Low Power Binary Comparator Design for Better Operated Portable Devices

Meenu Shrivastava[1] and Shweta Agrawal[2]
[1]Research scholar,[2]Assistant Professor,
Electronics and Comm. dept., SRCEM Banmore, Morena, India

***Abstract*-The low power efficient devices must have very convenient functioned for portable devices. It has applications in communication and calculation areas. The main intention of this paper is to provide a new binary comparator that gives significant reduction in the power and area. The effectiveness of the proposed technique over the existing is evaluated by coding in Verilog, synthesized and simulated on Xilinx tool chain and design metrics are evaluated. The simulation results on FPGA show that the proposed comparator provides more than 2.56% reduction in power over the best-known comparator.**

Keywords: **Portable devices, Comparator, High speed integrated circuits, VLSI, Low-power design, FPGA**

I. INTRODUCTION

Comparator is an important data path element for any general-purpose architecture [1]. Moreover, it is also a crucial component in application-specific and signal-processing architectures [2]. A binary comparator has always been an important block in an arithmetic logic unit and also has extensive applications in many digital systems. Comparator is a basic arithmetic unit that compares the magnitude of two binary numbers, say A and B, and produces output bits: greater (A>B) or small (A<B) or equal (A=B). It is an important data-path element for any general purpose architecture as well as an essential device for application-specific and signal processing architectures [1, 2]. Comparators are also used in sorting networks which play an important role in areas such as parallel computing, multi-access memories and multiprocessing [3, 4, 5, 6].

In digital system, comparison of two numbers is an arithmetic operation that determines if one number is greater than, equal to, or less than the other number [1]. So comparator is used for this purpose. Magnitude comparator is a combinational circuit that compares two numbers, A and B, and determines their relative magnitudes (Fig.1). The outcome of comparison is specified by three binary variables that indicate whether A>B, A=B, or A<B [5]. The Comparator is a very basic and useful arithmetic component of digital systems. There are several approaches to designing CMOS comparators, each with different operating speed, power consumption, and circuit complexity. One can implement the comparator by flattening the logic function directly [3][8].

The performance of the comparator significantly affects the overall performance of these processing units [2]. Significant efforts have been given to improve area, power and delay parameters of the comparator at different level of abstraction [3-6]. The existing architectures of the magnitude comparator are not power and performance efficient [3], thus, demanding novel comparator architecture that provides highly energy efficient comparison of two numbers. This paper presents an energy efficient comparator for different signal processing application.

In this paper following are the sections organized: Section II presents the work done to achieve low power high performance comparator and further critically analyses each of the design techniques. Section III details proposed low power comparator architecture whereas its effectiveness using via simulation is given in Section IV. Finally Section IV concludes the paper.

II. LOW POWER COMPARATORS

The section explores different comparator architectures in details. It starts with the discussion on traditional comparator followed by priority based and look-ahead based comparator designs.

2.1 Traditional Comparator Architecture

The block diagram of the magnitude comparator as shown in Figure 1 has two inputs and three outputs [3]. The comparator compares two inputs A and B and provides output as either EQ (A=B), G (A>B) or S (A<B) given by the following equations below.

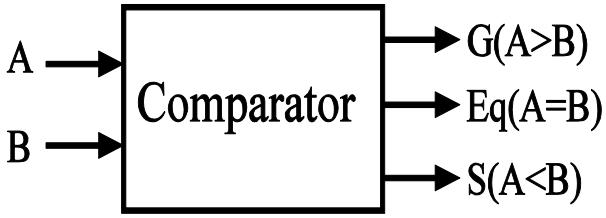

Figure 1: Block diagram of magnitude comparator

The circuit for comparing two n-bit numbers has 2^n inputs and 2^{2n} entries in the truth table, for example in 1-bit comparator has 4-rows in the truth table, whereas 2-bit comparator has 16 rows in the truth table.

2.1.1 4- bit Comparator Architecture

The 4-bit comparator compares two 4-bit binary numbers A and B, and gives three outputs (G, S and Eq). Let inputs A and B have bits A3, A2, A1, A0 and B3, B2, B1, B0 respectively.

The logical expression for the 4-bit comparators output *G*, *S* and Eq are given by the equations below.

$$Eq = (A_3 \odot B_3).(A_2 \odot B_2).(A_1 \odot B_1).(A_0 \odot B_0)$$

$$G = A_3.\overline{B_3} + (A_3 \odot B_3).A_2.\overline{B_2} + (A_3 \odot B_3).(A_2 \odot B_2).A_1.\overline{B_1} + (A_3 \odot B_3).(A_2 \odot B_2).(A_1 \odot B_1).A_0.\overline{B_0}$$

$$S = \overline{A_3}.B_3 + (A_3 \odot B_3).\overline{A_2}.B_2 + (A_3 \odot B_3).(A_2 \odot B_2).\overline{A_1}.B_1 + (A_3 \odot B_3).(A_2 \odot B_2).(A_1 \odot B_1).\overline{A_0}.B_0$$

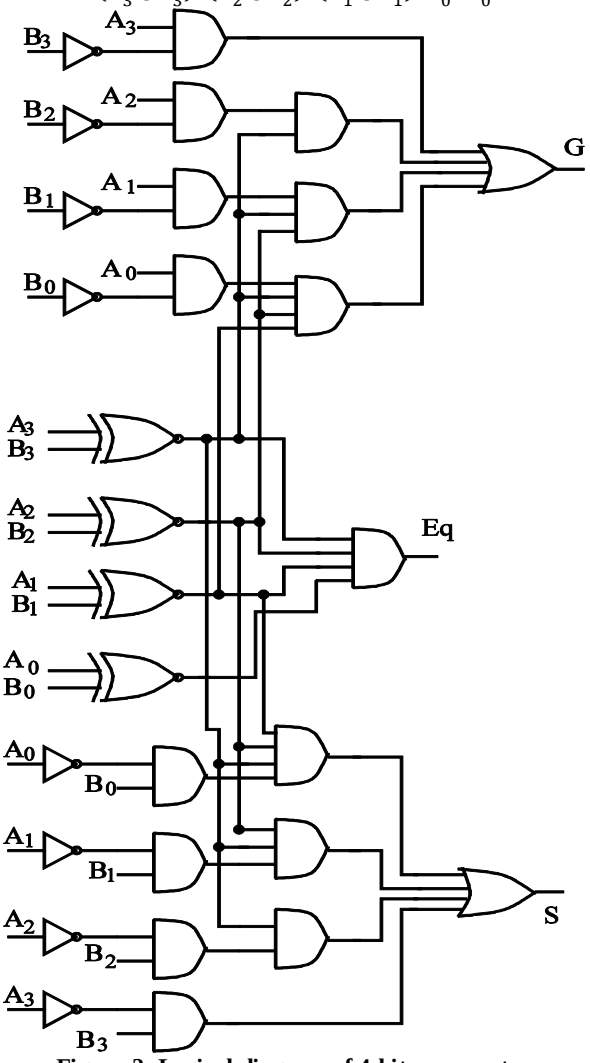

Figure 2: Logical diagram of 4-bit comparator

The logical diagram for the greater (*G*), smaller (*S*) and equal (Eq) are shown in the Figure 2. It can be observed that as the input bit-width increases the complexity of the designs increases significantly. Therefore direct implementation of the higher bit-width comparator is costly in terms of area, power and delay as increasing complexity increases the power and delay. Hence, 4-bit comparator is used to design higher bit-width comparator.

It is observed that as the input bit-width increases the complexity of the designs increases significantly. Therefore direct implementation of the higher bit-width comparator is costly in terms of area, power and delay as increasing complexity increases the power and delay. Hence, 4-bit comparator is used to design higher bit-width comparator.

2.1.2 Extensive bit-width Comparator

The direct implementation of large bit-width comparator is very complex and costly [7-10]. Therefore, we can implement extensive bit-width comparator using small bit-width comparator. Figure 3 shows 16-bit comparator design using five 4-bit comparators [11]. It can be seen from the figure that these 4-bit comparator have only G and S signal but not Eq signal, as it equality condition can be generated by greater and smaller signal. Thus, removing generation of equal signal in the 4-bit comparator reduces the design complexity of the extensive bit-width comparator.

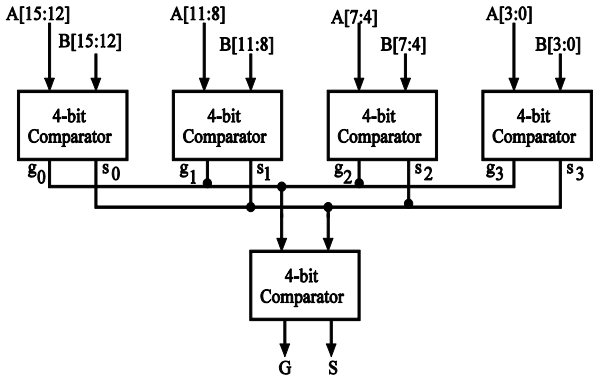

Figure 3: 16-bit comparator using five 4-bit magnitude comparators

In the first stage, four comparators compares group of four bits of 16-bit number whose outputs are given to the final comparator in the second stage.

2.2 Priority Based Comparator (PBC) Architecture

Priority based comparator [13] has three stages to compute output as is shown in Figure 4. First stage identify 1's in each input that may cause the number may be greater over the other number. The second step identify most significant in each number. The third step identifies which number have 1's at more significant position over the other.

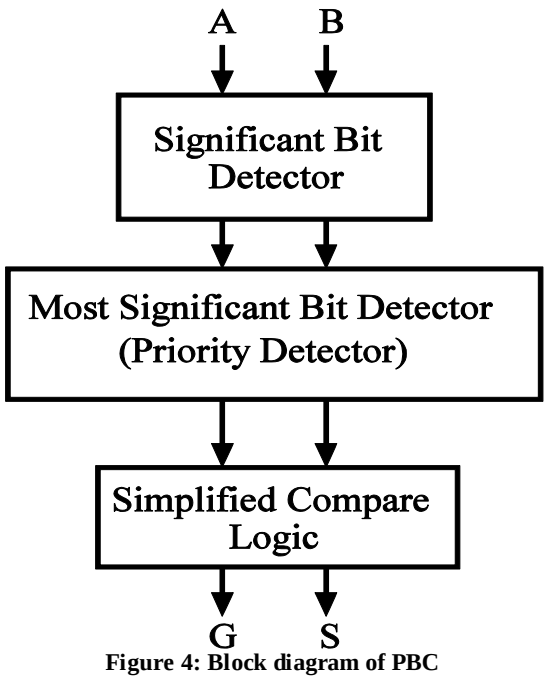

Figure 4: Block diagram of PBC

All the three stages are combined to achieve priority based comparator as shown in Figure. 5

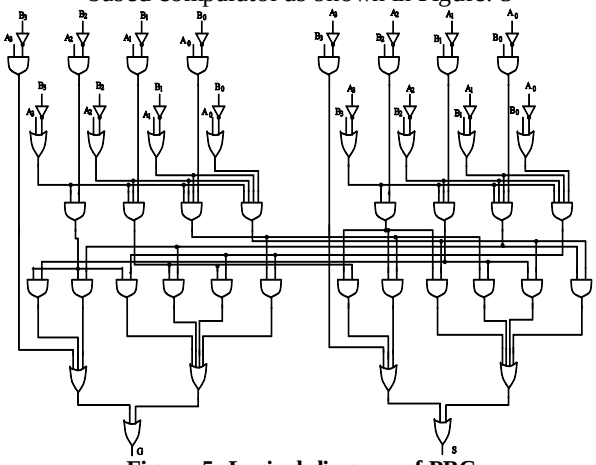

Figure 5: Logical diagram of PBC

2.3 Look-ahead Comparator Architecture

The look-ahead comparator (LAC) [13] is based on the concept of look-ahead adder where carry-in is calculated in advance to eliminate carry dependency. In the LAC, a look-ahead logic computes the bits which decides the which number is greater/smaller. The block diagram of the look-ahead block accepts two four bit numbers and generates 4-bit comp output. Only one bit of the comp out will be high if the number is greater or smaller else are zero it reflect that numbers are equal.

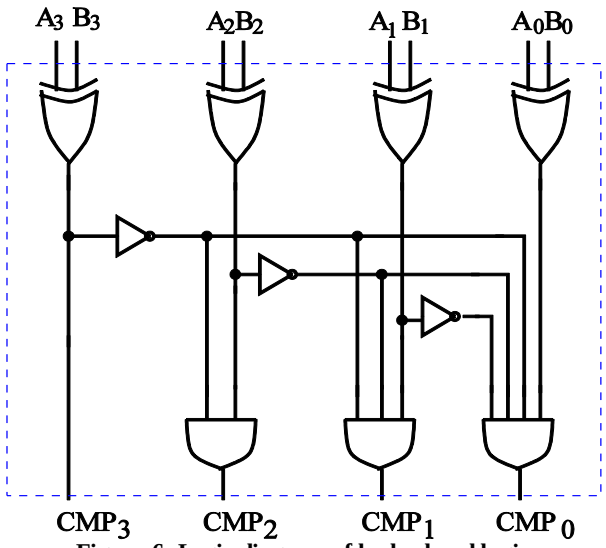

Figure 6: Logic diagram of look-ahead logic

The logical diagram of the look-ahead logic [14] as shown in Figure 6 requires four XOR gate to find 1's which corresponds to the '0' in the other number and then provides ultimate compare output based on the value of the XNOR out. If the most significant XOR is at logic '1' it reflect that this bit will cause output will be greater or smaller, if this bit is on logic '0', other significant XOR will be searched. In this way it evaluates the compare signal.

2.4 32-bit Comparator Architecture

The schematic for 32-bit level implementation of the traditional and proposed comparators is shown in Figure 7.The blocks of the first stage compute the comparison result for every 4 bits of the input numbers. The blocks in the second stage take the result of four sets of 4-bit numbers and compute the result for the two 16-bit numbers which are obtained when the four sets of 4-bit numbers are concatenated. This logic is repeated in the third stage where the 2-bit block takes the results of two sets of 16-bit numbers and computes the result for the two 32-bit numbers [13].

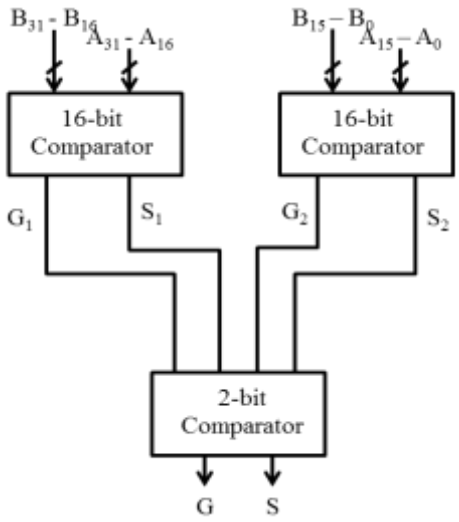

Figure 7: Block diagram of 32-bit Comparator

III. PROPOSED LOW POWER HIGH PERFORMANCE COMPARATORS

The architecture of the proposed comparator is shown in Figure 8. The proposed comparator implement the logic for greater and equal signal. The equal signal is used further used to generate small signal.

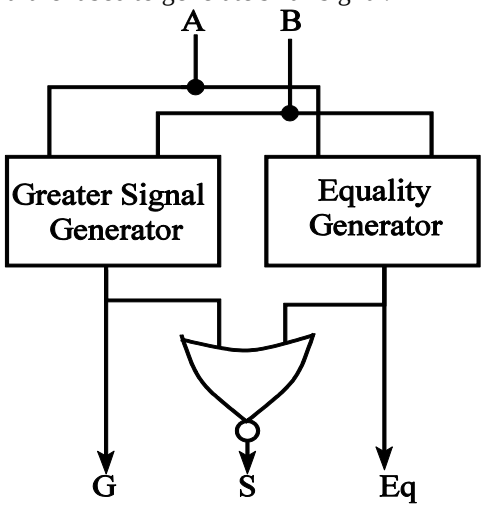

Figure 8: Proposed comparator Architecture

It is observed in the literature that complexity in terms of area, power and delay are more for greater and smaller over the equal. Therefore, we introduce comparator as shown in Figure 8 which computes greater and equal in place of greater and smaller. The greater and equal signals are further used to generate the small signal.

In order to compare the complexity of the proposed design over the existing we implemented the whole design with the 2-input NAND gate as 2-input NAND gate is universal gate which is standard benchmark. The equal logic requires 23 NAND gates while the greater and smaller logic requires 33 2-input NAND gates. Since in the traditional and existing architectures, logic for G and S are implemented simultaneously, it requires 66 two input NAND gates. On the other hand proposed scheme requires only 57 NAND gates as it implements only greater and equal logic and do not implement equal logic.

As the proposed approach requires less area over the traditional, it will consume less power and will have less delay. The simulation results in the next section show the efficacy of the proposed over existing comparator architectures.

IV. SIMULATION ENVIORNMENT AND RESULTS DISCUSSION

This section provides the simulation environment and detailed analysis of the simulation results to compute the efficacy of the proposed comparator architecture. All the comparator designs are implemented in Verilog. The Xilinx ISE 14.5 is used to synthesize the proposed and existing comparator architectures. Test bench for all the designs are created and simulated to verify the functionality of each designs. Further, design metrics such as area, power and delay are extracted and compared. Following subsections provides the simulation results and their analysis for the proposed comparator over the existing comparators.

4.1 Simulation results of 32-bit comparators:

All the comparator designs are coded in Verilog and implemented on Xilinx Vertex XC7VX330T. The implementation complexity in terms of area, maximum combinational delay and power consumed is evaluated. The area of the design is calculated in terms of number of look-up tables (LUT) used. The design metrics are shown in the Table 1.

Table 1: Metrics of the 32-bit comparators

Technique Comparator	Area (#LUTs)	Delay (nS)	Power (mW)
Traditional	32	4.09	241.4
Priority Based	32	3.94	244.3
Look-ahead	31	4.84	252.8
Subtractor based	28	4.67	260.8
Proposed	32	4.23	235.2

It can be observed from the simulation results as shown in the Figure 5.18 that the power requirement of the proposed comparator is minimum over the existing comparator architectures. The proposed comparator requires 2.56%, 3.68%, 6.74%, and 9.61% reduced power consumption over Traditional, Priority based, Look-ahead and Subtractor based comparator architectures respectively. Further, the proposed

comparator also requires less delay over the most of the existing comparator architectures. Finally it can be observed the power-delay production (PDP) which reflects the energy requirement is also small from the look-ahead and Subtractor based comparator architectures.

The area requirement of the proposed comparator is very small over the existing comparator archiectures as shown in Figure 9. The area is measured in terms of number of (Look Up Tables) LUTs which reflects the required combination logic to implement the desired logic where proposed comparator archtiecture requires same number of LUT as required by conventional.

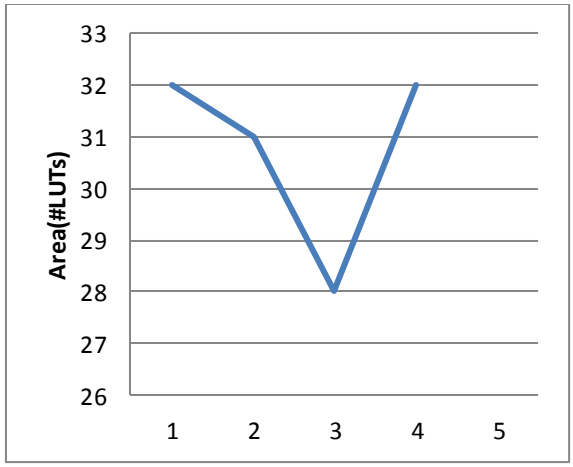

Figure 9: Area of the different 32-bit comparator Architecture

It can be observed from the Figure 9 that proposed comparator requires 4% reduced area over the Subtractor based comparator.

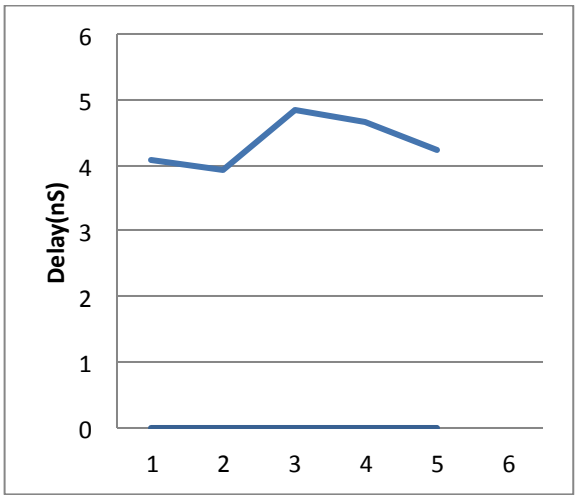

Figure 10: Delay of the different 32-bit comparators

It can be observed from the Figure 10, proposed 32-bit comparator exhibits smaller delay over the look-ahead comparator. The proposed comparator requires 12.6% and 9.5% less delay over look-ahead and subtractor based comparator architectures respectively.

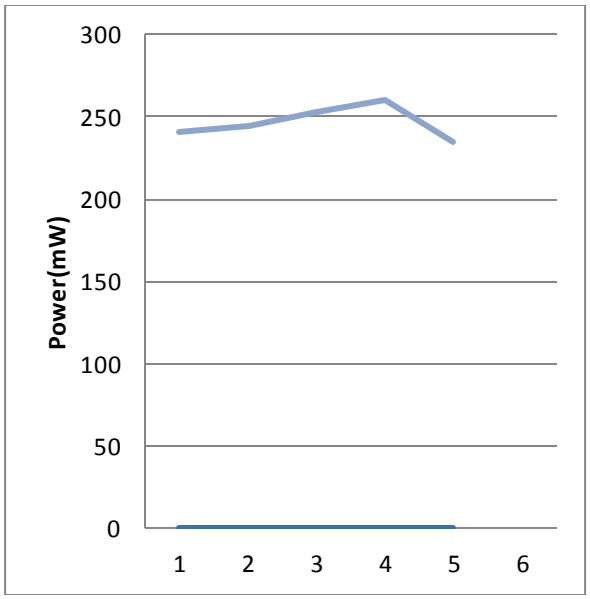

Figure 11: Power of the different 32-bit comparators

It can be observed from the simulation results as shown in the Figure 11 that the power requirement of the proposed comparator is minimum over the existing comparator architectures. The proposed comparator requires 2.56%, 3.68%, 6.74%, and 9.61% reduced power consumption over Tradtional, Priority based, Look-ahead and Subtractor based comparator architectures respectively. Further, the proposed comparator also requires less delay over the most of the existing comparator architectures. Finally it can be observed the power-delay production (PDP) which reflects the energy requirement is also small from the look-ahead and subtractor based comparator architectures.

V. CONCLUSION

Thus, the simulation results are obtained in 32-bit comparator the Area, Power, Delay, and Power Delay Production provides low power design as compared to existing comparators. Hence it is suggest that the proposed comparator can be effectively utilized in the applications requiring low power and high performance. The proposed and existing comparator are implemented in Verilog and processed with Xilinx ISE tool chain. Further the designs are synthesized and post synthesis results are extracted. The simulation results show nearly 2.56% power reduction over the best known architecture. Hence, it is well suited for the battery operated portable devices. Further, this comparator can

be also utilized in the application requires higher density.

REFERENCES

[1] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers, 2002.

[2] J. Eyre and J. Bier, "DSP processors hit the mainstream", IEEE Computer, pp. 51–59, 1999.

[3] M. Morris Mano, Digital Logic and Computer Design.

[4] Fabio Frustaci, Stefania Perri, Marco Lanuzza, and Pasquale Corsonello, "A New Low-Power High-Speed Single-Clock-Cycle Binary Comparator", pp. 317-320, 2010.

[5] Vijaya Shekhawat, Tripti Sharma and Krishna Gopal Sharma, "2-Bit Magnitude Comparator using GDI Technique", IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), pp. 225-229, May 09-11-2014.

[6] Anjuli and Satyajit Anand, "2-Bit Magnitude Comparator Design Using Different Logic Styles", International Journal of Engineering Science Invention, Vol. 2 Issue: 1, pp. 13-24, January 2013.

[7] Geetanjali Sharma, Uma Nirmal, and Yogesh Misra, "A Low Power 8-bit Magnitude Comparator With Small Transistor Count Using Hybrid PTL/CMOS Logic", IJCEM International Journal of Computational Engineering & Management, Vol. 12, pp. 110-115, April 2011.

[8] Oguz Ergin, Kanad Ghose, Gurhan Kucuk, and Dmitry Ponomarev, "A Circuit–Level Implementation of Fast, Energy–Efficient CMOS Comparators for High–Performance Microprocessors", in IEEE International Conference on Computer Design: VLSI in Computers and Processors (ICCD'02), pp. 115-118.

[9] Chung-Shun Huang, Jinn-Shyan Wang, "High-Performance and Power-Efficient CMOS Comparators", IEEE Journal of Solid-State Circuits, Vol. 38, No.2, pp. 254-262, February 2003.

[10] Hing-Mo Lam and Chi-Ying Tsui, "A MUX-based High-Performance Single-Cycle CMOS Comparator", IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 54, pp. 591-595, 7 July 2007.

[11] Stefania Perri and Pasquale Corsonello, "Fast Low-Cost Implementation of Single-Clock-Cycle Binary Comparator", IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 55, pp. 1239-1243, 12 December 2008.

[12] Pierce Chuang, David Li, and Manoj Sachdev, "A Low-Power High-Performance Single-Cycle Tree-Based 64-Bit Binary Comparator", IEEE Transactions on Circuits and Systems—II: Express Briefs, Vol. 59, No. 2, pp. 108-112, February 2012.

[13] Suman Deb and Saurabh Chaudhury, "High-Speed Comparator Architectures for Fast Binary Comparison", Third International Conference on Emerging Applications of information Technology (EAIT), pp. 454- 457, 2012.

[14] T. Suryakala, B. Swaroopa, Bhaskara Rao Doddi, "Circuit Design of Low area 8-bit magnitude Comparator with Low Power by Static CMOS", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Vol. 4, Issue: 10, pp. 3982-86, October 2015.

[15] Shun-Wen Cheng, "A High-Speed Magnitude Comparator with Small Transistor Count" in Proceedings of IEEE international conference ICECS, Vol. 3, pp. 1168 – 1171, December 2003.

