Area-Efficient Parallel-Prefix Binary Comparator

Amit Kumar Panda
Department of Electrical Engineering
IIT Patna, Bihta, Patna, India
amitee.srf13@iitp.ac.in

Rakesh Palisetty
Department of ECE, KL University
Greenfields, Vaddeswaram, AP, India
rakeshp@kluniversity.ac.in

Kailash Chandra Ray
Department of Electrical Engineering
IIT Patna, Bihta, Patna, India
kcr@iitp.ac.in

***Abstract*— Binary comparator is the most basic component for comparing the two binary operands in different arithmetic functional blocks, digital communication and encryption/ decryption devices. Normally, the binary comparator gives three decision outputs in many cases but in some applications like coupled linear congruential generator (CLCG) based pseudorandom bit generator (PRBG), it gives single output decision on the basis of inequality equation $x_i > y_i$. Magnitude and mux-based comparator are the most commonly used binary comparator for comparing the two binary operands. However, the parallel prefix structure of these comparators increases the area in the order of $O(n)$. Therefore, in this paper a new area efficient single decision output binary comparator is developed using merging circuit technique for the computation of inequality comparison $x_i > y_i$ in CLCG based PRBG methods. The use of merging circuit in the prefix computation stages considerably reduces the area of the proposed binary comparator as compared to other popular exiting comparator architectures. The proposed comparator of 32- and 64- bit word size is designed with Verilog HDL and implemented on Spartan3E FPGA device to study its hardware performances. It reports that the proposed 32- bit comparator consumes 23.6% and 16.1% less LUT area as compared to the magnitude and mux-based comparators respectively.**

Keywords*— Binary comparator, Digital arithmetic, Coupled linear congruential generator (CLCG), FPGA prototype.

I. INTRODUCTION

Binary comparator is considered as one of the basic arithmetic operation that is used in various hardware digital systems. It is widely used in many digital processing and computing blocks such as CPU, microcontroller, digital communication systems, sorting data, and encryption/ decryption for determining whether one binary number is greater than, equal to, or less than the other binary number [1]-[3]. So, the binary comparator compares the two binary coded numbers and generates three decision outputs such as A_{big} $(A > B)$, EQ $(A = B)$ and B_{big} $(A < B)$. In some applications such as coupled linear congruential generator (CLCG) and CLCG based pseudorandom bit generator (PRBG) involve inequality comparison to generate pseudorandom bit sequence [4]-[7]. In the CLCG system, the binary coded output x_i of one linear congruential generator (LCG) compares with the binary coded output y_i of another LCG and generates single bit output. If $x_i > y_i$ then the output of CLCG is '1' else it is '0' [4]. Therefore, a single decision output inequality comparator ($x_i > y_i$) is an essential component in CLCG based PRBG. The performance of these fast computing digital processing hardware systems relies on the efficient implementation of the binary comparator.

The conventional binary comparator using subtractive method (2's complement) is the simplest technique which incorporates two-operand adder [8]. However, the adder based comparator has the limitation of the slower operating speed. It is only useful when the number of input bits of the binary comparator is short. The slow response of the adder based binary comparator is overcome by the parallel prefix tree structure [9] for longer input bits. There have been several parallel prefix tree structure binary comparators previously published in the literature [9]-[11]. Among these, magnitude [9] and mux-based [10] binary comparators are the most widely used parallel prefix comparator techniques which significantly improve the delay performance. Beside the improvement of the delay and speed performance in the parallel prefix tree structure comparators, the area increases linearly in the order of $O(n)$ [9]-[11].

Therefore, in this paper a new area efficient binary comparator is proposed for comparing the two binary operands ($A > B$) in different arithmetic functional blocks. The proposed comparator is a parallel prefix tree structure in which the first stage is the array of 2- bit magnitude comparator and the remaining stages are designed with 3-to-1 merging circuit. The merging circuit is the three input and one output combinational block in which the bit wise comparison outputs of the previous stages are merged together with the help of AND, OR and NOT gate logics to produce a single comparison output (A_{big} or B_{big}) [12]. The proposed binary comparator consumes considerably less hardware area as compared to the existing parallel-prefix techniques. The proposed binary comparator of 32- and 64-bit word size is designed with Verilog and synthesized using Xilinx XST tool for performance analysis. Further the design is prototyped on Xilinx Spartan3E XC500E FPGA device for functional verification.

Rest of this paper is organized as follows; Existing parallel prefix comparators are briefly discussed in Section-II. The proposed binary comparator and its first order complexity analysis are presented in Section-III. Section-IV highlights the FPGA prototype and physical implementation results of the proposed comparator architecture. Finally, Section-V concludes the paper.

II. EXISTING PARALLEL PREFIX COMPARATOR

A brief description of the existing parallel prefix tree structure binary comparators is provided in this section. Magnitude and mux-based comparators are the two most commonly used parallel prefix comparator techniques which reduce the critical delay and hence improve the speed of the comparator. The conventional magnitude comparator is a combinational circuit that compares two binary coded numbers A and B , and then determines their relative magnitudes A_{big} $(A > B)$, EQ $(A = B)$ and B_{big} $(A < B)$ as shown in Fig. 1 [9]. The logic diagram of 1-, 2-, 4- and 16-bit magnitude comparators are presented in [9]. In all four designs, the comparison output provides three binary variables. The n- bit parallel prefix binary comparator using 2- bit magnitude comparator is presented in [6] for efficient implementation of the inequality equation $x_i > y_i$ of the modified dual-CLCG architecture. The logical expressions of the 2- bit magnitude comparator for computing A_{big} and B_{big} are defined as,

978-1-7281-4655-3/19/$31.00 ©2019 IEEE
DOI 10.1109/iSES47678.2019.00016

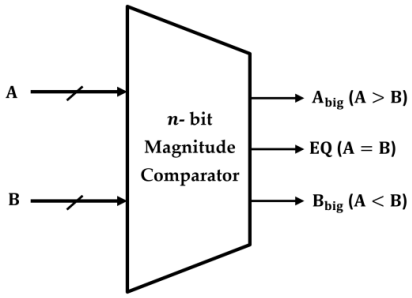


Fig. 1 Block diagram of n- bit magnitude comparator

$$\mathrm{A_{big}} = \mathrm{A_1}\overline{\mathrm{B_1}} + \left(\overline{\mathrm{A_1}\overline{\mathrm{B_1}} + \overline{\mathrm{A_1}}\mathrm{B_1}}\right) \cdot \mathrm{A_0}\overline{\mathrm{B_0}} \quad (1)$$

$$\mathrm{B_{big}} = \overline{\mathrm{A_1}}\mathrm{B_1} + \left(\overline{\mathrm{A_1}\overline{\mathrm{B_1}} + \overline{\mathrm{A_1}}\mathrm{B_1}}\right) \cdot \overline{\mathrm{A_0}}\mathrm{B_0} \quad (2)$$

The logical diagram of 2- bit magnitude comparator is shown Fig. 2 that computes $\mathrm{A_{big}}\ (\mathrm{A} > \mathrm{B})$ and $\mathrm{B_{big}}\ (\mathrm{A} < \mathrm{B})$ signals by comparing two 2- bit binary operands [6].

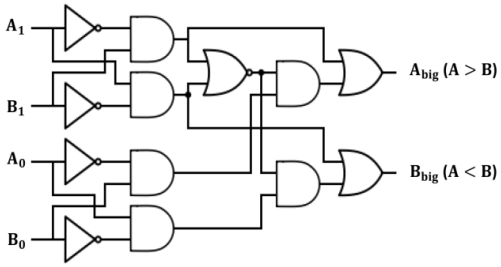


Fig. 2 Logic diagram of 2- bit magnitude comparator [6]

The n- bit magnitude comparator requires $(\log_2 n + 1)$ stages of 2- bit comparator array to compute the two-operand binary comparison $\mathrm{A_{big}}\ (\mathrm{A} > \mathrm{B})$ and therefore, the critical path delay is in the order of $O(\log_2 n)$. Hence, the complexity of area ($\mathrm{A_{mag}}$) and delay ($\mathrm{T_{mag}}$) of the magnitude comparator are evaluated [6] as follows,

$$\text{Area, } \mathrm{A_{mag}} = (n-1)[9\mathrm{A_G} + 4\mathrm{A_N}]$$

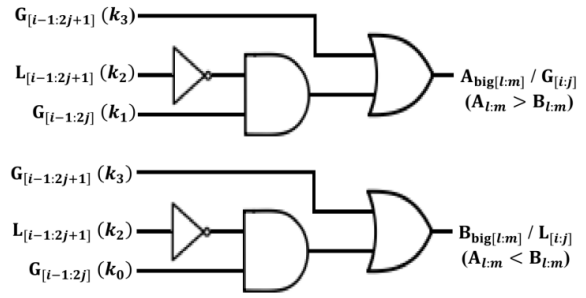
$$\text{Delay, } \mathrm{T_{mag}} = 4(\log_2 n)\mathrm{T_G}$$

Here, $\mathrm{A_G}$ and $\mathrm{A_N}$ are denoted as area of 2-input basic gate and NOT gate respectively. $\mathrm{T_G}$ and $\mathrm{T_X}$ denote the delay of 2-input basic gate and NOT gate respectively.

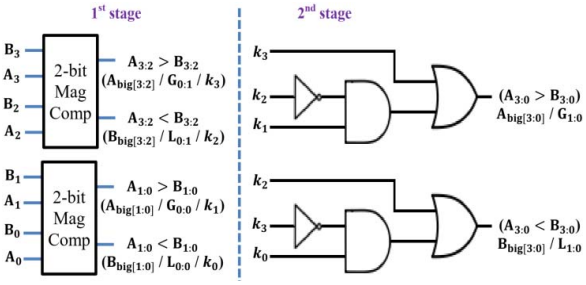
Similarly, the mux-based comparator as presented in [7] and [9] is also a parallel prefix tree structure in which the first stage is the parallel combination of OR-AND gates to generate G_i and P_i. Where, $\mathrm{G}_i = \mathrm{A}_i + \overline{\mathrm{B}_i}$ and $\mathrm{P}_i = \mathrm{A}_i\overline{\mathrm{B}_i}$. The subsequent comparator stages are the parallel combination of 2-to-1 multiplexers. It has been observed that the mux-based comparator consumes less area than the magnitude comparator. The complexity of area ($\mathrm{A_{mxc}}$) and delay ($\mathrm{T_{mxc}}$) of the mux-based binary comparator are evaluated [7] as follows,

$$\text{Area, } \mathrm{A_{mxc}} = [8n - \log_2 n - 7]\mathrm{A_G} + [3n - \log_2 n - 2]\mathrm{A_N}$$

$$\text{Delay, } \mathrm{T_{mxc}} = (2\log_2 n + 1)\mathrm{T_G} + (\log_2 n + 1)\mathrm{T_N}$$



(a)



(b)

Fig, 3 Logic diagram of (a) 3-to-1 merging circuit and (b) 4- bit proposed comparator using merging circuit.

Therefore, the area complexity in both parallel prefix comparators increases, the area increases linearly in the order of $O(n)$. Hence, the area of the comparator is necessary to reduce for the efficient comparison of large input size binary coded numbers.

III. Proposed Binary Comparator

To further reduce the area, a new efficient inequality binary comparator is proposed in this section that compares two binary coded numbers and produces single decision output $\mathrm{A_{big}}\ (\mathrm{A} > \mathrm{B})$. The proposed comparator technique is a parallel prefix structure which is designed using the parallel combination of 3-to-1 merging circuit. The first stage of the proposed comparator is the parallel combination of 2- bit magnitude comparator as shown in Fig. 2. The subsequent stages of the proposed comparator are the parallel combination of 3-to-1 merging circuit. The merging circuit is the three input and one output combinational block in which the bit wise comparison outputs of the previous stages are merged together with the help of AND, OR and NOT logic gates to produce a single comparison output ($\mathrm{A_{big[}}_{l:m]}$ or $\mathrm{B_{big[}}_{l:m]}$) [12]. The logical expressions of the 3-to-1 merging circuit for computing $\mathrm{A_{big}}$ and $\mathrm{B_{big}}$ are defined as follows,

$$\begin{aligned}\mathrm{A_{big[}}_{l:m]} &= \mathrm{A_{[}}_{l:m]} > \mathrm{B_{[}}_{l:m]} = \mathrm{G_{[}}_{i:j]} \\ &= k_3 + \overline{k_2} \cdot k_1 \\ &= \mathrm{G_{[}}_{i-1:2j+1]} + \overline{\mathrm{L_{[}}_{i-1:2j+1]}} \cdot \mathrm{G_{[}}_{i-1:2j]} \quad (3)\end{aligned}$$

$$\begin{aligned}\mathrm{B_{big[}}_{l:m]} &= \mathrm{A_{[}}_{l:m]} < \mathrm{B_{[}}_{l:m]} = \mathrm{L_{[}}_{i:j]} \\ &= k_2 + \overline{k_3} \cdot k_0 \\ &= \mathrm{L_{[}}_{i-1:2j+1]} + \overline{\mathrm{G_{[}}_{i-1:2j+1]}} \cdot \mathrm{L_{[}}_{i-1:2j]} \quad (4)\end{aligned}$$

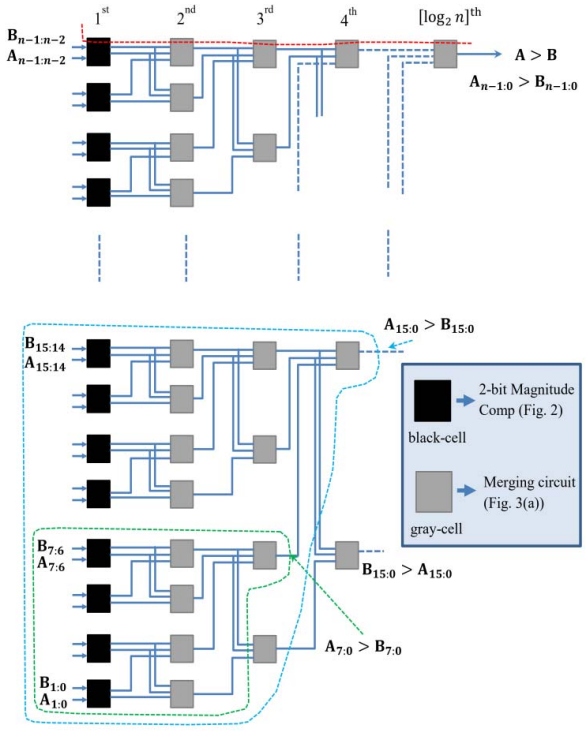


Fig. 4 Architecture of the proposed n- bit parallel prefix comparator for comparing $A > B$.

In the equations (3) and (4), the notations l and m are defined as, $l = (j+1)\cdot 2^{(i+1)} - 1$, $m = j \cdot 2^{(i+1)}$.

Here, $k_0 = \mathrm{L}_{[i-1:2j]}$, $k_1 = \mathrm{G}_{[i-1:2j]}$, $k_2 = \mathrm{L}_{[i-1:2j+1]}$ and $k_3 = \mathrm{G}_{[i-1:2j+1]}$ are the comparison outputs of the previous stage comparators. The notations i and j are the integer numbers such that $0 \le i, j < n$. The logical diagram of 3-to-1 merging circuit and the proposed comparator of 4- bit operand size using merging circuit are shown in Fig. 3(a) and Fig. 3(b) respectively. The logical expression of the 4- bit proposed comparator using merging circuit is defined as,

$$\mathrm{A}_{\mathrm{big}[3:0]} = \mathrm{A}_{[3:0]} > \mathrm{B}_{[3:0]} = \mathrm{G}_{[1:0]}$$
$$= k_3 + \overline{k_2}\cdot k_1 = \mathrm{G}_{[0:1]} + \overline{\mathrm{L}_{[0:1]}}\cdot \mathrm{G}_{[0:0]}$$
$$\mathrm{B}_{\mathrm{big}[3:0]} = \mathrm{A}_{[3:0]} < \mathrm{B}_{[3:0]} = \mathrm{L}_{[1:0]}$$
$$= k_2 + \overline{k_3}\cdot k_0 = \mathrm{L}_{[0:1]} + \overline{\mathrm{G}_{[0:1]}}\cdot \mathrm{L}_{[0:0]}$$

Similarly, the architecture of the proposed parallel prefix comparator for the n- bit operand size can be developed by employing 3-to-1 merging circuits in the subsequent stages as shown in Fig. 4. The architecture of the proposed n- bit parallel prefix comparator shown in Fig. 4 has $\log_2 n$ comparison stages. The first stage is the parallel combination of 2- bit magnitude comparators that compares two 2- bit binary coded numbers and provides output signals $\mathrm{A}_{\mathrm{big}[l+m]}$ and $\mathrm{B}_{\mathrm{big}[l+m]}$. The 2-bit magnitude comparator is highlighted as black-cell in Fig. 4 which is realized with the logical expressions as defined in the equations (1) and (2). The detail logical diagram of 2- bit magnitude comparator is shown in Fig. 2. The subsequent stages of the proposed comparator are the parallel combination of the merging circuit that provides $\mathrm{A}_{\mathrm{big}[l+m]}$/ $\mathrm{G}_{[i:j]}$ or $\mathrm{B}_{\mathrm{big}[l+m]}$/ $\mathrm{L}_{[i:j]}$ depending on the input combinations as defined in equations (3) and (4). The 3-to-1 merging circuit is highlighted as gray cell in Fig. 4 which is realized with the logical expressions as defined in the equations (3) and (4). The detail logical diagram of 3-to-1 merging circuit is shown in Fig. 3(a).

TABLE I
AREA AND TIMING COMPLEXITY OF THE PROPOSED COMPARATOR

Method	Timing Complexity	Area Complexity
for n- bit		
RCA based comp	$2\mathrm{T_X} + 2n\mathrm{T_G}$	$(n+1)[2\mathrm{A_X} + 3\mathrm{A_G} + \mathrm{A_N}]$
Magnitude comp [6]	$4(\log_2 n)\mathrm{T_G}$	$(n-1)[9\mathrm{A_G} + 4\mathrm{A_N}]$
Mux-based comp [7]	$(2\log_2 n + 1)\mathrm{T_G} + (\log_2 n + 1)\mathrm{T_N}$	$[8n - \log_2 n - 7]\mathrm{A_G} + [3n - \log_2 n - 2]\mathrm{A_N}$
Proposed comparator	$(2\log_2 n + 1)\mathrm{T_G} + (\log_2 n)\mathrm{T_N}$	$(7n-6)\mathrm{A_G} + 3(n-1)\mathrm{A_N}$
for $n = 32$- bit		
RCA based comp	$2\mathrm{T_X} + 64\mathrm{T_G}$	$66\mathrm{A_X} + 99\mathrm{A_G} + 33\mathrm{A_N}$
Magnitude comp [6]	$20\mathrm{T_G}$	$279\mathrm{A_G} + 124\mathrm{A_N}$
Mux-based comp [7]	$11\mathrm{T_G} + 6\mathrm{T_N}$	$258\mathrm{A_G} + 89\mathrm{A_N}$
Proposed comparator	$11\mathrm{T_G} + 5\mathrm{T_N}$	$218\mathrm{A_G} + 93\mathrm{A_N}$
for $n = 64$- bit		
RCA based comp	$2\mathrm{T_X} + 128\mathrm{T_G}$	$130\mathrm{A_X} + 195\mathrm{A_G} + 65\mathrm{A_N}$
Magnitude comp [6]	$24\mathrm{T_G}$	$567\mathrm{A_G} + 252\mathrm{A_N}$
Mux-based comp [7]	$13\mathrm{T_G} + 7\mathrm{T_N}$	$513\mathrm{A_G} + 184\mathrm{A_N}$
Proposed comparator	$13\mathrm{T_G} + 6\mathrm{T_N}$	$442\mathrm{A_G} + 189\mathrm{A_N}$

A. Complexity analysis of the proposed comparator

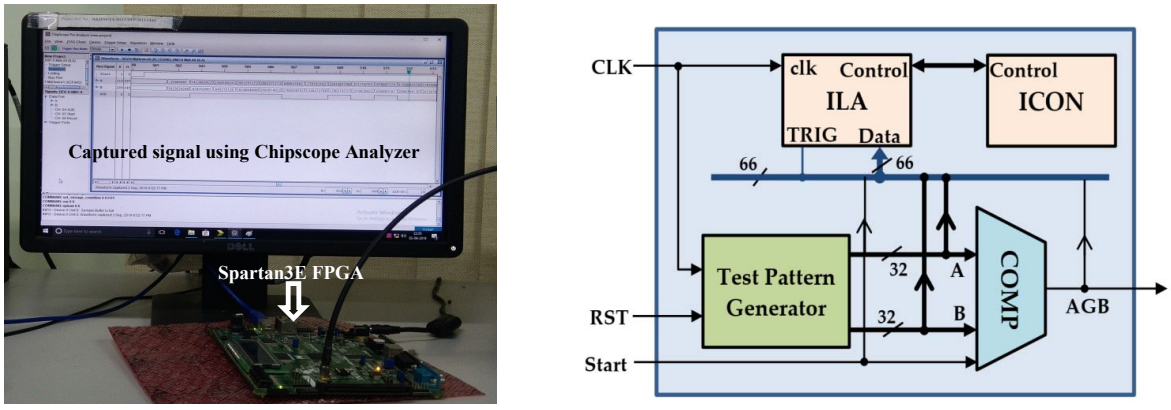
Since the performance of the proposed comparator mainly depends on merging circuit, the complexity of area and maximum delay are evaluated as follows,

$$\text{Area, } \mathrm{A_{prop}} = (7n-6)\mathrm{A_G} + 3(n-1)\mathrm{A_N}$$

$$\text{Critical path, } \mathrm{T_{prop}} = (2\log_2 n + 1)\mathrm{T_G} + (\log_2 n)\mathrm{T_N}$$

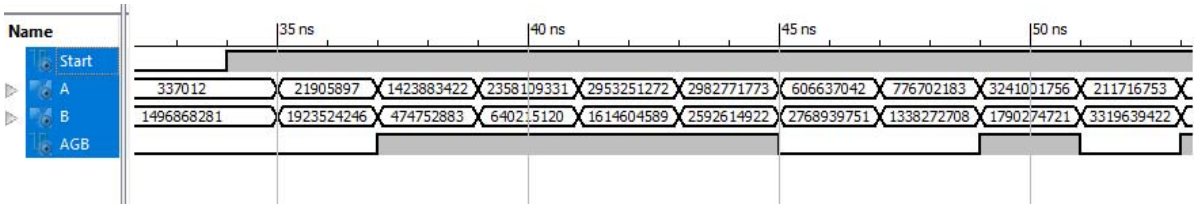
Here, $\mathrm{A_G}$ and $\mathrm{A_N}$ are denoted as area of 2-input basic gate and NOT gate respectively. $\mathrm{T_G}$ and $\mathrm{T_X}$ denote the delay of 2-input basic gate and NOT gate respectively.

The area and timing complexity of the proposed parallel prefix comparator using merging circuit along with RCA-based, magnitude and mux-based comparators are summarized in Table I. Further, the comparison of 32- and 64- bit size proposed architecture is also highlighted in Table I. It is observed that the proposed comparator saves 21.8% and 15.5% basic gate ($\mathrm{A_G}$) area as compared to the magnitude and mux-based comparators respectively for n= 32- bit operand. Similarly, it saves the basic gate ($\mathrm{A_G}$) area of 22.1% and 13.8% as compared the magnitude and mux-based comparators respectively for n= 64- bit operand. Therefore, the proposed parallel prefix comparator using merging circuit is the most area efficient comparator technique that considerably consumes less area while maintaining the less critical delay as compared to the other existing binary comparators.

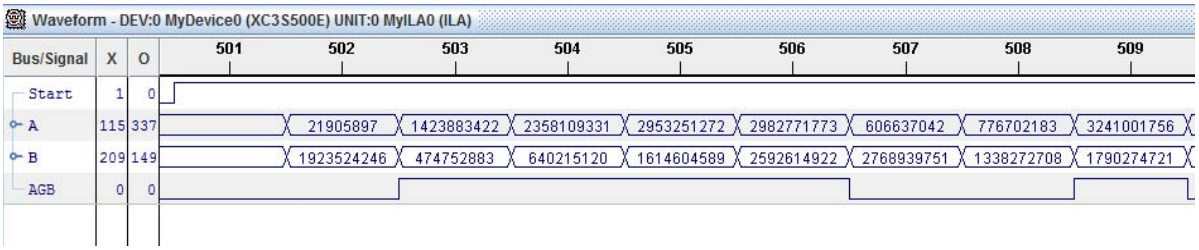


(a) (b)

Fig. 5 (a) Laboratory experimental setup of the proposed parallel prefix binary comparator, (b) Block diagram for functional testing of the proposed comparator using Chipscope analyzer,.



(a)



(b)

Xilinx Chipscope analyzer.

IV. FPGA Prototype and Physical Implementation Results of the Proposed Comparator

The proposed binary comparator architecture is designed using Verilog HDL and implemented on the commercially available Xilinx Spartan3E XC3S500E FPGA device to study its timing performance and resource utilization. The experimental laboratory setup of the proposed binary comparator architecture is demonstrated in Fig. 5(a). The comparator output signals of the 32- bit architecture is captured through USB-JTAG programming cable in real-time and displayed on the PC using Xilinx Chipscope analyzer tool.

A. Functional Verification Using Xilinx Chipscope

The proposed comparator architecture is implemented on the targeted Spartan3E FPGA device and the real-time comparator output signal is captured using Xilinx Chipscope analyzer for functional verification. To capture the rea-time output signal, the Chipscope IP core module (ICON and ILA) is inserted along with the proposed comparator into the FPGA. The schematic diagram of the proposed comparator block integrated with Chipscope IP core block is shown in Fig. 5(b). For the readers' convenience, a 32- bit design of the proposed comparator architecture is simulated with Xilinx ISIM tool which is shown in Fig. 6(a). The behavioral simulation result of the proposed comparator is further validated with the real-time captured signal in Fig. 6(b).

B. Physical Implementation Results

The physical implementation results of the proposed binary comparator for n = 32- and 64- bit are highlighted in Table II. For a fair comparison, other comparator architectures are also implemented on the same FPGA platform and the results are reported in Table II.

It is observed that the proposed binary comparator saves the area of 23.6% and 16.1% LUT resources on Spartan3E FPGA device as compared the magnitude and mux-based comparators respectively for n= 32- bit. Similarly, it saves

TABLE II

PHYSICAL IMPLEMENTATION RESULTS

Method	Word Size n	Area (LUTs)	Delay (C.P.)
RCa-based comparator	64- bit	80	57.214
	32- bit	39	30.259
Magnitude comparator	64- bit	142	42.956
	32- bit	68	23.980
Mux-based comparator	64- bit	130	17.908
	32- bit	62	19.024
Proposed comparator	64- bit	112	18.757
	32- bit	52	15.982

the area of 21.1% and 13.8% LUT resources on Spartan3E FPGA device as compared the magnitude and mux-based comparators respectively for n= 64- bit. It is also observed that the proposed binary comparator takes less combinational delay to compute the comparison of two binary coded numbers as compared other parallel prefix tree structure comparators.

Therefore, the proposed parallel prefix comparator can be used at the output stage of the CLCG architecture [4] for efficient generation of the pseudorandom bit sequence. The CLCG consists of two linear congruential generator blocks and one comparator that generates pseudorandom bit sequence by comparing the LCG outputs x_{i+1} with y_{i+1} as defined in [4]. The area of the CLCG method can be reduced significantly by using the proposed comparator technique.

V. CONCLUSION

A new efficient parallel prefix binary comparator is proposed using 3-to-1 merging circuit that significantly reduces the hardware area while maintaining the less critical delay as compared to the other existing binary comparators. The first stage of the proposed adder is designed with the parallel combination of 2- bit magnitude comparator and the subsequent stages are developed with the parallel combination of 3-to-1 merging circuit. The merging circuit is the three input and one output combinational block in which the bit wise comparison outputs of the previous stages are merged together and produces a single comparison output. The use of merging technique in the proposed comparator considerably reduces the overall area. The proposed comparator architecture is synthesized using Xilinx XST tool and further prototyped on the Spartan3E FPGA device. The physical implementation results of the proposed comparator reported that the overall area is reduced by 23.6% and 16.1% for 32- bits size as compared to the magnitude and mux-based comparators respectively. Similarly, it is reduced by 21.1% and 13.8% for 64- bit size. Therefore, the proposed area efficient parallel-prefix binary comparator can be used for computing the inequality equation $x_i > y_i$ in CLCG based PRBG methods.

ACKNOWLEDGMENT

This research work is supported by SMDP-C2SD Project (Ref. no.- R&D/SP/EE/DeitY/SMDP-C2SD/2015-2016/126) funded by Ministry of Electronics and Information Technology (MeitY), Govt. of India, New Delhi, India.

REFERENCES

[1] Shun-Wen Cheng, "Arbitrtuy Long Digit Sorter HWISW Co-Design," in *Proc. Asia and South Pacific Design Auromarion Conj*, ASPDAC'03, pp. 538-543, Jan. 2003.

[2] S. J. Piestrak, "Efficient Hamming weight comparators of binary vectors," in *Electronics Letters*, vol. 43, no. 11, pp. 611-612, 24 May 2007.

[3] Z. Ullah, "LH-CAM: Logic-Based Higher Performance Binary CAM Architecture on FPGA," in *IEEE Embedded Systems Letters*, vol. 9, no. 2, pp. 29-32, June 2017.

[4] Raj S. Katti and S. Srinivasan, "Efficient hardware implementation of a new pseudo-random bit sequence generator," in *Proc. IEEE International Symposium on Circuits and Systems*, ISCAS-2009, pp. 1393–1396, 24-27 May 2009, Taipei, Taiwan.

[5] R. S. Katti, R. G. Kavasseri, and V. Sai, "Pseudorandom bit generation using coupled congruential generators", *IEEE Trans. on Circuits and Systems II*, vol. 57, no. 3, pp. 203-207, 2010.

[6] A. K. Panda and K. C. Ray, "Modified dual-CLCG method and its VLSI architecture for pseudorandom bit generation", *IEEE Transaction on Circuit and System –I: Regular Paper*, vol. 66, no. 4, pp. 989-1002, 2019.

[7] A. K. Panda and K. C. Ray, "A coupled variable input LCG method and its VLSI architecture for pseudorandom bit generation", *IEEE Transaction on Instrumentation and Measurement*, April, 2019 (early access).

[8] V. G. Oklobdzija, B. R. Zeydel, H. Dao, S. Mathew, and S. Krishnamurthy, "Energy-delay estimation technique for high-performance microprocessor VLSI adders," in *Proc. Int. Symp. Comp. Arithmetic (ARITH-16)*, 2003, pp. 272–279.

[9] S. Cheng, "A high-speed magnitude comparator with small transistor count," *10th IEEE Int. Conf. on Electronics, Circuits and Systems, 2003. ICECS 2003*, pp. 1168-1171 Vol.3.

[10] H. Lam and C. Tsui, "A mux-based High-Performance Single-Cycle CMOS Comparator," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 54, no. 7, pp. 591-595, July 2007.

[11] C. Chua, R.B.N. Kumar and B. Sireesha, "Design and analysis of low-power and area efficient N-bit parallel binary comparator", Analog Integr Circ Sig Process, Springer, vol. 92, issue 2, pp. 225-231, June 2017.

[12] V. V. R. Teja, K. C. Ray, I. Chakrabarti and A. S. Dhar, "High Throughput VLSI Architecture for One Dimensional Median Filter," *2008 International Conference on Signal Processing, Communications and Networking*, Chennai, 2008, pp. 339-344.