# Design of Leading Zero Counters on FPGAs

Stefania Perri, *Senior Member, IEEE*, Fanny Spagnolo, *Member, IEEE*, Fabio Frustaci, *Senior Member, IEEE*, and Pasquale Corsonello, *Member, IEEE*

***Abstract*—This letter presents a novel leading zero counter (LZC) able to efficiently exploits the hardware resources available within state-of-the-art FPGA devices to achieve high speed performances with limited energy consumption. Post-implementation results, obtained for operands bit-widths varying between 4- and 64-bit, demonstrate that the new design improves its direct competitors in terms of occupied lookup tables (LUTs), power consumption and computational speed. As an example, when implemented using the Xilinx Artix-7 xc7a100tcsg324 device, the new 64-bit LZC utilizes up to 36% less LUTs, dissipates up to 2.8 times lower power and is up to 20% faster than state-of-the-art counterparts.**

***Index Terms*—Digital circuits, field-programmable gate arrays (FPGAs), leading zero counting (LZC).**

## I. INTRODUCTION

EFFICIENT hardware implementations of leading zero counters (LZCs) are required in several applications, like the floating-point arithmetic computations [1], [2], the conversion of floating-point data to other formats [3], the design of mixed-precision computational units [4], the quantization of Deep Neural Networks (DNNs) [5] and the probabilistic approximate computing [6], just to cite some representative examples.

In recent years, field-programmable gate arrays (FPGAs) have evolved into hardware implementation platforms adequate to support the computational demands of the above cited applications. On the one hand, many researchers focus their efforts towards the design of complex computational data-paths [7], [8], while on the other hand it is of interest to design basic computational modules, such as adders [9], multipliers [10] and LZCs [11], [12]. Typically, novel complex data-paths are designed to utilize the advanced resources on-chip available within an FPGA device, such as digital signal processors (DSPs) and intellectual property (IP) cores, in the best possible manner. On the contrary, to make gate-level innovative designs effective, basic computational modules are designed to use the logic resources based on lookup tables (LUTs), fast carry-chains and flip-flops (FFs) as efficiently as possible.

This letter presents a new FPGA-based design for LZCs. The architecture here described utilizes LUTs more efficiently than previous designs demonstrated in [11]-[12] and exhibits significantly reduced hardware resources requirement, power consumption and computational delay. This is a graceful result, given that, as a part of the critical computational path, the LZC can contribute up to 30% to the worst-case delay of a floating-point unit [13] and up to 15% to the resources utilization [14].

The new LZCs have been implemented and evaluated using the Xilinx Artix 7-series xc7a100tcsg324 [15] and the Altera Cyclone 10 LP 10CL006YE144A7G [16] devices. In both the cases, obtained results clearly show the benefit of the proposed approach over its competitors.

## II. BACKGROUND AND RELATED WORKS

An LZC is a basic computational module able to count the number of consecutive zeros (or ones) within a binary input, starting from its most significant bit (MSB). When an $n$-bit binary number $A_{(n-1:0)}$ is processed, the LZC provides $\log_2(n)+1$ output bits, one of which (typically called $V$) flags that all the $n$ input bits are equal to zero, while the remaining bits (usually named $Z_{(\log_2 n-1:0)}$) represent the number of counted zeros. As an example, in the case of the 8-bit input $A$=00000011, an LZC furnishes $V=0$ with $Z_{(2:0)}=110$.

Several methods exist to determine the leading zero count. In the following, we refer to the FPGA-based implementations [11]-[12] and to the approach presented in [17], that being originally developed for ASIC designs, was replicated in FPGA to extend the comparison with the new method. The basic logic exploited in [11], [12] and [17] for the design of 8-bit LZCs is summarized in the truth table of Fig. 1. Wider leading zero

| | [11] and [17] | [12] | New |
|---|---|---|---|
| $A_7\,A_6\,A_5\,A_4\,A_3\,A_2\,A_1\,A_0$ | $\overline{V}\ \overline{Z_2}\ \overline{Z_1}\ \overline{Z_0}$ | $V\ Z_2\ Z_1\ Z_0$ | $V\ Z_2\ Z_1\ Z_0$ |
| 0 0 0 0 0 0 0 0 | 0 0 0 0 | 1 1 1 1 | 1 1 1 0 |
| 0 0 0 0 0 0 0 1 | 1 0 0 0 | 0 1 1 1 | 0 1 1 1 |
| 0 0 0 0 0 0 1 X | 1 0 0 1 | 0 1 1 0 | 0 1 1 0 |
| 0 0 0 0 0 1 X X | 1 0 1 0 | 0 1 0 1 | 0 1 0 1 |
| 0 0 0 0 1 X X X | 1 0 1 1 | 0 1 0 0 | 0 1 0 0 |
| 0 0 0 1 X X X X | 1 1 0 0 | 0 0 1 1 | 0 0 1 1 |
| 0 0 1 X X X X X | 1 1 0 1 | 0 0 1 0 | 0 0 1 0 |
| 0 1 X X X X X X | 1 1 1 0 | 0 0 0 1 | 0 0 0 1 |
| 1 X X X X X X X | 1 1 1 1 | 0 0 0 0 | 0 0 0 0 |

'X' stay for *don't care*

**Fig. 1.** Truth tables of several 8-bit LZCs.

Manuscript received……., accepted ….. Date of publication ….. date of current version …..

The work of Fanny Spagnolo was supported by "PON Ricerca & Innovazione - Ministero dell'Università e della Ricerca (Grant 1062_R24_INNOVAZIONE).

*(Corresponding author: Stefania Perri).*

Stefania Perri is with the Department of Mechanical, Energy and Management Engineering, University of Calabria, 87036 Rende, Italy (e-mail: s.perri@unical.it).

Fanny Spagnolo, Fabio Frustaci and Pasquale Corsonello are with the Department of Informatics, Modeling, Electronics and Systems Engineering, University of Calabria, 87036 Rende, Italy (e-mail: f.spagnolo@dimes.unical.it; f.frustaci@dimes.unical.it; p.corsonello@unical.it).

This article has been accepted for publication in IEEE Embedded Systems Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LES.2022.3217861



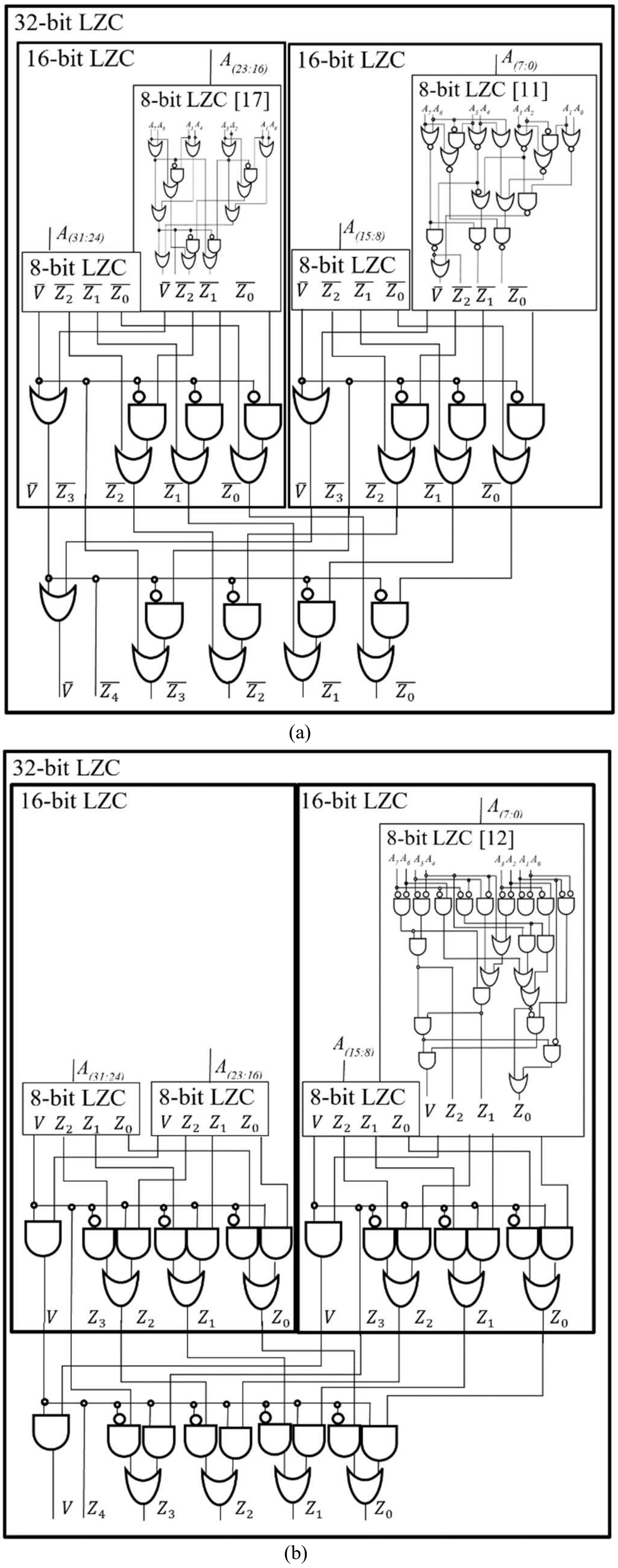


**Fig. 2.** LZC architectures used in a) [11] and [17]; b) [12].

counters are implemented by combining several instances of the 8-bit LZC into a hierarchical structure, as shown in Figs. 2a and 2b for the 32-bit LZCs presented in [11], [12] and [17], respectively. It is worth noting that [11] and [17] use the same hierarchical structure, but, as depicted in the insets of Fig. 2a, their 8-bit LZCs employ quite different logics, that obviously lead to different hardware characteristics. From Figs. 1 and 2b it can be observed that the logic implemented in [12] is completely different from [11] and [17] in both the 8-bit LZC and in the construction of wider LZCs. Indeed, it computes the direct outputs $V$ and $Z_{(2:0)}$ instead of their inverses $\bar{V}$ and $\overline{Z_{(2:0)}}$. Such a logic has been purposely tailored to the Xilinx's FPGA fabric available in the series 7 devices [15].

As an alternative to the above architectures, the fast carry-chains (FC) available within modern FPGA devices may be exploited as shown in [18]. For purposes of comparison, also FC-based LZCs are characterized in the following.

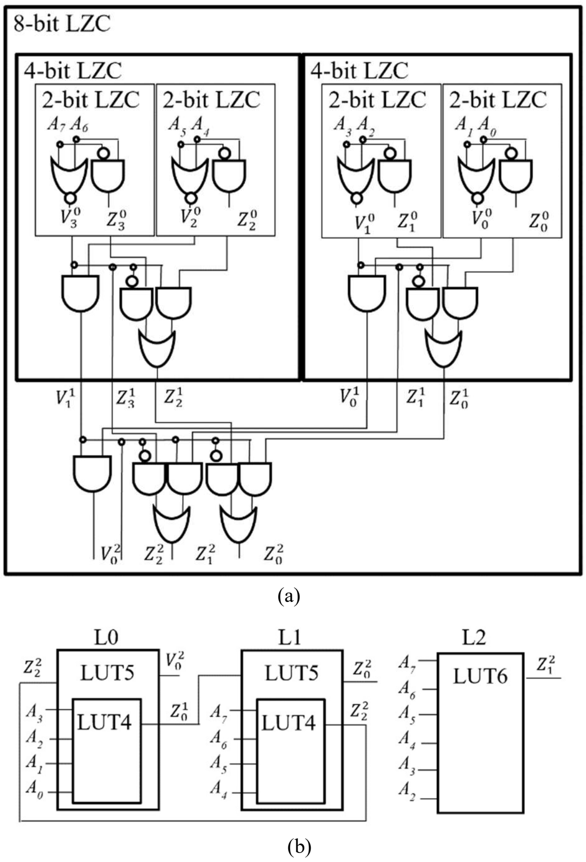


**Fig. 3.** The new 8-bit LZC: (a) the logic; (b) the LUTs usage.

## III. The Proposed Design

This Section introduces the new approach here proposed to design LZCs on FPGAs. It differently treats the condition in which all the input bits are equal to zero, based on the consideration that when $V$=1 the count value $Z$ does not matter at all. However, if a specific value of $Z$ is required in such case, the proposed approach does not require much different additional logic as that required by traditional approaches. With respect to the previously described designs, the proposed method exploits a different granularity. In fact, it uses the 2-bit LZC, instead of the 8-bit one, as the basic block. Consequently, it requires deeper hierarchical architectures to construct wider LZCs. As shown in the following, these choices lead to a LUTs utilization more efficient than [11], [12] and [17], even without applying any optimization process to keep a specific device structure into consideration. Fig. 3a shows the hierarchical structure of the new 8-bit LZC based on two instances of the 4-bit LZC, each being in turn constructed using two instances of the basic 2-bit block whose outputs are combined by four auxiliary gates. The same auxiliary logic is utilized to construct the 8-bit LZC by combining the results obtained from two 4-bit LZCs, and so on for even wider operands. In this paper, $n$-bit LZCs have been designed and characterized, with $n$ varying from 4 to 64. The $n$-bit LZC consists of $\log_2(n)$ hierarchical levels. The first one is composed by $\frac{n}{2}$ instances of the 2-bit LZC and implements (1)-(2), with $j = 0, \dots, \frac{n}{2} - 1$. Conversely, the $l$-th subsequent level, with $l = 1, \dots, \log_2(n) - 1$, implements (3)-(5), with $i = 0, \dots, \frac{n}{2^{l+1}} - 1$ and $x = 0, \dots, l - 1$.

$$V_j^0 = \overline{A_{2j+1} + A_{2j}} = \overline{A_{2j+1}} \cdot \overline{A_{2j}} \tag{1}$$

$$Z_j^0 = \overline{A_{2j+1}} \cdot A_{2j} \tag{2}$$

$$V_i^l = V_{2i}^{l-1} \cdot V_{2i+1}^{l-1} \tag{3}$$

$$Z_{l+i\cdot 2^l}^l = V_{1+2i}^{l-1} \tag{4}$$

$$Z_{x+i\cdot 2^l}^l = \overline{V_{1+2\cdot i}^{l-1}} \cdot Z_{x+i\cdot 2^l+2^{l-1}}^{l-1} + V_{1+2\cdot i}^{l-1} \cdot Z_{x+i\cdot 2^l}^{l-1} \tag{5}$$

It is worth underlining that the proposed 8-bit LZC complies with the third column of the truth table shown in Fig. 1. In fact, in the proposed logic, the case of all zero bits causes the flag $V$ and the output bits $Z_{(\log_2(n)-1:1)}$ to be asserted, while $Z_0$ is zeroed. This behavior allows simplifying the overall logic.

The proposed LZCs have been described using the Very High-Speed Integrated Circuits Hardware Description Language (VHDL) to be then synthesized and implemented within a FPGA device. Fig. 3b shows how the VHDL description of the proposed 8-bit LZC can be implemented within only three 6-input LUTs. The LUT L0 is configured to perform in parallel the 5-input and the 4-input logic functions producing $V_0^2$ and $Z_0^1$. Analogously, L1 computes the signals $Z_0^2$ and $Z_2^2$ by means of the 5- and the 4-input LUTs, respectively. Finally, L2 is configured as one 6-input LUT to compute $Z_1^2$. The logic functions implemented in each LUT are obtained as follows:

- $Z_2^2$ is computed by L1 as given in (6), which is obtained by applying, in order: (4), with $l$=2 and $i$=0; (3), with $l$=1 and $i$=0; (1), with $j$=2, 3.

$$Z_2^2 = V_1^1 = V_2^0 \cdot V_3^0 = \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \tag{6}$$

- L0 computes $V_0^2$ by (7) that is obtained by applying, orderly: (3) and (4), with $l$=2 and $i$=0; (1), with $j$=0, 1.

$$V_0^2 = V_0^1 \cdot V_1^1 = V_0^0 \cdot V_1^0 \cdot Z_2^2 = Z_2^2 \cdot \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot \overline{A_0} \tag{7}$$

- $Z_0^1$ is computed by L0 as given in (8a) that comes from: (5), with $l=1$, $x=0$, $i=0$; (1) and (2), with $j$=0, 1. Then, $Z_0^2$ is provided by L1 as shown in (8b) that is obtained by applying: (5), with $l$=2, $x=0$ and $i=0$; again (5), with $l$=1, $x$=0, $i$=1, to calculate $Z_2^1$; (4), with $l$=2 and $i$=0; (1) and (2), with $j$=2, 3.

$$Z_0^1 = \overline{V_1^0} \cdot Z_1^0 + V_1^0 \cdot Z_0^0 = \overline{A_3} \cdot A_2 + \overline{A_3} \cdot \overline{A_2} \cdot \overline{A_1} \cdot A_0 \tag{8a}$$

$$\begin{aligned} Z_0^2 &= \overline{V_1^1} \cdot Z_2^1 + V_1^1 \cdot Z_0^1 = \overline{Z_2^2} \cdot \left(\overline{V_3^0} \cdot Z_3^0 + V_3^0 \cdot Z_2^0\right) + Z_2^2 \cdot Z_0^1 = \\ &= \overline{Z_2^2} \cdot (\overline{A_7} \cdot A_6 + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot A_4) + Z_2^2 \cdot Z_0^1 = \\ &= (\overline{A_7} \cdot A_6 + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot A_4) + \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_5} \cdot \overline{A_4} \cdot Z_0^1 \end{aligned} \tag{8b}$$

- finally, $Z_1^2$ is computed by L2 as reported in (9) that is obtained by applying: (5), with $l$=2, $x$=1, $i$=0; (4) to calculate $Z_3^1$ and $Z_1^1$; and (1) for computing $V_3^0$, $V_1^0$ and $Z_2^0$.

$$\begin{aligned} Z_1^2 &= \overline{V_1^1} \cdot Z_3^1 + V_1^1 \cdot Z_1^1 = \overline{Z_2^2} \cdot \overline{V_3^0} + Z_2^2 \cdot \left(\overline{V_3^0} \cdot Z_3^0 + V_3^0 \cdot Z_2^0\right) = \\ &= \overline{A_7} \cdot \overline{A_6} \cdot \overline{A_3} \cdot \overline{A_2} + \overline{A_7} \cdot \overline{A_6} \cdot A_4 + \overline{A_7} \cdot \overline{A_6} \cdot A_5 \cdot \overline{A_4} \end{aligned} \tag{9}$$

The 16-bit LZC uses two instances of the 8-bit LZC combined as schematized in Fig. 4. In this case, the LUT-based configuration depicted in Fig. 3b is used twice and, to apply (1)-(4) in the third hierarchical level, two additional LUTs are instantiated, each performing a couple of 4-input logic functions, as required to compute $V_0^3$, $Z_3^3$, $Z_2^3$, $Z_1^3$ and $Z_0^3$.

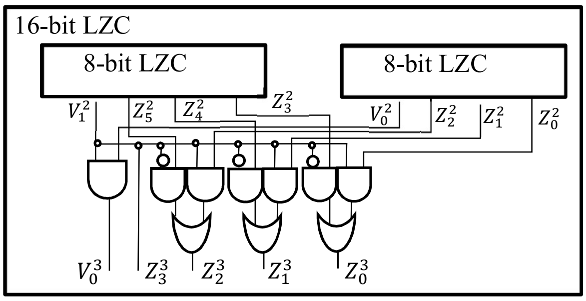


**Fig. 4.** The new 16-bit LZC.

## IV. Implementation Results

The new LZCs have been implemented using the Xilinx Artix 7-series 28-nm xc7a100tcsg324 FPGA device [15]. They have been characterized in terms of occupied LUTs, fast carry-chains (Carry4), computational delay ($D$) and dynamic energy consumption ($E$). Table I summarizes post-implementation results obtained in comparison with other LZCs, including the built-in high-level-synthesis (HLS) leading zero counting function characterized in [12]. The new designs utilize less LUTs, are faster and dissipate lower dynamic energy than the designs [11], [12], [17] and HLS. These benefits come from the deeper hierarchy exploited in wider LZCs and the simplification introduced to treat the case in which all the input bits are equal to zero. As an example, when $n$=64, the new LZC uses 37.3%, 21.7%, 33.8% and 35.6% less LUTs than the designs [11], [12], [17] and HLS, respectively. Moreover, it is 16.2%, 23%, 13.2% and 19.4% faster and dissipates 3.37, 2.89, 3.1 and 3.2 times lower energy.

In comparison with the FC-based designs, the new LZCs always save significant amounts of resources and dissipate up to ~6.5 times lower energy at the expense of a delay at most only ~13% worse.

The proposed LZC architecture overcomes its direct competitors also in terms of the cost function *EnDeLUC* defined in (10), It can be seen that, with $n$=8, higher improvements are achieved with respect to [11] and [17]. Then, as $n$ increases to 16 or greater, also the advantage over the designs [12], HLS and FC-based becomes quite evident.

TABLE I
POST-IMPLEMENTATION RESULTS RELATED TO THE XC7A100TCSG324 DEVICE

| | $n$ | #LUTs | #Carry4 | $D$ [ns] | $E$ [pJ] | *EnDeLUC* |
|---|---|---|---|---|---|---|
| New | 4 | 2 | 0 | 0.92 | 0.019 | 0.035 |
| [11] | 8 | 6 | 0 | 1.92 | 2.71 | 31.22 |
| [12] | 8 | 4 | 0 | 1.87 | 1.61 | 12.04 |
| [17] | 8 | 5 | 0 | 1.62 | 2.14 | 17.33 |
| HLS | 8 | 5 | 0 | 2.2 | 1.9 | 20.9 |
| FC-based | 8 | 8 | 7 | 1.25 | 0.78 | 14.62 |
| New | 8 | 3 | 0 | 1.2 | 0.12 | 0.43 |
| [11] | 16 | 16 | 0 | 1.98 | 4.67 | 147.95 |
| [12] | 16 | 10 | 0 | 2.71 | 2.79 | 75.61 |
| [17] | 16 | 14 | 0 | 2.06 | 4.16 | 119.97 |
| HLS | 16 | 14 | 0 | 2.38 | 4.76 | 158.6 |
| FC-based | 16 | 16 | 15 | 1.74 | 1.8 | 97.1 |
| New | 16 | 10 | 0 | 1.69 | 0.49 | 8.28 |
| [11] | 32 | 39 | 0 | 2.76 | 5.05 | 543.58 |
| [12] | 32 | 26 | 0 | 3.03 | 4.12 | 324.57 |
| [17] | 32 | 33 | 0 | 2.84 | 5.03 | 471.41 |
| HLS | 32 | 36 | 0 | 2.92 | 5.17 | 543.47 |
| FC-based | 32 | 32 | 41 | 1.99 | 3.21 | 466.32 |
| New | 32 | 24 | 0 | 2.26 | 0.89 | 48.27 |
| [11] | 64 | 75 | 0 | 3.52 | 6.47 | 1708.08 |
| [12] | 64 | 60 | 0 | 3.83 | 5.55 | 1275.39 |
| [17] | 64 | 71 | 0 | 3.4 | 5.93 | 1431.5 |
| HLS | 64 | 73 | 0 | 3.66 | 6.11 | 1632.47 |
| FC-based | 64 | 64 | 97 | 2.81 | 5.31 | 2402.3 |
| New | 64 | 47 | 0 | 2.95 | 1.92 | 266.21 |

$$EnDeLUC = E \cdot D \cdot (\#LUTs + \#Carry4) \qquad (10)$$

Note that the HLS built-in function occupies less LUTs and dissipates less energy than [11]. Indeed, while HLS designs are mapped on cascaded LUTs, the LZCs [11] exploit parallel logic that leads to slightly lower delays.

Results obtained from a leading zero detector, used as a simple toy circuit, have shown that out of 359 LUTs, 7.29ns of maximum delay and 31pJ of energy consumption, the contribute of our 32-bit LZC is 6.7%, 31% and 2.9%, respectively.

TABLE II
EVALUATIONS FOR THE 10CL006YE144A7G DEVICE

| LZC | $n$ | LEs | | | | $D$ [ns] |
|---|---|---|---|---|---|---|
| | | Total | 4-input | 3-input | 2-input | |
| New | 4 | 3 | 2 | 0 | 1 | 1.08 |
| [12] | 8 | 9 | 7 | 2 | 0 | 1.72 |
| New | 8 | 8 | 5 | 2 | 1 | 1.53 |
| [12] | 16 | 20 | 16 | 1 | 3 | 2.68 |
| New | 16 | 17 | 13 | 2 | 2 | 2.36 |
| [12] | 32 | 42 | 33 | 7 | 2 | 3.12 |
| New | 32 | 38 | 30 | 2 | 6 | 2.94 |
| [12] | 64 | 86 | 67 | 12 | 7 | 3.96 |
| New | 64 | 78 | 60 | 11 | 7 | 3.56 |

LUT requirements and computational delays have been evaluated also for the Altera Cyclone 10 LP series 60nm 10CL006YE144A7G device [16] at the 1.2V Slow Corner Model @85°C. This device has been chosen since it provides 4-input LUTs. Table II shows that, in comparison with the new designs, the best implementations among [11], [12] and [17] are up to 13.5% slower and utilize up to 17% more LUTs.

## V. CONCLUSION

This letter presented new designs of LZC that use the 2-LZC as the basic block and adopt a different way of dealing with the case in which all the input bits are zero. In comparison with state-of-the-art competitors, the new LZCs are cheaper, faster and consume significantly lower energy. As a further result, the efficiency of the proposed designs has been demonstrated referring to different FPGA devices.

## REFERENCES

[1] H.Suzuki, H. Morinaka, H. Makino, Y. Nakase, K. Mashiko, T. Sumi, “Infrared navigation—Part I: An assessment of feasibility,” *IEEE J. of Solid-State Circ.*, vol. 31, no. 8, pp. 1157–1163, Aug. 1996.
[2] V. G. Oklobdzija, “An Algorithmic and Novel Design of a Leading Zero Detector Circuit: Comparison with Logic Synthesis”, *IEEE Trans. On Very Large Scale Int. (VLSI) Sys.*, Vol. 2, n° 1, pp. 124-128, March 1994.
[3] Aneesh R, V. Patil, Sobha PM and A. David selvakumar, "HMFPCC: - Hybrid-mode floating point conversion co-processor", in *Proc.* VLSI-SATA, Bengaluru, India, 2015, pp. 1-6.
[4] H. Zhang, H. J. Lee and S. -B. Ko, "Efficient Fixed/Floating-Point Merged Mixed-Precision Multiply-Accumulate Unit for Deep Learning Processors," in *Proc.* ISCAS, Florence, Italy, 2018, pp. 1-5.
[5] H. F. Langorudi, V. Karia, T. Pandit, D. Kudithipudi, “TENT: Efficient Quantization of Neural Networks on the tiny Edge with Tapered FixEd PoiNT”, *arXiv preprint arXiv:2104.02233, 2021*.
[6] S. Liu, P. Reviriego, P. Junsangsri, F. Lombardi, “Probabilistic Approximate Computing at Nanoscales”, *IEEE Nanotech. Mag.*, vol. 16, no. 1, pp. 16-24, Feb. 2022.
[7] S. Perri, F. Frustaci, F. Spagnolo and P. Corsonello, "Design of Real-Time FPGA-based Embedded System for Stereo Vision," in *Proc.* ISCAS, Florence, Italy, 2018, pp. 1-5.
[8] F. G. Zacchigna, "Methodology for CNN Implementation in FPGA-based Embedded Systems," *IEEE Embedded Systems Letters*, early access, doi: 10.1109/LES.2022.3187382.
[9] S. Perri, F. Frustaci, F. Spagnolo, P. Corsonello, "Efficient Approximate Adders for FPGA-Based Data-Paths," *Electronics*, vol. 9, no. 9, pp. 1-19, Sept. 2020.
[10] S. Ullah, T. D. A. Nguyen and A. Kumar, "Energy-Efficient Low-Latency Signed Multiplier for FPGA-Based Hardware Accelerators," *IEEE Embedded Systems Letters*, vol. 13, no. 2, pp. 41-44, June 2021.
[11] J. Miao, S. Li, "A design for high speed leading-zero counter", in *Proc.* ISCE, Kuala Lumpur, Malaysia, 2017, pp. 22-23.
[12] A. Zahir, A. Ullah, P. Reviriego and S. R. U. Hassnain, "Efficient Leading Zero Count (LZC) Implementations for Xilinx FPGAs", *IEEE Embedded Systems Letters*, vol. 14, no. 1, pp. 35-38, March 2022.
[13] M. Olivieri, F. Pappalardo, S. Smorfa, G. Visalli, “Analysis and Implementation of a Novel Leading Zero Anticipation Algorithm for Floating-Point Arithmetic Units”, *IEEE Trans. on Circ. and Sys. – II: Expr. Brief*, vol. 54, no. 8, pp. 685-689, August 2007.
[14] J. Hrica, “Floating-point design with Vivado HLS”, XAPP599 (V1.0), September 20, 2012. [Online]. Available: https://www.xilinx.com
[15] “7 Series FPGAs Configurable Logic Block”, UG474 (v1.8), September 27, 2016. [Online]. Available: https://docs.xilinx.com
[16] “Intel Cyclone 10 LP Device Overview”, C10LP51001, 2020.05.21. [Online]. Available: https://www.intel.com/content/www/us/en/docs/
[17] G. Dimitrakopoulos, K. Galanopoulos, C. Mavrokefalidis and D. Nikolos, "Low-Power Leading-Zero Counting and Anticipation Logic for High-Speed Floating Point Units," *IEEE Trans. on Very Large Scale Int. (VLSI) Sys.*, vol. 16, no. 7, pp. 837-850, July 2008.
[18] K. Chaudhary, “Method for implementing priority encoders using FPGA carry logic”, 1998, [Online]. Available: https://patents.google.com/patent/US6081914A/en.