Digital Comparator With Multiple Inputs

1st Abhay Chopde
Electronic And Telecommunication
Vishwakarma institute of Technology
Pune, Indian
abhay.chopde@vit.edu

2nd Swapnil Patil
Electronic And Telecommunication
Vishwakarma institute of Technology
Pune, Indian
swapnil.patil21@vit.edu

3rd Devyani Ushir
Electronic And Telecommunication
Vishwakarma institute of Technology
Pune, Indian
devyani.ushir21@vit.edu

4th Sameer Sumbhe
Electronic And Telecommunication
Vishwakarma institute of Technology
Pune, Indian
sameer.sumbhe21@vit.edu

Abstract— ***Digital comparators are critical circuit components for a wide range of applications that require comparison. A digital comparator compares binary-formatted input numbers and determines whether one of the provided inputs is less, greater, or equal to the other inputs. To compare binary data, a traditional digital comparator has two inputs. In the field of digital design, multi-input comparator designs have been uncommon. So, in this paper, we will look at three algorithms for designing a four-bit four-input digital comparator, as well as the schematic circuit design for all algorithms. When more than one input is greatest, the output shows not only the greatest input, but all the greatest inputs. To simulate schematic circuits of the, the Xilinx ISE tool was used.***

Keywords— ***Digital Comparator, Multi-input Comparator, Xilinx tool, Binary Data, Digital Design, Verilog, HDL***

I. INTRODUCTION

A comparator is fundamentally an arithmetic element that is a crucial component of combinational logic in modern circuitry. A digital comparator's most basic design compares two binary numbers and decides which of them is greater, smaller, or both equal.[1] There are two basic types of conventional two-input digital comparators: Identity comparators and magnitude comparators. Identity comparators demonstrate only the equality of two inputs. The magnitude comparator demonstrates the relation between them as whether one is greater, smaller or both are equal. Digital comparators are essential components of the data path for many different applications, including graphics and image processing. Also, in disciplines like digital signal processing, picture and speech processing, etc., the comparison function is extensively employed. Thus, a high-performance comparator is hugely important for real-time VLSI (Very Large Scale Integration) applications. For example, comparisons are heavily utilized during the key-point localization step of object recognition in image processing as the images are represented using binary data. The comparison also plays a significant role in data-intensive applications like 3D graphics. Comparison instructions are widely used in computations by both general-purpose processors and processors specialized for particular applications, such as media processors and vision processors. . Sorting networks and multi-processing applications also employ comparison operations.

Comparators are also a crucial part of microcontrollers and central processing units. Though voltage comparators or analog comparators are extensively used in microcontrollers, digital comparators perform to be a main part of the circuitry when the data is in binary format and requires the functioning as bitwise. Many applications, including battery chargers, analog to digital converters, and IR sensors, need the employment of such analog comparators when the controller needs to compare two signals. In ADCs and DACs also, comparators are the heart of the circuitry. A conventional comparator permits us to compare up to two inputs only. So, when there are more than two inputs, the data is compared subsequently i.e., the cascading structure of more than one comparator is used. In such a design of comparators, the first two inputs are compared and then their result is compared with further inputs, and so on. However, such cascading structures are not relevant in high-speed systems. Thus, the main issue with the usual approach is that performance declines proportionately as the number of inputs rises. A multi-input comparator indicates the lowest and (or) greatest signal values among a given collection of input signals. Because there has not been enough high-level digital research and development on multi-input comparators, designers cannot get the performance they need. A comparator with multiple inputs has a big effect on how fast the circuit is overall. Therefore, there is a need for effective, more simplified multi-input comparators' designs.

In this paper, three algorithms for design of such a multiple-input digital comparator are proposed. The comparator has four inputs of four binary bits and will indicate which of the inputs are greatest in magnitude. The multi-output issue has also been looked upon in the discussion. The paper also discusses these three algorithms in detail by performing an analysis of their delay and power usage. Xilinx ISE 14.7 suite has been used to verify the three algorithms and the design of the comparator on the Spartan6 board.

II. LITERATURE SURVEY

Mukherjee et al aims to design and implement a digital comparator using different logic techniques to compare power consumption, propagation delay, and transistor count. The digital comparator is designed for low power consumption, higher packing densities, and high-speed interface applications in 45-nanometer technology at 0.7 v supply voltage. The paper proposes a modified transmission gate-based two-bit digital comparator designed with only 30 transistors, which is 72.97% less than the existing transmission gate logic-based two-bit digital comparator. The proposed design uses a half-adder logic circuit, which results in 74.46% less power consumption than the existing transmission gate logic-based two-bit digital comparator. The paper simulated the results of the proposed design using the EDA Tanner tool. [1]

Maji and et al. tried a new technique to improve the speed of a 12-bit comparator by using three 4-bit comparator blocks instead of one 12-bit comparator. They designed and made a 2-bit digital comparator and compared different ways to make it. They checked how much power it used, how long it took to compare two numbers, and how many transistors were needed. They used a computer program called EDA Tanner to simulate their design. The simulation was done using 32-nanometer technology and a power supply of 0.7 volts. They found out that the new

technique made the 12-bit comparator work 30% faster than the old technique.[2]

seo et.al proposes a new method to compare multiple digital input signals using a simple digital logic function to find the largest or smallest value and position. The downside of this method is that it uses more hardware resources. The method was implemented using Verilog-HDL and tested with a 28 nm CMOS library. The results show that the proposed method can increase the operating clock frequency by up to 3.45 times for 32 input signals, with an increase in hardware resources of up to 4.26 times.[3]

Dibal et al presents designs for both analog and digital comparators. The analog comparator is based on the threshold voltage operation characteristic of floating-gate MOSFETs, which allows for hysteresis-tuning. The proposed analog comparator circuit is then converted into a digital comparator by replacing the input devices with floating-gate MOSFETs, whose drain currents are linear sums of the weighted multiple-input voltages. Simulation results are presented using 1.2pm CMOS technology.[4]

D.N panda et al research A modified transmission gate logic based 2-bit magnitude comparator was designed and compared with other design techniques in terms of power consumption, delay and transistor count. The proposed design consumed 35.77% less power than the existing transmission gate logic-based comparator, while having almost the same power consumption as the pass transistor logic based comparator. The proposed design also had a lower transistor count, resulting in a smaller overall area. However, pass transistor logic had the advantage of providing less power delay product, but with the drawback of not providing full output voltage swing.[5]

Anjuli et al proposes a design for a 2-bit magnitude comparator using different logic styles. The paper compares the different logic styles used in the design using simulation performed in the Tanner EDA Tool at 90nm technology. The main focus of the paper is to analyze and compare the performance of different logic styles in designing the 2-bit magnitude comparator.[6]

Kumari and et al focuses on designing 16, 32, and 64-bit comparators using parallel prefix structure for various applications. The existing design shows that the increase in bit range does not double the delay, memory, LUT, and power to the desired level. The proposed design replaces each element in the parallel prefix structure with a universal logic (multiplexer), leading to reduced power consumption and delay parameters. Results will be compared with the existing and reversible logic designs.[7]

Nandhasri et al proposes a new sorting technique called Snake-like sorting for median filters, which is implemented as a parallel architecture. The proposed architecture requires less comparators for rank ordering, resulting in better performance in terms of power, speed, and area compared to other rank ordering algorithms. The architecture is implemented on parallel and pipelined schemes and is tested on a Spartan 3e device, achieving an operating frequency of 81 MHz and a gate count of 5,640.[8]

Vasanth et al presents the design of a 2-bit binary Magnitude Comparator using Conventional CMOS (CCMOS) logic and Pass Transistor Logic (PTL). The proposed design was compared with 5 existing MC designs and showed a significant improvement in speed and power, resulting in a higher Power Delay Product (PDP). The proposed MC can be considered as a highly effective alternative to existing MC designs due to its significant enhancement in performance.[9]

Sorwar et al proposed a design for a low-power and high-speed 4-bit comparator using transmission gate logic. The authors compare the proposed design with other conventional designs and demonstrate that it has better performance in terms of speed and power consumption. The paper presents a useful design approach for developing low-power, high-speed comparators for various applications.[10]

Maji B proposed the design and implementation of a 4-bit Flash Analog-to-Digital Converter (ADC) using Inverter Threshold Comparator (ITC) in 45nm technology. The authors compare the proposed design with other existing designs and demonstrate that it has better performance in terms of power consumption and speed. The paper provides valuable insights into the design of Flash ADCs using ITC, which can be applied in various low-power applications.[11]

Jitkasem et al proposed a design for a 4-bit comparator using 1-bit full adder modules for area and power efficiency. The authors compare the proposed design with other conventional designs and demonstrate that it has better performance in terms of area and power consumption. The paper presents a useful design approach for developing low power and area-efficient comparators for various applications.[12]

Prajpat et al presents the design and analysis of a low power and high-speed magnitude comparator. The authors compare the proposed design with other existing designs and demonstrate that it has better performance in terms of power consumption and speed. The paper provides valuable insights into the design of low power, high-speed magnitude comparators, which can be applied in various digital circuits and systems.[13]

Devadiga et al system presents a precision multi-input current comparator and its application in an analog median filter implementation. The authors describe the design and performance of the comparator, and demonstrate its application in the filter. The paper provides valuable insights into the design of precision comparators and their use in analog signal processing applications.[14]

Sharma et al demonstrates its logic circuit at the register transfer level and compares the amplitude of many digital input signals. They used an uncomplicated digital logic function to deliver information of the highest or the smallest value or position for particular inputs. They used multiple input signals like 4, 8, 16 and 32.[15]

Power, speed and area are the important parameters in VLSI technology. The research done by Mukherjee et al. has studied power and area as the main factors. They have designed a 12-bit comparator with the help of a multiplexer which gives smaller power dissipation and a lesser area. To enhance the speed of the comparator they have used three 4-bit comparator blocks irrespective of a 12-bit comparator.[16]

Vlassis et al proposed a compact, quad-binary number comparator with 64 bits. The suggested Bit-wise Comparing Logic Chain (BCLC) and Sequential Strobes (SS) technology enables a 64-bit quad binary number comparison in 1.55 nanoseconds which is a 16% improvement over a traditional comparator. The suggested quad binary number comparator uses 0.18 um CMOS technology and consumes 0.015 mm2 with the help of the BCLC and SS scheme, suggested comparator shows a 9% reduction in transistor count and a 13% reduction in area.[17]

Hasia et al presents research on Every comparator was compared using two basic parameters that are area, power and time delay. But every system has a different number of inputs like 4, 16, 32 and so on. Different bits of comparator were designed using different softwares like Xilinx ISE, Cadence, etc. The proposed design has a power delay of 73.98% to 81.15% with differing power supply of 1.1V to 2.0V and with varying temperatures from 50°C to 10°C in the improvement of power delay with 66.76% to 82.97%.[18]

A two-stage, lower power dynamic comparator was proposed by Ahmed et al In this system PMOS transistors were used for the initial and next stages of the comparator. To produce a regulated preamplifier, the second stage was activated or engaged by the first stage with a predetermined delay. After some time, the first stage was deactivated to lower overall power utilization.[19]

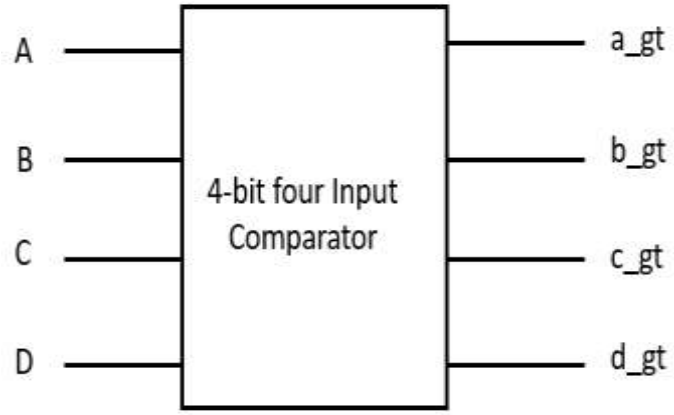
Saiyu ren et al proposed a paper in which 64-bit digital comparator using 90 nm 1.2 V multi-threshold technology with lower power consumption and a smaller area. The design was analyzed using the Cadence-Virtuoso layout editor. Different parameters like power consumption, propagation delay, and transistor count were compared. Results were simulated on the EDA tanner tool for realizing 45- nanometer technology at 0.7 voltages.[20]

Siskos et al proposed a multi-input current comparator. The work was designed on the usage of a multi-input current maximum circuit. Using a feedback circuit, the max circuit's inherent corner error was removed, enhancing the comparator's accuracy. Only one output labeled as 1 was maximum, others were 0. Using double-poly double-metal 2 m CMOS MIETEC technology, a five-input comparator and a three-input median filter were created. [21]

III. PROPOSED SYSTEM

The objective of the proposed design was to develop a new structure to replace the existing cascading structure utilized to compare multiple inputs. The study aimed to design a 4-bit digital comparator that can take four inputs, A, B, C, and D, each consisting of four bits. The proposed design includes four output signals, one for each input, namely a_gt, b_gt, c_gt, and d_gt, with each output consisting of a single bit.

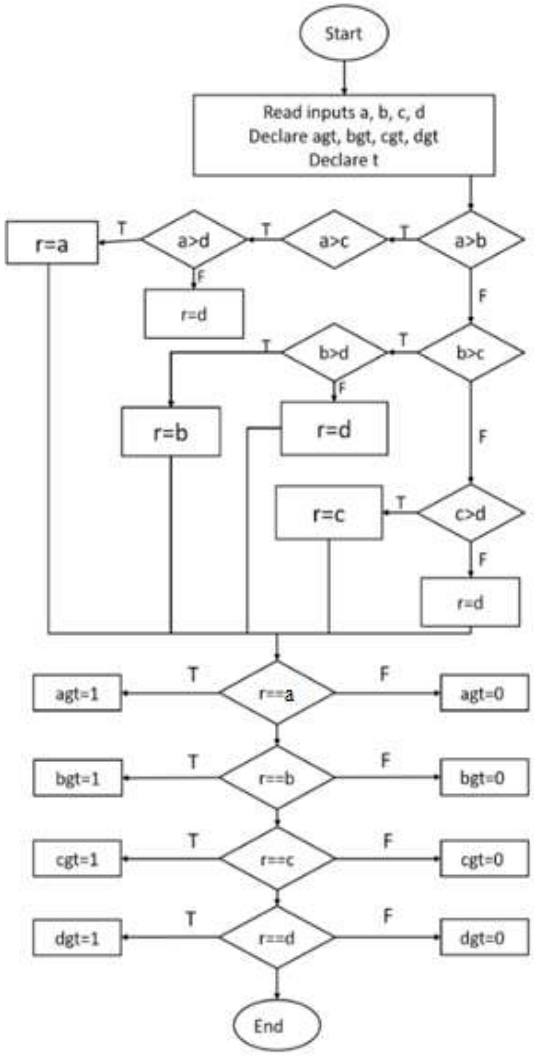
In Figure 1, you can see a diagram of the proposed 4-bit, four-input digital comparator. This design offers an innovative solution to compare multiple inputs with a simplified structure that reduces the number of stages needed to compare the inputs. With this design, the outputs can be generated directly without the need for additional logic gates or circuits.



(Fig 1. Block diagram of 4 bit input comparator)

A. Flow diagram for algorithm 1

The proposed algorithm is designed to find the greatest number among four inputs, A, B, C, and D, where each input consists of four bits. The algorithm utilizes four output variables, a_gt, b_gt, c_gt, and d_gt, each containing a single bit, to indicate which input variable is the greatest. The algorithm starts by declaring the input and output variables and a register variable, r, to hold the greatest value among the inputs. Then, in the always block, the algorithm compares each input variable with the rest to determine which one is the greatest. If a is greater than b, the algorithm checks if a is greater than c, then if a is greater than d, and sets r to a if it is. If a is not greater than d, r is set to d. Similarly, if a is not greater than c, the algorithm checks if b is greater than c, and so on until it finds the greatest input, which is stored in the register r. After finding the greatest input value, the algorithm checks which input variable matches the value of r and sets the corresponding output variable to 1. If the input variable doesn't match the value of r, the corresponding output variable is set to 0.

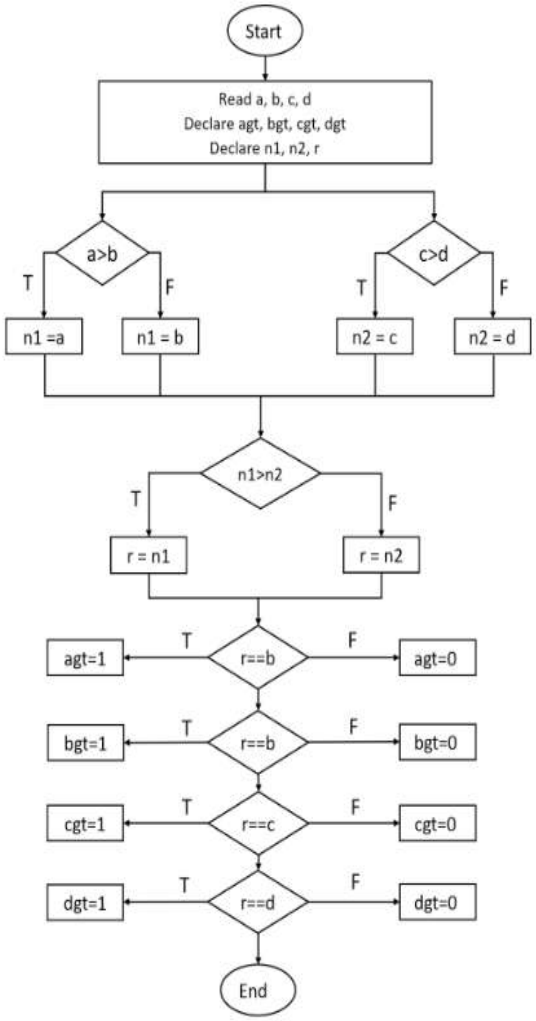


(Fig 2. flow diagram of algorithm 1)

B. *Flow chart for algorithm 2*

The proposed algorithm finds the maximum value among four input variables, a, b, c, and d. It uses four output variables, agt, bgt, cgt, and dgt, to indicate which input variable has the maximum value. Three registers are declared, r, n1, and n2, to store the maximum values.

The algorithm compares a and b, assigns the maximum value to n1, and compares c and d, assigns the maximum value to n2. Then, it compares n1 and n2, assigns the maximum value to r, and checks which input variable is equal to r. Finally, it sets the corresponding output signal to 1 and all others to 0. This algorithm is efficient and can be implemented in digital systems with minimal hardware resources. It can also be extended to handle more input variables.

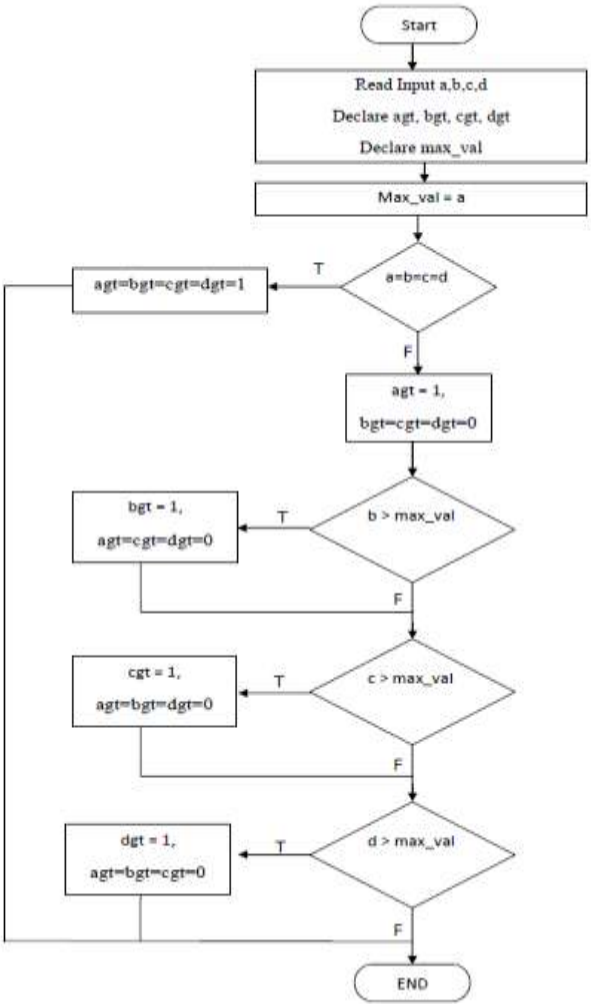


(Fig 3. Flow diagram for algorithm 2)

C. *Flow chart for algorithm 3*

The proposed algorithm is designed to find the maximum value among four input variables, a, b, c, and d, and indicate which input variable has the maximum value using four output variables, agt, bgt, cgt, and dgt. The algorithm begins by declaring the input and output variables and a register, max_val, with a width of 4 bits to store the maximum value among the inputs.

The algorithm then uses the "always" block to execute a series of steps whenever any of the inputs (a, b, c, or d) changes. Firstly, it assigns the value of a to max_val, then checks if all the inputs are equal, and assigns 1 to all the output signals in that case. If not, it compares the remaining input variables to the current maximum value stored in max_val, assigns the new maximum value to max_val, and sets the corresponding output signal to 1 and all others to 0. Finally, the "always" block ends.



(Fig 4. Flow diagram for algorithm 3)

IV. NOVELTY

The novelty of our project lies in the implementation and comparison of three different algorithms for finding the maximum value among four input variables. We have evaluated the performance of these algorithms based on various parameters such as delay, power consumption, number of comparators and latches used, and memory usage. Our study provides insights into the design and optimization of comparators with multiple inputs. The comparison of the algorithms helped us determine the optimal trade-off between the various parameters, allowing us to identify the algorithm that performs better than the others for the given requirements. Our findings can be useful for researchers and engineers working in the field of digital systems design, as the proposed circuits can be utilized in various applications that require the identification of the largest value among multiple inputs, From the results, we can conclude that Algorithm 3 outperformed the other two algorithms. It consumed less memory, did not use any latches, and had a lower number of comparators. Additionally, its hardware requirement was minimal, making it a more efficient and cost-effective option. Therefore, we suggest that Algorithm 3 be preferred for finding the maximum value among multiple inputs in digital systems.

V. RESULTS AND ANALYSIS

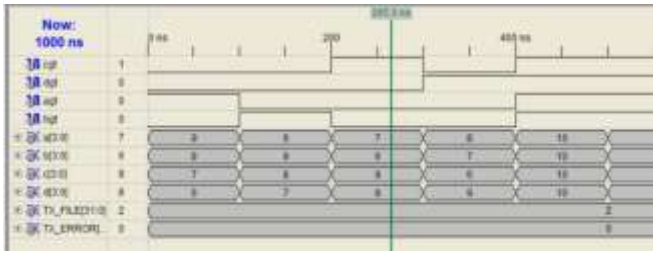
In this study, we have implemented three different algorithms in Xilinx ISE to find the maximum value among four input variables. The first algorithm uses three registers to store the maximum value and compare it with each input variable. The second algorithm uses a priority encoder to determine the maximum value, and the third algorithm uses a single register and comparator logic to determine the maximum value.

To evaluate the performance of these algorithms, we calculated their delay, total power, latches and the number of comparators and memory used.

TABLE I. COMPARISON PARAMETERS OF ALGORITHMS

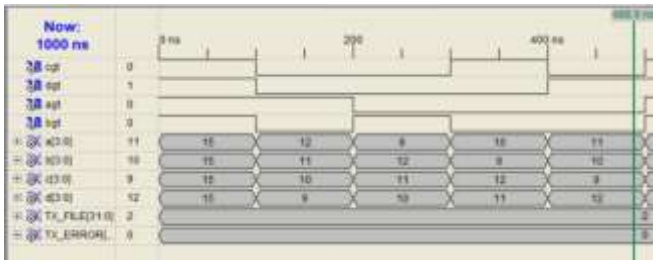
	Algorithm 1	**Algorithm 2**	**Algorithm 3**
Comparator	10	7	6
Latches	4	0	0
Delay	9.666ns	22.098ns	20.365ns
Total Power	37.00mW	37.00mW	37.00mW
Total Memory Usage	157636KB	157316KB	157380KB

According to Table 1, Algorithm 1 has 10 Comparators, 4 Latches, a 9.666 ns Delay, and utilizes 157636 KB of Memory. Additionally, we can see from Fig. 4 below that Algorithm 1 responds to inputs and outputs by going up when 1 and down when 0.



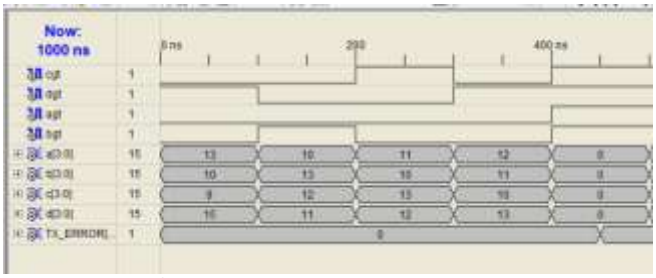
(Fig 4. Analysis of algorithm 1)

Table 1 shows that algorithm 2 includes seven comparisons, does not employ latches, has a delay up to 22.098 ns more than algorithm 1, and requires 157316 kb of total memory. The result is displayed in Fig. 5 with various inputs and the desired output.



(Fig 5. Analysis of algorithm 2)

From the results, we can see that Algorithm 3 has the lowest number of comparators and power consumption, while Algorithm 1 uses the most comparators and latches. However, it is essential to note that the number of comparators and latches used can vary based on the implementation details of each algorithm. For algorithm 3 it uses total comparisons that is less than other two also when talking about delay then algorithm 1 uses minimum delay than others upto 9.666ns. Table 1 shows this all-comparison results of the three algorithms detailly.



(Fig 6. Analysis of algorithm 3)

CONCLUSION

We presented and compared three algorithms for finding the maximum value among four input variables. We implemented these algorithms in Xilinx ISE and evaluated their performance based on various parameters, including delay, power consumption, number of comparators and latches used, and memory usage. From the results, we can see that Algorithm 3 has the lowest number of comparators and power consumption, while Algorithm 1 has the highest number of comparators and latches. However, the number of comparators and latches used can vary based on the implementation details of each algorithm. In terms of delay, Algorithm 1 performed better than the other two algorithms, with a delay of up to 9.666ns.

Our study demonstrates that simple comparison operations can be used to identify the largest value among multiple inputs, and the proposed algorithms can be implemented with minimal hardware resources. The designs can be extended for comparators with more than four inputs, and we hope that the proposed circuits can be utilized in a range of digital systems. Overall, the choice of algorithm for finding the maximum value among multiple inputs should depend on the specific requirements and constraints of the application.

REFERENCES

[1] Mukherjee, D. N., Panda, S., & Maji, B. (2018). Performance evaluation of digital comparator using different logic styles. *IETE Journal of Research*, *64*(3), 422-429

[2] Mukherjee, D. N., Panda, S., & Maji, B. (2022). A novel design of 12-bit digital comparator using multiplexer for high speed application in 32-nm cmos technology. *IETE Journal of Research*, *68*(2), 1350-1357.

[3] Seo, Y. H., Park, S. H., & Kim, D. W. (2019). High-level hardware design of digital comparator with multiple inputs. *Integration*, *68*, 157-165.

[4] Dibal, P. Y. Design of a 4-bit Magnitude Comparator using Simulink. *Arid Zone Journal of Engineering, Technology and Environment*, *9*.

[5] Mukherjee, D. N., Panda, S., & Maji, B. (2016). Optimization of digital comparator using transmission gate logic style. *Int. J. Adv. Res. Eng. Technol*, *7*(4), 06-16.

[6] Anjuli, S. A., & Satjajit, A. (2013). 2-bit magnitude comparator design using different logic styles. *International Journal of Engineering Science Invention*, *2*(1), 13-24.

[7] KUMARI, P. S., & VANI, A. J. (2014). Realization of Scalable Digital Comparator using Multiplexers.

[8] Nandhasri, K., & Ngarmnil, J. (2001, May). Designs of Analog and digital Comparators with FGMOS. In *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196)* (Vol. 1, pp. 25-28). IEEE..

[9] Vasanth, K., Kavirajan, A. A. F., Ravi, T., & Raj, N. (2014). A Novel 8 bit digital comparator for 3x3 fixed kernel based modified shear sorting. *Indian journal of science and technology*, *7*(4), 452.

[10] Sorwar, A., Sojib, E. A., Dipto, M. A. Z., Rangon, M. M. T., Chowdhury, M. S. A., & Siddique, A. H. (2020, July). Design of a high-performance 2-bit magnitude comparator using hybrid logic style. In *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-5). IEEE..

[11] Maji, B. (2017).digital comparator using different logic styles. *IETE Journal of Research*, *64*(3), 422-429

[12] Nandhasri, Krissanapong, and Jitkasem Ngarmnil. "Designs of Analog and digital Comparators " In ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No. 01CH37196), vol. 1, pp. 25-28. IEEE, 2001.

[13] Prajpat, Govind, Akhilesh Joshi, Aman Jain, Kumkum Verma, and Sanjay Kr Jaiswal. "Design of low power and high speed 4-bit comparator using transmission gate." In 2013 International Conference on Machine Intelligence and Research Advancement, pp. 379-382. IEEE, 2013.

[14] Devadiga, Ramyashree, and Satheesh Rao. "Design of 4-Bit Flash ADC Using Inverter Threshold Comparator in 45nm Technology." In 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 978-982. IEEE, 2018.

[15] Sharma, Anjali, and Pranshu Sharma. "Area and power efficient 4-bit comparator design by using 1-bit full adder module." In 2014 International Conference on Parallel, Distributed and Grid Computing, pp. 1- 6. IEEE, 2014.

[16] Singh, Pranay, and Pramod Kumar Jain. "Design and Analysis of Low Power, High Speed 4-Bit Magnitude Comparator." In 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), pp. 1680-1683. IEEE, 2018.

[17] Vlassis, S., and S. Siskos. " comparator and its application to analog median filter implementation." Analog Integrated Circuits and Signal Processing 34 (2003): 233-245.

[18] Hsia, Shih-Chang. "A high speed multi-input comparator with clocking-charge based for low-power systems." In The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications, 2003. Proceedings., pp. 130-133. IEEE, 2003.

[19] Ahmed, Syed Ershad, S. Sweekruth Srinivas, and M. B. Srinivas. "A Hybrid Energy Efficient Digital Comparator." In 2016 29th International Conference on VLSI Design and 2016 15th International Conference on Embedded Systems (VLSID), pp. 567-568. IEEE, 2016.

[20] Vijaya Krishna Boppana, N. V., and Saiyu Ren. "A low-power and area-efficient 64-bit digital comparator." Journal of Circuits, Systems and Computers 25, no. 12 (2016): 1650148

[21] S. Siskos. "Precision multi-input current comparator and its application to analog median filter implementation." Analog Integrated Circuits and Signal Processing 34 (2003): 233-245.