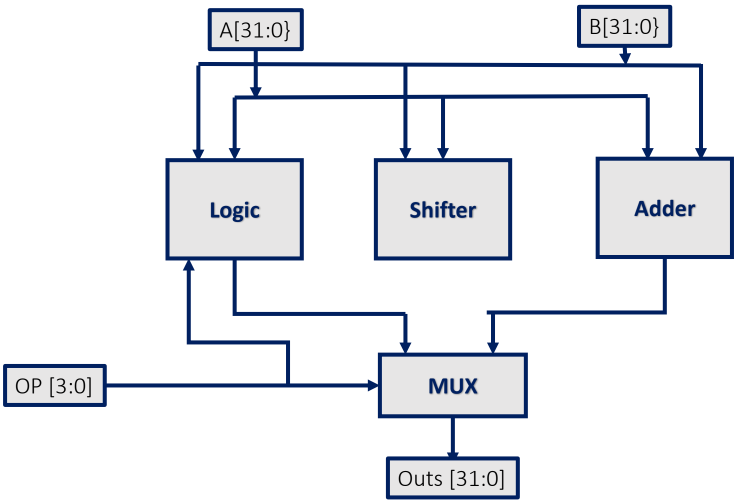

Fig. 1: A simple block-level diagram of 32-bit ALU

```
entity ALU is
port(
Clk : in std_logic;
Reset : in std_logic;
A : in std_logic_vector(31 downto 0);
B : in std_logic_vector(31 downto 0);
Op : in std_logic_vector(3 downto 0);
Outs : out std_logic_vector(31 downto 0));
end ALU;
* 0000: add A+B (signed)
* 0001: add A+B (unsigned)
* 0010: sub A-B (signed)
* 0011: sub A-B (unsigned)
* 0100: bitwise AND
* 0101: bitwise OR
* 0110: bitwise NOR
* 0111: bitwise XOR
* 1000: shift left
* 1001: shift right (logical)
* 1010: shift right (arithmetical, signed)
* for all other Op codes a zero output vector should be given
```

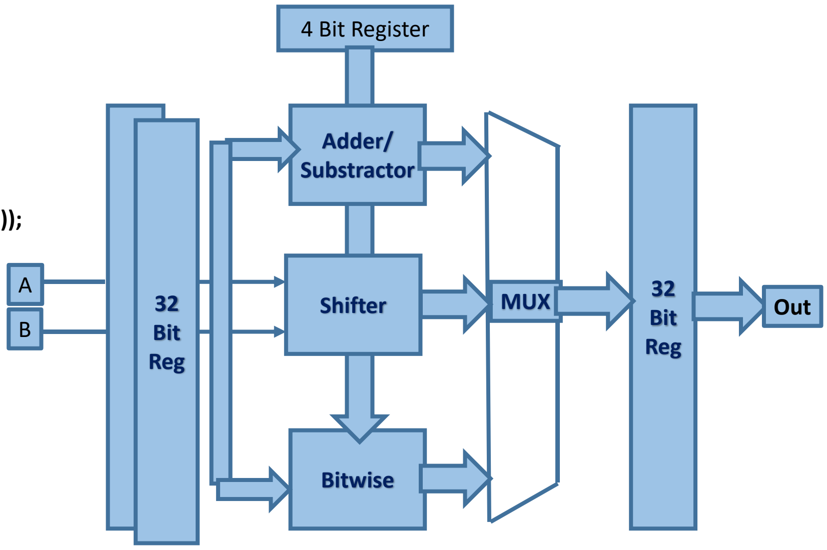

Fig. 2: Arithmetic logic unit (ALU) specifications and simple architecture

3. Ripple-ALU

The first design was ripple carry adder ALU_RCA. We synthesized our design without any timing constraints with low effort. Table 1 shows the worst-case delay and estimated area results with different synthesis timing constraints.

From Table 1, we observe that the ripple-adder meets the timing constraints in the case of 2681ps but it failed in the case of 1250ps. Fig. 3 shows the worst-case path of ALU_RCA.

By checking the worst-case path shown in Fig. 3, it is observed that the worst-case path of ALU_RCA is composed of all stages of the ripple-adder component, regardless of the timing constraint.

Table 1: ALU implementation with ripple carry adder

Timing constraint (ps)	Worst case delay (ps)	Estimated area (μm^2)
Unconstraint	4464	22909
1250	1250	15540

4. Sklansky-ALU

Table 2 shows the worst-case delay and estimated area results with different synthesis timing constraints using the Sklansky adder.

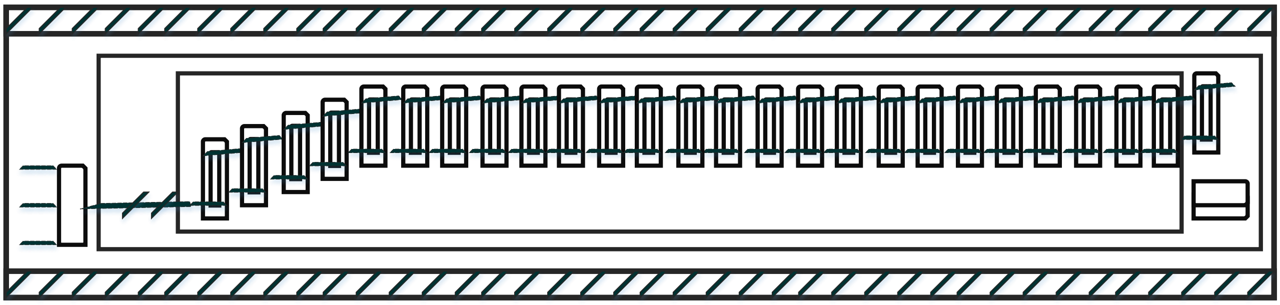

Fig. 3: The worst-case path of ALU_RCA

Table 2: ALU implementation with Sklansky adder

Timing constraint (ps)	Worst case delay (ps)	Estimated area μm^2
Unconstraint	5223	21721
2683	2681	14521
1250	2096	15203

Table 3 shows the results of power analysis using ripple carry adder and the Table 4 represents the results of power analysis using Sklansky adder. All these results were obtained using a Cadence 130nm process technology design kit. Fig. 4 shows the worst-case path of the Sklansky adder. We observed that the shifter path has taken the place of the adder path to constitute the worst-case path, i.e. the adder cell has changed to a non-dominant factor in terms of circuit delay, which can be demonstrated by the 10_critical_path report we acquired during the simulation. The entire first three critical paths belong to the shifter block, and then the arithmetic block and the shifter block (Khan et al., 2018b; 2017; Merkel, 2018; Markov et al., 2015; Nouaiti et al., 2019; O'Dare and Arslan, 1994).

Table 3: ALU_RCA power report

Worst delay [ps]	Leakage power (0.02) (μw)	Dynamic power (0.02) (μw)	Total power (0.02) (μw)	Leakage power (0.1) (μw)	Dynamic power (0.1) (μw)	Total power (0.1) (μw)
2200	416568	2215903	2632472	416329	5004717	5421046
2400	371298	1958989	2330288	371509	4582199	4953708
2600	360711	1823009	2183721	360611	4447638	4808250

Table 4: ALU_SKL power report

Worst delay [ps]	Leakage power (0.02) (μw)	Dynamic power (0.02) (μw)	Total power (0.02) (μw)	Leakage power (0.1) (μw)	Dynamic power (0.1) (μw)	Total power (0.1) (μw)
2097	355049	1881255	2236304	355291	4233334	4588626
2381	338584	1789888	2128472	338780	4185260	4524040
2580	337865	1692527	2030392	338331	4045702	4384033

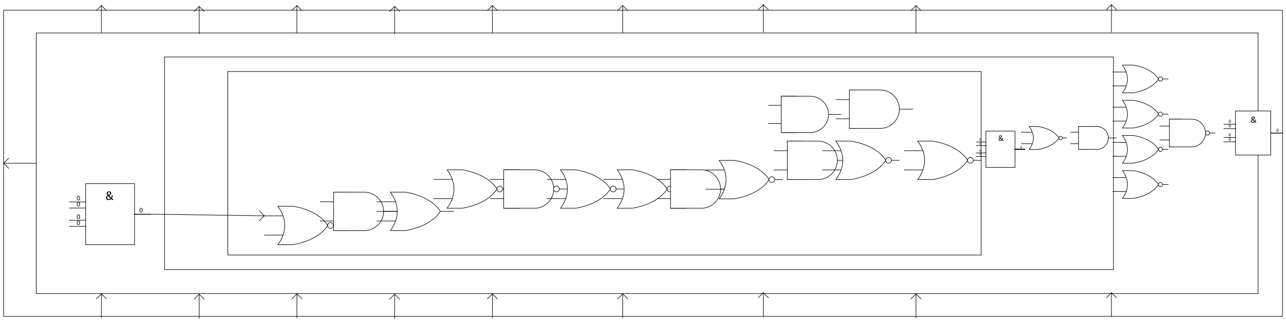

Fig. 4: The worst-case path of ALU_SKL

Fig. 5 presents a visual juxtaposition of the implementation area when employing both the ripple carry adder and the Sklansky adder under varying timing constraints. In a complementary manner, Fig. 6 provides an analytical examination of power consumption, elucidating a comparative assessment between the ripple carry and Sklansky ALU across different timing constraints.

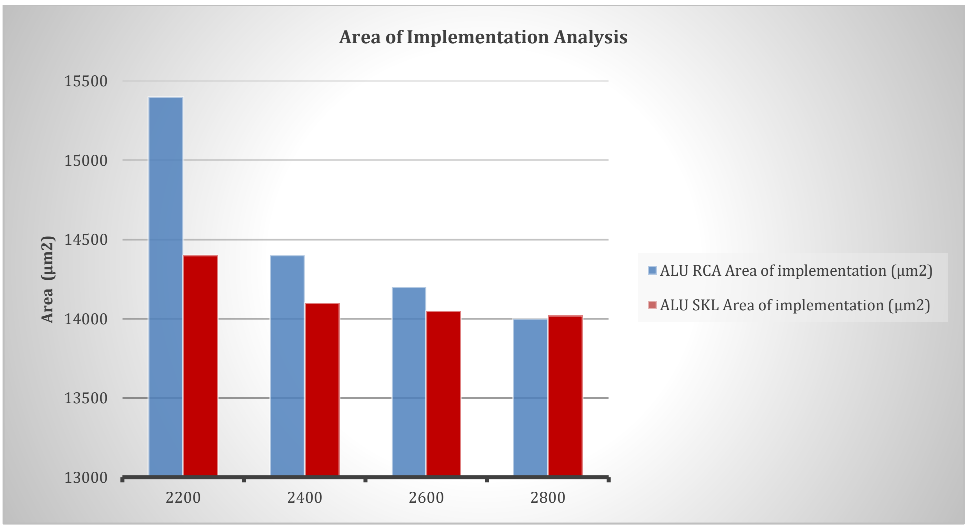

Fig. 5: Area of implementation of ALU-RCA and ALU-SKL

It is interesting to compare how the area scales with stricter timing constraints between the ALU_RCA and ALU_SKL. According to the results obtained, both ALU_RCA and ALU_SKL sacrifice the area utilization to shorten the worst-case delay. Meanwhile, under the same timing constraints, the area of the former is larger than the latter and the ratio of them tends to decrease with the timing constraint rising from 2200ps to 2800ps.

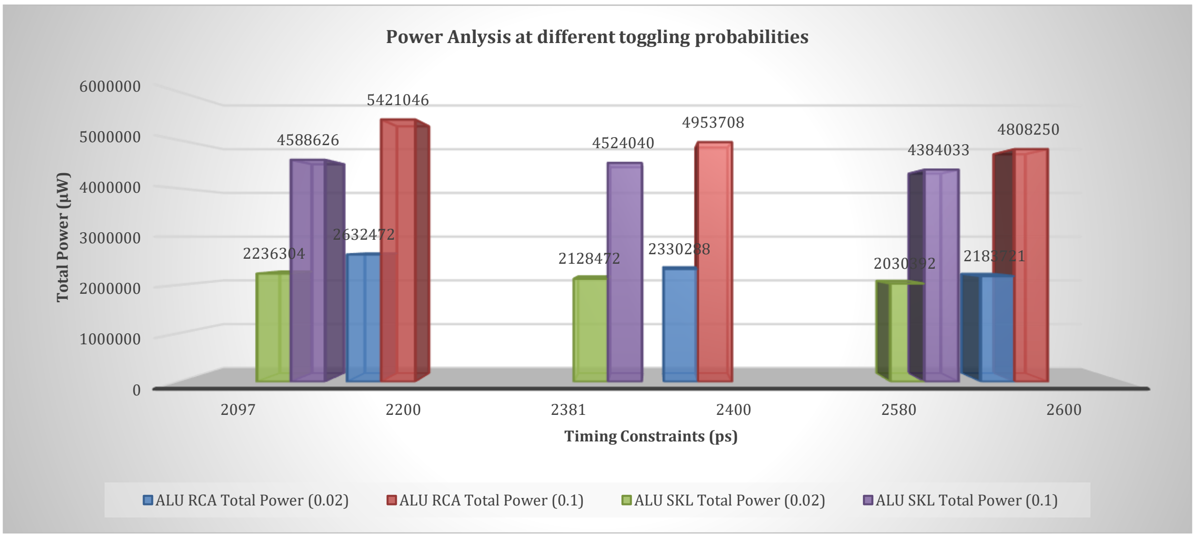

Fig. 6: Power analysis comparison of ALU_RCA and ALU_SKL

5. Power analysis

In the ensuing analysis, we embark upon a comparative evaluation between ALU_RCA and ALU_SKL with respect to power consumption, subject to varying timing constraints. Specifically, we have set the input probability for the high logic state at a fixed value of 0.5, while delineating the toggling probability at two distinct levels, namely 0.02 and 0.1. Tables 5 and 6, upon initial examination, reveal two salient observations. Firstly, regardless of the chosen ALU architecture, there exists a positive correlation between the stringency of the imposed timing constraints and the magnitude of power dissipation. In other words, stricter timing constraints result in greater power dissipation. Secondly, when operating under identical timing constraints, it becomes evident that the ALU associated with an input signal exhibiting a toggling probability of 0.1 dissipates nearly twice as much power as its counterpart group. This discrepancy can be attributed to the fundamental principle that the toggling probability bears a direct proportionality to dynamic power, which, in turn, constitutes the predominant component of the overall power consumption (Khan et al., 2014a; 2014b; Pedroni, 2020; Buzdar et al., 2017a; 2017b). From Fig. 6, it is obvious that the power dissipation of ALU_RCA is larger than that of ALU_SKL in case of the same values of timing constraint and the toggling probability. Furthermore, the ratio of them tends to get more and closer to one, i.e. the ALU_SKL gradually takes less advantage over the ALU_RCA in terms of power efficiency. In order to get more insight into this trend, more simulations and analyses have been done to generate Tables 5 and 6. Table 6 shows that ALU_SKL dissipates more power than the other structure under the condition of relaxed timing constraints. However, it is usually meaningless to operate ALU_SKL with much-relaxed timing constraints, and hence in most cases, ALU_RCA does not compete with the ALU_SKL in terms of power. Figs. 7 and 8 show the power dissipation of ALU_RCA and ALU_SKL using different timing constraints. Fig. 9. shows the Power comparison between ALU_RCA and ALU_SKL on different timing constraints (Salman et al., 2009; Stojanovic and Oklobdzija, 1999; Scheffer et al., 2006).

From Table 6 It can be illustrated that the power dissipation of the random vectors is much larger than the other types of test vectors because the dynamic power of the random vector is the largest of all, which can be attributed to its high toggling rate. After we got verified and synthesized netlists, the designs would be taken into place and route. Cadence Soc Encounter was used to implement in this stage and an already finished netlist was used. The general place and route flow and optimization in each step as well as the CAD tool was reviewed (Simicic et al., 2018; Shoukat and Khan 2018a; 2018b; Thomas, 2023; Tang and Yao, 2007; Choi and Swartzlander, 2008).

Table 5: Power analysis on the relaxed timing constraint

Worst delay [ps]	Ripple (0.02) (μw)	Skl (0.02) (μw)	Ripple (0.1) (μw)	Skl (0.1) (μw)
4300	1582673	1630714	3868661	3998876
3800	1792065	1739224	4450330	4185087

Table 6: The comparison of the power dissipation between the test vectors with different characters

Worst delay [ps]	Leakage power (μw)	Dynamic power (μw)	Total power (μw)
Random	582896	462392	1045289
Regular	550044	171421	721466
Realtrace	551901	201363	753264

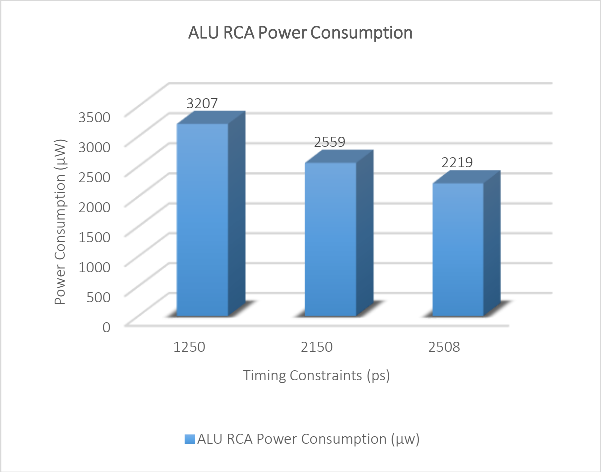

Fig. 7: Power dissipation of ALU_RCA using different timing constraints

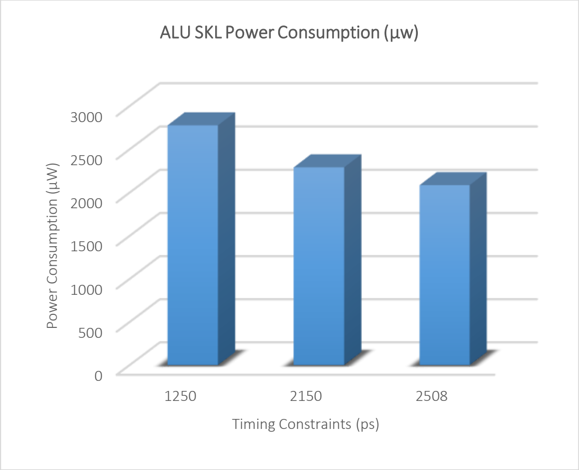

Fig. 8: Power dissipation of ALU_SKL using different timing constraints

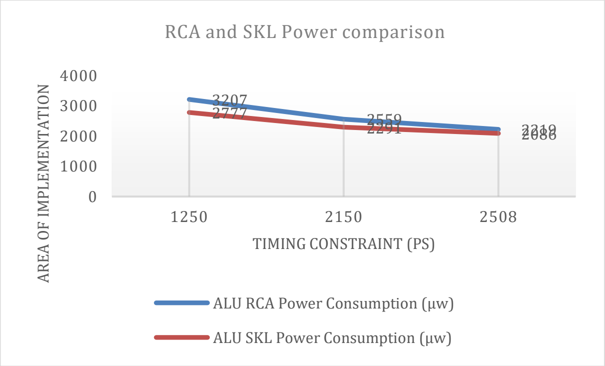

Fig. 9: Power comparison between ALU_RCA and ALU_SKL

6. Place and route

Following the successful verification and synthesis of netlists, we transitioned to the "place and route" phase, facilitated by the Cadence SOC Encounter tool. SOC Encounter served as the platform for executing floor planning, standard cell placement, and routing tasks. Notably, floor planning entails the strategic determination of the spatial arrangement of various design blocks on the chip, with a profound emphasis on achieving a well-optimized partitioning of the design. In our initial endeavor, as delineated in Table 7, we adhered to the floor plan illustrated in Fig. 10. Subsequent to the floor planning stage, we proceeded with the requisite ancillary tasks, encompassing pin placement, power grid routing, standard cell placement, clock tree synthesis, and routing. Regrettably, our initial attempt failed to conform to the stringent timing constraint of 3.2 ns, evidenced by the presence of negative timing slack, signifying non-compliance with the stipulated temporal requirements. Within this initial attempt, four violations surfaced, primarily attributable to geometric infringements, specifically the existence of minimum cuts. These infractions were diligently addressed, while no process-related violations were detected. In our second endeavor, as detailed in Table 8, we adopted a distinct approach by relocating all components beyond the confines of the designated box. This approach obviated the need for explicit partitioning and manual pin placement, delegating these tasks to the tool's automated decision-making process concerning the placement of standard cells. Remarkably, this second approach yielded significantly improved results, with timing slack registering at just -0.270 ns, denoting a notable enhancement compared to our initial attempt (Table 7).

Table 7: First attempt at optimization

	Pre CTS optimization	Post CTS optimization	Routing	Post routing	Final timing
Required time (ns)	3.138	3.141	3.416	3.138	3.138
Arrival time (ns)	4.084	4.361	4.314	4.084	4.084
Time slack (ns)	-0.945	-0.947	-0.947	-0.945	-0.945

Table 8: Second attempt at optimization

	Pre CTS optimization	CTS	Post CTS optimization	Routing	Post routing	Final timing
Required time (ns)	3.135	3.455	3.447	3.464	3.467	3.467
Arrival time (ns)	3.948	4.119	3.770	3.766	3.737	3.737
Time slack (ns)	-0.8413	-0.864	-0.323	-0.302	-0.270	-0.270

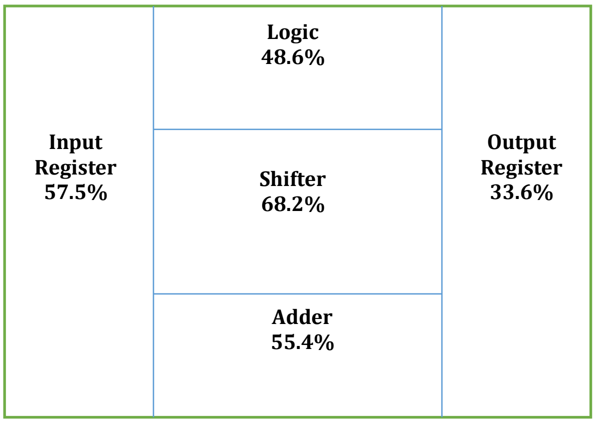

Fig. 10: Logic partitioning during place and route

7. Conclusion

This article's research entails the realization of a 32-bit ALU employing two distinct adder architectures: the ripple carry adder and the Sklansky adder. Our investigation involves a comprehensive comparison of the outcomes achieved through these two different adder implementations, with particular emphasis on diverse timing constraints. The synthesis results revealed a discernible disparity in the dynamic evolution of the area occupied by the ripple carry adder in contrast to the Sklansky adder. This discrepancy arises from the inherent necessity of the ripple carry adder to expend more resources in order to comply with more stringent timing constraints, albeit at the cost of increased spatial occupancy. Conversely, the ALU founded upon the Sklansky adder exhibited commendable performance, effortlessly meeting the stringent timing constraints without necessitating significant expansions in area or augmentations in power consumption. Intriguingly, our analysis also discerned that the ALU_RCA exhibited a relatively smaller footprint and lower power consumption when juxtaposed against the ALU_SKL. Consequently, it becomes evident that, under conditions where timing constraints are not overly demanding, the utilization of the ALU_RCA proves to be a more efficacious design choice in terms of optimizing area and power consumption.

Acknowledgment

This research has been funded by the Scientific Research Deanship at the University of Ha'il–Saudi Arabia through project number GR-22 106."

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alrashdi A and Khan MI (2022). Design and comparative analysis of high speed and low power ALU using RCA and Sklansky adders for high-performance systems. Engineering, Technology and Applied Science Research, 12(2): 8426-8430. https://doi.org/10.48084/etasr.4817

Alshortan HH, Alogla A, Eleiwa MA, and Khan MI (2021). Low cost high gain 8×8 planar array antenna for 5G applications at 28GHz. Engineering, Technology and Applied Science Research, 11(6): 7964-7967. https://doi.org/10.48084/etasr.4609

Buzdar AR, Sun L, Azhar MW, Khan MI, and Rao K (2017a). Area and energy efficient Viterbi accelerator for embedded processor datapaths. International Journal of Advanced Computer Science and Applications, 8(3): 402-407. https://doi.org/10.14569/IJACSA.2017.080355

Buzdar AR, Sun L, Rao K, Azhar MW, and Khan MI (2017b). Cyclic redundancy checking (CRC) accelerator for embedded processor datapaths. International Journal of Advanced Computer Science and Applications, 8(2): 321-325. https://doi.org/10.14569/IJACSA.2017.080242

Choi Y and Swartzlander EE (2008). Speculative carry generation with prefix adder. IEEE Transactions on Very Large Scale

Integration (VLSI) Systems, 16(3): 321-326. https://doi.org/10.1109/TVLSI.2007.915502

Dossis M (2015). High-level synthesis integrated verification. Engineering, Technology and Applied Science Research, 5(5): 864-870. https://doi.org/10.48084/etasr.596

Durrani YA, Riesgo T, Khan MI, and Mahmood T (2016). Power analysis approach and its application to IP-based SoC design. COMPEL-The International Journal for Computation and Mathematics in Electrical and Electronic Engineering, 35(3): 1218-1236. https://doi.org/10.1108/COMPEL-08-2015-0283

Gan Z, Salman E, and Stanaćević M (2015). Figures-of-merit to evaluate the significance of switching noise in analog circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 23(12): 2945-2956. https://doi.org/10.1109/TVLSI.2014.2386315

Han D (2013). Comparison of commonly used image interpolation methods. In the 2nd International Conference on Computer Science and Electronics Engineering, Atlantis Press, Hangzhou, China: 1556-1559. https://doi.org/10.2991/iccsee.2013.391

Huang D, Yang X, Chen H, Khan MI, and Lin F (2018). A 0.3–3.5 GHz active-feedback low-noise amplifier with linearization design for wideband receivers. AEU-International Journal of Electronics and Communications, 84: 192-198. https://doi.org/10.1016/j.aeue.2017.12.003

Iannacci J (2021). The WEAF Mnecosystem: A perspective of MEMS/NEMS technologies as pillars of future 6G, tactile internet and super-IoT. Microsystem Technologies, 27: 4193–4207. https://doi.org/10.1007/s00542-021-05230-3

Jha AK, Kumar A, Schaefer G, and Ahad MAR (2014). An efficient edge preserving image interpolation algorithm. In the International Conference on Informatics, Electronics and Vision, IEEE, Dhaka, Bangladesh: 1-4. https://doi.org/10.1109/ICIEV.2014.6850820 **PMCid:PMC3952416**

Khan MI (2023). Harmonic estimation and comparative analysis of ultra-high speed flip-flop and latch topologies for low power and high performance future generation micro-/nano electronic systems. ACM Transactions on Design Automation of Electronic Systems, 28(4): 1-17. https://doi.org/10.1145/3590770

Khan MI and Lin F (2014a). Impact of transistor model accuracy on harmonic spectra emitted by logic circuits. In the 12th IEEE International Conference on Solid-State and Integrated Circuit Technology, IEEE, Guilin, China: 1-3. https://doi.org/10.1109/ICSICT.2014.7021316

Khan MI and Lin F (2014b). Comparative analysis and design of harmonic aware low-power latches and flip-flops. In the IEEE International Conference on Electron Devices and Solid-State Circuits, IEEE, Chengdu, China: 1-2. https://doi.org/10.1109/EDSSC.2014.7061282

Khan MI, Alshammari AS, Alshammari BM, and Alzamil AA (2021). Estimation and analysis of higher-order harmonics in advanced integrated circuits to implement noise-free future-generation micro-and nanoelectromechanical systems. Micromachines, 12(5): 541. https://doi.org/10.3390/mi12050541 **PMid:34068549 PMCid:PMC8151574**

Khan MI, Dong H, Shabbir F, and Shoukat R (2018b). Embedded passive components in advanced 3D chips and micro/nano electronic systems. Microsystem Technologies, 24: 869-877. https://doi.org/10.1007/s00542-017-3586-3

Khan MI, Qamar A, Shabbir F, and Shoukat R (2017). Design, development and implementation of a low power and high speed pipeline A/D converter in submicron CMOS technology. Microsystem Technologies, 23: 6005-6014. https://doi.org/10.1007/s00542-017-3550-2

Khan MI, Shoukat R, Mukherjee K, and Dong H (2018a). Analysis of harmonic contents of switching waveforms emitted by the ultra-high speed digital CMOS integrated circuits for use in future micro/nano systems applications. Microsystem Technologies, 24: 1201-1206. https://doi.org/10.1007/s00542-017-3486-6

Li R, Yang Z, and Cai B (2023). Design and test of a linear micro-motion stage with adjustable stiffness and frequency. Microsystem Technologies, 29(3): 377-385. https://doi.org/10.1007/s00542-023-05433-w

Marculescu R, Marculescu D, and Pedram M (1998). Probabilistic modeling of dependencies during switching activity analysis. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 17(2): 73-83. https://doi.org/10.1109/43.681258

Markov IL, Hu J, and Kim MC (2015). Progress and challenges in VLSI placement research. In Proceedings of the IEEE, 103(11):1985–2003. https://doi.org/10.1109/JPROC.2015.2478963

Merkel C (2018). Device solutions to scientific computing. Nature Electronics, 1(7): 382-383. https://doi.org/10.1038/s41928-018-0108-y

Nouaiti A, Saad A, Mesbahi A, Khafallah M, and Reddak M (2019). Design and test of a new three-phase multilevel inverter for PV system applications. Engineering, Technology and Applied Science Research, 9(1): 3846-3851. https://doi.org/10.48084/etasr.2573

O'Dare MJ and Arslan T (1994). Generating test patterns for VLSI circuits using a genetic algorithm. Electronics Letters, 30(10): 778-779. https://doi.org/10.1049/el:19940524

Pedroni VA (2020). Circuit design with VHDL. MIT Press, Cambridge, USA.

Salman E, Jakushokas R, Friedman EG, Secareanu RM, and Hartin OL (2009). Methodology for efficient substrate noise analysis in large-scale mixed-signal circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 17(10): 1405-1418. https://doi.org/10.1109/TVLSI.2008.2003518

Scheffer L, Lavagno L, and Martin G (2006). EDA for IC system design, verification, and testing. Taylor and Francis Group, Boca Raton, USA.

Shoukat R and Khan MI (2018a). Design and development of a clip building block system for MEMS. Microsystem Technologies, 24: 1025-1031. https://doi.org/10.1007/s00542-017-3453-2

Shoukat R and Khan MI (2018b). Nanotechnology based electrical control and navigation system for worm guidance using electric field gradient. Microsystem Technologies, 24: 989-993. https://doi.org/10.1007/s00542-017-3444-3

Simicic M, Weckx P, Parvais B, Roussel P, Kaczer B, and Gielen G (2018). Understanding the impact of time-dependent random variability on analog ICs: From single transistor measurements to circuit simulations. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 27(3): 601-610. https://doi.org/10.1109/TVLSI.2018.2878841

Stojanovic V and Oklobdzija VG (1999). Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems. IEEE Journal of Solid-State Circuits, 34(4): 536-548. https://doi.org/10.1109/4.753687

Tang M and Yao X (2007). A memetic algorithm for VLSI floorplanning. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 37(1): 62-69. https://doi.org/10.1109/TSMCB.2006.883268 **PMid:17278559**

Thomas S (2023). Chips that feel the heat. Nature Electronics, 6(3): 179-179. https://doi.org/10.1038/s41928-023-00946-8