

Design and Comparative Analysis of High Speed and Low Power ALU Using RCA and Sklansky Adders for High-Performance Systems

Ali Fraih Alrashdi

Department of Electrical Engineering
University of Hail
Hail, Saudi Arabia
gnem.h2003@gmail.com

Muhammad Imran Khan

Department of Electrical Engineering
University of Hail
Hail, Saudi Arabia
imran@alumni.chalmers.se

Received: 9 February 2022 | Revised: 28 February 2022 | Accepted: 1 March 2022

Abstract—This study examines how different initial design decisions affect the area, timing, and power of technology-mapped designs. ASIC design flow, tools used during the flow, and the factors to consider to maximize the performance and power ratio are discussed. The ALU (Arithmetic Logic Unit) is a fundamental part of all processors. In this study, two ALUs were implemented using two different types of adder circuits: a Ripple Carry Adder (RCA) and a Sklansky adder. The Cadence EDA tools were used for the implementation. A comparative analysis was conducted for the two designed ALUs in terms of area, power, and timing analysis. The ALU design was also used as an example to examine the whole workflow front-end wise by constructing a block schematic and back-end wise by floorplanning, placing, and routing the physical design.

Keywords—arithmetic logic unit (ALU); ripple carry adder; sklansky adder; VHDL

I. INTRODUCTION

The basic purpose of this study was to examine the impact of initial design decisions on the area, timing, and power of ASIC designs and to introduce the ASIC design flow from the beginning to the end, i.e. from RTL design to floorplanning and layout. The Arithmetic Logic Unit (ALU) is a key element of a processor that performs arithmetic and logical operations on binary numbers [1-7]. Several 32-bit ALUs have been designed and coded in VHDL [8-13]. An ALU consists of three major parts: Adder, logic unit, and shifter. The adder is responsible for the subtraction and addition operations of signed and unsigned numbers. The logic unit is responsible for bitwise logical operations, and the shifter unit is responsible for the arithmetic and logical shift operations, as shown in Figure 1 [14-20]. In the first ALU design, a Ripple Carry Adder (RCA) was used, while in the second one, a prefix tree of Sklansky-type adder was used. The adder is a very important component in digital systems, and in the past, much research has been conducted on various adder types to improve their speed and area requirements [21-25]. Apart from the adder, both ALU types consisted of Logical and Shifter blocks.

The design in Figures 1 and 2 has three main blocks: adder/subtractor, bitwise operations, and shifter. The adder block has two different implementations: A ripple carry and a Sklansky type. The ALU registers in data (A/B), opcode (Op), and output were positively edge-triggered. The design was implemented using two approaches. The first approach, shown in Figure 2, has all the functional blocks that compute the operations and the mux at the output decides which functional block is connected to the output. The other approach intended to guard the individual functional blocks using a mux after the input registers. This input mux intended that if the opcode is not meant for the particular functional block, it will get all zeroes as data input, reduce the combinatorial switching, and hence decrease power consumption. But then, the first design approach was chosen as including six 32-bit muxes would take up too much area and would probably kill the basic idea of saving power consumption. [25-30].

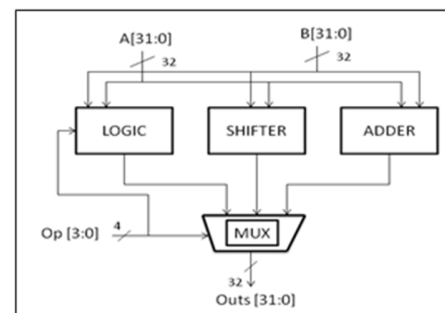


Fig. 1. Block diagram of a 32-bit ALU

II. SIMULATIONS AND VERIFICATION

After completing the design in VHDL, the functionality of the ALU design was verified using a test bench and test vectors. The test bench was also designed in VHDL as a non-synthesizable code, while the rest of the ALU code was synthesizable. The test bench was designed to be able to read the test vectors and send them as input to the 32-bit ALU. The test bench compared the outputs received from the ALU to the

Corresponding author: Muhammad Imran Khan

expected outputs. As these test vectors and expected outputs were a design specification, running them on the test bench was a functional verification. Functional verification of RCA and SKL was performed by the Cadence simulation tool called NCSIM. Three different sets of test vectors were used to verify the ALUs: 1001 random test vectors, 1001 regular test vectors, and 154999 real trace test vectors. These test vectors were not enough to provide the full test coverage, however 268 combinations of test vectors can be a reliable number.

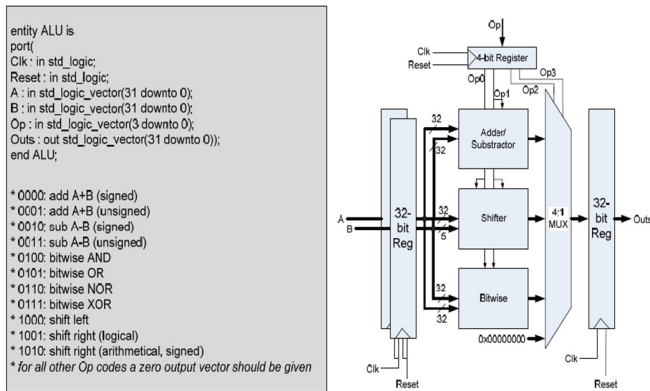


Fig. 2. Specifications of the ALU.

III. SYNTHESIS

After the successful simulation and verification of the design, the next step was to synthesize it. Logic Synthesis is the process to convert HDL code into RTL netlist in terms of standard cells. A 130nm process technology was used for the standard cell library. The RTL compiler tool generated the netlist at the end of synthesis in the form of Verilog code. Opening this Verilog code revealed that it was an entirely structural code with hardware description in terms of connections between standard cells. After making sure that the ALU design was synthesizable, Static Timing Analysis was used to analyze the timing performance of the design without simulating the design.

The ALU RCA design was synthesized using various timing constraints and Tables I-II show the results of those attempts. The RCA-based ALU did not meet the timing constraint for an 800MHz processor. The best timing was achieved by the RCA-based ALU at around 2250ps (444 MHz). The critical path of the RCA-based ALU laid in the RCA block for all the above implementations, though the exact path changed as stricter timing constraints were introduced. But for the most part, the path consisted of input carry rippling through the circuit from the LSB to the MSB.

TABLE I. STA RESULTS FOR ALU RCA

Effort	Time constraint (ps)	Worst case delay (ps)	Area of implementation (μm^2)
Low	None	4369	12619
Medium	2689	2688	12482
Medium	1250	2000	13906
Medium	2250	2250	13049

The SKL based ALU was able to meet the timing constraint for an 800MHz processor and was even able to clock up to 1GHz, although it pays in terms of area to reach this frequency. One major difference between the RCA and SKL designs was in terms of the critical path. For stricter timing constraints, the RCA had its critical path lying in the adder block, whereas the SKL had it in the shifter block. This is understandable since the Sklansky adder is much faster compared to the RCA. Examining the 10 worst paths showed that the critical paths for the SKL design were evenly distributed between the shifter and adder blocks. [31-35].

TABLE II. STA RESULTS FOR ALU SKL

Effort	Time constraint (ps)	Worst case delay (ps)	Area of implementation (μm^2)
Low	None	4528	13960
Medium	2500	2480	13283
Medium	1250	1250	16069
Medium	1000	1000	18185

IV. COMPARISON

For both RCA and SKL adders, the power at 0.1 toggling probability is greater than at 0.02, i.e. $\text{Power}(0.1) > \text{Power}(0.02)$. The leakage power is almost the same at different toggling probabilities, but dynamic power increases with an increase in toggling probability, as Table III shows. This makes sense as increasing the toggling probability increases the switching activity of internal nodes and nuts, which results in increased power. SKL dissipates less power than RCA. The area of implementation depends on the effort used by the RTL compiler and the timing constraints. The structure of SKL is more regular compared to RCA because SKL has an improved structure. The clock capacitance of RCA is more than SKL. Figures 3 and 4 show the area of implementation of the RCA and SKL ALUs respectively.

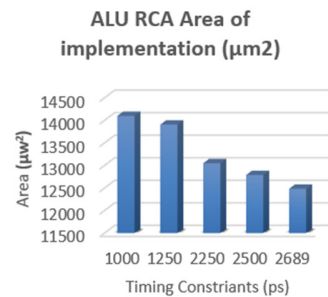


Fig. 3. ALU RCA area implementation (μm^2).

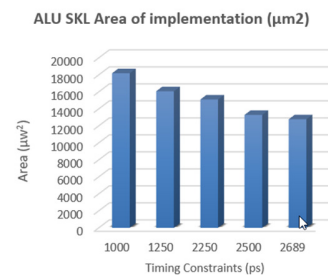


Fig. 4. ALU SKL area implementation (μm^2).

Figure 5 shows the area of implementation for both ALUs, while Figure 6 shows their power dissipation plots. Figure 7 shows a comparison of the efficiency of both ALUs in terms of power consumed under different timing constraints. RCA is more power-efficient than SKL. SKL increases its power efficiency in lower timing constraints. The choice of the input sequence is very important for power analysis. The input sequence can be chosen from real logic simulations. Three Value Change Dump (VCD) files were created for RCA, having random, regular, and real trace vectors as sets of input patterns. The VCD file gets real switching information in the power analysis and goes into the RTL compiler.

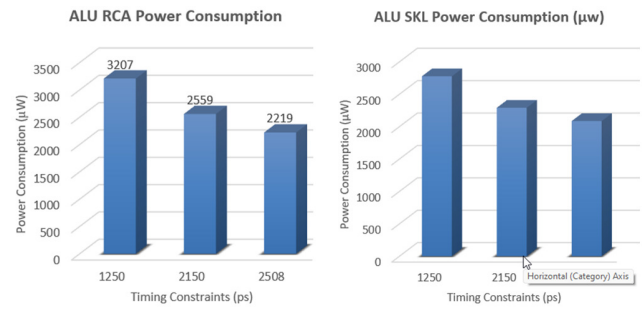


Fig. 6. Power dissipation of RCA and SKL ALUs using different timing constraints.

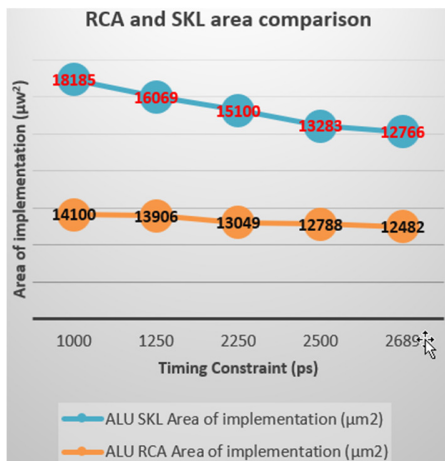


Fig. 5. Implementation area comparison of RCA and SKL ALUs.



Fig. 7. Power comparison between RCA and SKL ALUs.

TABLE III. POWER ANALYSIS OF SKL AND RCA BASED ON VARYING TOGGLING PROBABILITIES

Timing Constraint (ps)	Adder type	Toggling probability	Leakage power (μW)	Dynamic power (μW)	Total power (μW)	Clock power (μW)	Clock capacitance (fF)
Unconstraint	RCA	0.02	294	1334	1608	104	309.400
Unconstraint	RCA	0.1	294	3892	4186	104	309.400
Unconstraint	Sklansky	0.02	326	1760	2086	176	307.400
2508	Sklansky	0.1	326	4322	4648	176	307.400
2508	RCA	0.02	346	1873	2219	208	363.000
2508	RCA	0.1	346	4695	5041	208	363.000
2508	Sklansky	0.02	328	1963	2291	208	310.200
2150	Sklansky	0.1	328	4701	5029	208	310.200
2150	RCA	0.02	369	2190	2559	293	437.400
2150	RCA	0.1	369	5194	5563	293	437.400
1250	Sklansky	0.02	350	2857	3207	378	328.100

TABLE IV. POWER ESTIMATION BASED ON TEST VECTORS

Timing constraint (ps)	Adder type	Test vector type	Leakage power (nW)	Dynamic power (nW)	Total power (nW)
2508	RCA	Random	346558.721	6983081.572	7329640.292
2508	RCA	Regular	348672.104	36022780.146	3951452.250
2508	RCA	Real trace	351631.026	3954686.641	4306377.667
2508	Sklansky	Random	32364.014	6358256.700	6685620.713
2508	Sklansky	Regular	330858.982	3205877.702	3536070.684
2508	Sklansky	Real trace	340725.210	3516652.143	3857377.353
2150	RCA	Random	370941.482	8473036.411	8843977.893
2150	RCA	Regular	372972.466	4485655.551	4858628.017
2150	RCA	Real trace	373893.091	4747270.943	5121064.034
2150	Sklansky	Random	328880.228	7649393.717	7978273.945
2150	Sklansky	Regular	331153.317	4040497.554	4371650.871
2150	Sklansky	Real trace	335008.116	4308426.483	4643434.599

Table IV indicates that random vectors consume more power while the regular and real trace vectors consume less. The real trace vectors consume a little more power than the regular vectors. The flow for creating a use case power analysis was to synthesize the design with a specified timing constraint and compile cell library files along with the Verilog netlist of the synthesis and the testbench. A simulation was then run on a test vector set and signal switching activity was written into a VCD file. This file was compiled by an RTL compiler to obtain "real" switching information in the power analysis [28-30].

Comparing the power dissipation from the analyses made with random, regular, and real-trace test vectors showed that the random file is the most power-consuming, while the regular and the real trace are close, and the latter consumed a little more power. Random vectors were expected to consume more power. The real trace was expected to be the least power-consuming set but it seems that the regular set has more regularity and thus less switching than the real trace. Power consumption depends on the use of the design. Assuming that the design meets the power constraints, when it is only analyzed with a limited and perhaps irrelevant test vector set it can result in faulty conclusions. It is always good if real use cases can be tried, which is not always the case [35-40].

V. PLACE AND ROUTE

Finally, the design was examined through the back-end flow using the Cadence SoC Encounter. Floorplanning is the first step of the place and route flow. Floorplanning means specifying where to place different blocks of the design. In this step, it is important to make a good modular partitioning of the design. Also, since the critical path is in the RCA block, the maximum area should be provided in terms of utilization percentage. Once the floorplanning was complete, the necessary stages of physical design, such as pin placement, power grid routing, standard cell placement, clock tree synthesis, and routing, followed. The next step was the Clock Tree Synthesis (CTS) which was optimized in three steps: Pre-CTS, CTS, and Post-CTS. It is useful to perform a timing analysis between each optimization to see if the timing constraints are met.

The actual CTS step is like mapping the design to actual cells. The positions of clock buffers and clock tree were checked and it was found that the design had one level of buffers. The timing was checked again after this step but the constraint still wasn't met. The last step of CTS, i.e. post-CTS optimization, was followed to achieve the optimization based on the existing clock tree. The timing was checked again and it was found to be improved a lot, as the slack time was -0.466, much better than before when it was -2.259. Routing and post-route optimization were performed. The clock and reset signals should have the highest priority because they have to be provided to every block in the design and therefore are critical. Filler cells were used to fill the gaps and connect them to the power rails. Layout verification was performed. Four MinCut violations were found, which were later removed using the fixMinCutVia command. The final timing analysis showed a slack time at -0.395.

VI. CONCLUSION

An ALU design was drawn through the EDA flow from the basic idea and block schematic through RTL, verification, and back-end design with place and route. The synthesis results showed that the area of the RCA ALU changes more rapidly than that of the SKL ALU because the former has to put more effort to meet the stricter timing constraint at the expense of more area. The SKL ALU is a fast adder and easily meets the stricter timing constraint without increasing area and power consumption. It was also observed that the RCA ALU used less area and power compared to the SKL ALU, so it is better to use it if the timing constraint is not high, as it can be a more efficient design in terms of area and power consumption.

Furthermore, it was very instructive to see how different initial decisions affected the area, timing, and power of the technology mapped design. A well-structured design and the knowledge of the circuit help to make the most of the back-end design steps. This makes it easier to perform good floorplanning, helping the tools perform better at too complex tasks.

REFERENCES

- [1] M. D. Dean and K. Kockelman, "Our self-driving future will be shaped by policies of today," *Nature Electronics*, vol. 5, no. 1, pp. 2–4, Jan. 2022, <https://doi.org/10.1038/s41928-021-00708-4>.
- [2] F. F. dos Santos *et al.*, "Reduced Precision DWC: An Efficient Hardening Strategy for Mixed-Precision Architectures," *IEEE Transactions on Computers*, vol. 71, no. 3, pp. 573–586, Mar. 2022, <https://doi.org/10.1109/TC.2021.3058872>.
- [3] T. Nguyen and A. McCaskey, "Enabling Pulse-Level Programming, Compilation, and Execution in XACC," *IEEE Transactions on Computers*, vol. 71, no. 3, pp. 547–558, Mar. 2022, <https://doi.org/10.1109/TC.2021.3057166>.
- [4] L. Ye *et al.*, "The Challenges and Emerging Technologies for Low-Power Artificial Intelligence IoT Systems," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 12, pp. 4821–4834, Sep. 2021, <https://doi.org/10.1109/TCSI.2021.3095622>.
- [5] H. Asadi Dereshgi, H. Dal, and M. Z. Yildiz, "Piezoelectric micropumps: state of the art review," *Microsystem Technologies*, vol. 27, no. 12, pp. 4127–4155, Sep. 2021, <https://doi.org/10.1007/s00542-020-05190-0>.
- [6] M. Kim, M. Liu, L. R. Everson, and C. H. Kim, "An Embedded nand Flash-Based Compute-In-Memory Array Demonstrated in a Standard Logic Process," *IEEE Journal of Solid-State Circuits*, vol. 57, no. 2, pp. 625–638, Oct. 2022, <https://doi.org/10.1109/JSSC.2021.3098671>.
- [7] D. Huang, X. Yang, H. Chen, M. I. Khan, and F. Lin, "A 0.3–3.5 GHz active-feedback low-noise amplifier with linearization design for wideband receivers," *AEU - International Journal of Electronics and Communications*, vol. 84, pp. 192–198, Oct. 2018, <https://doi.org/10.1016/j.aeue.2017.12.003>.
- [8] A. Mirhoseini *et al.*, "A graph placement methodology for fast chip design," *Nature*, vol. 594, no. 7862, pp. 207–212, Jun. 2021, <https://doi.org/10.1038/s41586-021-03544-w>.
- [9] C. Merkel, "Device solutions to scientific computing," *Nature Electronics*, vol. 1, no. 7, pp. 382–383, Jul. 2018, <https://doi.org/10.1038/s41928-018-0108-y>.
- [10] I. L. Markov, J. Hu, and M. C. Kim, "Progress and Challenges in VLSI Placement Research," *Proceedings of the IEEE*, vol. 103, no. 11, pp. 1985–2003, Aug. 2015, <https://doi.org/10.1109/JPROC.2015.2478963>.
- [11] M. Tang and X. Yao, "A Memetic Algorithm for VLSI Floorplanning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 37, no. 1, pp. 62–69, Oct. 2007, <https://doi.org/10.1109/TSMCB.2006.883268>.

- [12] M. Simicic, P. Weckx, B. Parvais, P. Roussel, B. Kaczer, and G. Gielen, "Understanding the Impact of Time-Dependent Variability on Analog ICs: From Single Transistor Measurements to Circuit Simulations," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 3, pp. 601–610, Mar. 2019, <https://doi.org/10.1109/TVLSI.2018.2878841>.
- [13] M. I. Khan, A. S. Alshammari, B. M. Alshammari, and A. A. Alzamil, "Estimation and Analysis of Higher-Order Harmonics in Advanced Integrated Circuits to Implement Noise-Free Future-Generation Micro- and Nanoelectromechanical Systems," *Micromachines*, vol. 12, no. 5, May 2021, Art. no. 541, <https://doi.org/10.3390/mi12050541>.
- [14] J. Iannacci, "The WEAF Mnecosystem: a perspective of MEMS/NEMS technologies as pillars of future 6G, tactile internet and super-IoT," *Microsystem Technologies*, vol. 27, no. 12, pp. 4193–4207, Sep. 2021, <https://doi.org/10.1007/s00542-021-05230-3>.
- [15] V. A. Pedroni, *Circuit Design and Simulation with VHDL*, second edition. London, UK: MIT Press, 2010.
- [16] P. J. Ashenden, *The Designer's Guide to VHDL*. Burlington, MA, USA: Morgan Kaufmann, 2008.
- [17] J. M. Rabaey, A. P. Chandrakasan, and B. Nikolić, *Digital Integrated Circuits: A Design Perspective*. Upper Saddle River, NJ, USA: Pearson Education, 2003.
- [18] N. H. E. Weste, *CMOS VLSI Design: A Circuits and Systems Perspective*. Boston, MA, USA: Addison Wesley, 2011.
- [19] L. Scheffer, L. Lavagno, and G. Martin, *EDA for IC System Design, Verification, and Testing*. Boca Raton, FL, USA: CRC Press, 2018.
- [20] M. I. Khan and F. Lin, "Impact of transistor model accuracy on harmonic spectra emitted by logic circuits," in *2014 12th IEEE International Conference on Solid-State and Integrated Circuit Technology (ICSICT)*, Guilin, China, Jul. 2014, pp. 1–3, <https://doi.org/10.1109/ICSICT.2014.7021316>.
- [21] M. I. Khan and F. Lin, "Comparative analysis and design of harmonic aware low-power latches and flip-flops," in *2014 IEEE International Conference on Electron Devices and Solid-State Circuits*, Chengdu, China, Jun. 2014, pp. 1–2, <https://doi.org/10.1109/EDSSC.2014.7061282>.
- [22] R. Shoukat and M. I. Khan, "Design and development of a clip building block system for MEMS," *Microsystem Technologies*, vol. 24, no. 2, pp. 1025–1031, Oct. 2018, <https://doi.org/10.1007/s00542-017-3453-2>.
- [23] R. Shoukat and M. I. Khan, "Nanotechnology based electrical control and navigation system for worm guidance using electric field gradient," *Microsystem Technologies*, vol. 24, no. 2, pp. 989–993, Oct. 2018, <https://doi.org/10.1007/s00542-017-3444-3>.
- [24] M. I. Khan, R. Shoukat, K. Mukherjee, and H. Dong, "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*, vol. 24, no. 2, pp. 1201–1206, Oct. 2018, <https://doi.org/10.1007/s00542-017-3486-6>.
- [25] M. I. Khan, H. Dong, F. Shabbir, and R. Shoukat, "Embedded passive components in advanced 3D chips and micro/nano electronic systems," *Microsystem Technologies*, vol. 24, no. 2, pp. 869–877, Oct. 2018, <https://doi.org/10.1007/s00542-017-3586-3>.
- [26] M. I. Khan, A. Qamar, F. Shabbir, and R. Shoukat, "Design, development and implementation of a low power and high speed pipeline A/D converter in submicron CMOS technology," *Microsystem Technologies*, vol. 23, no. 12, pp. 6005–6014, Sep. 2017, <https://doi.org/10.1007/s00542-017-3550-2>.
- [27] Y. A. Durrani, T. Riesgo, M. I. Khan, and T. Mahmood, "Power analysis approach and its application to IP-based SoC design," *COMPEL - The international journal for computation and mathematics in electrical and electronic engineering*, vol. 35, no. 3, Jan. 2016, <https://doi.org/10.1108/COMPEL-08-2015-0283>.
- [28] B. L. Dokic, "A Review on Energy Efficient CMOS Digital Logic," *Engineering, Technology & Applied Science Research*, vol. 3, no. 6, pp. 552–561, Dec. 2013, <https://doi.org/10.48084/etasr.389>.
- [29] M. Dossis, "High-level Synthesis Integrated Verification," *Engineering, Technology & Applied Science Research*, vol. 5, no. 5, pp. 864–870, Oct. 2015, <https://doi.org/10.48084/etasr.596>.
- [30] A. Nouaiti, A. Saad, A. Mesbahi, M. Khafallah, and M. Reddak, "Design and Test of a New Three-Phase Multilevel Inverter for PV System Applications," *Engineering, Technology & Applied Science Research*, vol. 9, no. 1, pp. 3846–3851, Feb. 2019, <https://doi.org/10.48084/etasr.2573>.
- [31] N. Ravindran and R. M. Lourde, "An optimum VLSI design of a 16-BIT ALU," in *2015 International Conference on Information and Communication Technology Research (ICTRC)*, Abu Dhabi, UAE, Feb. 2015, pp. 52–55, <https://doi.org/10.1109/ICTRC.2015.7156419>.
- [32] P. Larsson-Edefors and K. Jeppson, "Timing- and power-driven ALU design training using spreadsheet-based arithmetic exploration," in *10th European Workshop on Microelectronics Education (EWME)*, Tallinn, Estonia, Feb. 2014, pp. 151–154, <https://doi.org/10.1109/EWME.2014.6877416>.
- [33] H. Kaur and H. Singh, "Advanced ALU with inbuilt selection modules for Genetic Algorithm processor," in *2015 International Conference on Signal Processing, Computing and Control (ISPCC)*, Wanknaghat, India, Sep. 2015, pp. 405–410, <https://doi.org/10.1109/ISPCC.2015.7375065>.
- [34] M. Suzuki *et al.*, "A 1.5-ns 32-b CMOS ALU in double pass-transistor logic," *IEEE Journal of Solid-State Circuits*, vol. 28, no. 11, pp. 1145–1151, Aug. 1993, <https://doi.org/10.1109/4.245595>.
- [35] T. Y. Chang and M. J. Hsiao, "Carry-select adder using single ripple-carry adder," *Electronics Letters*, vol. 34, no. 22, pp. 2101–2103, Oct. 1998, <https://doi.org/10.1049/el:19981706>.
- [36] C.-J. Fang, C.-H. Huang, J.-S. Wang, and C.-W. Yeh, "Fast and compact dynamic ripple carry adder design," in *Proceedings. IEEE Asia-Pacific Conference on ASIC*, Taipei, Taiwan, Dec. 2002, pp. 25–28, <https://doi.org/10.1109/APASIC.2002.1031523>.
- [37] N. Burgess, "Fast Ripple-Carry Adders in Standard-Cell CMOS VLSI," in *2011 IEEE 20th Symposium on Computer Arithmetic*, Tuebingen, Germany, Jul. 2011, pp. 103–111, <https://doi.org/10.1109/ARITH.2011.23>.
- [38] Y. Wang and K. K. Parhi, "A unified adder design," in *Conference Record of Thirty-Fifth Asilomar Conference on Signals, Systems and Computers (Cat.No.01CH37256)*, Pacific Grove, CA, USA, Aug. 2001, vol. 1, pp. 177–182, <https://doi.org/10.1109/ACSSC.2001.986901>.
- [39] A. Baliga and D. Yagain, "Design of High Speed Adders Using CMOS and Transmission Gates in Submicron Technology: A Comparative Study," in *2011 Fourth International Conference on Emerging Trends in Engineering Technology*, Port Louis, Mauritius, Aug. 2011, pp. 284–289, <https://doi.org/10.1109/ICETET.2011.25>.
- [40] Y. Choi and E. E. Swartzlander, "Speculative Carry Generation With Prefix Adder," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 16, no. 3, pp. 321–326, Mar. 2008, <https://doi.org/10.1109/TVLSI.2007.915502>.