Turkish Journal of Computer and Mathematics Education Vol.12 No. 11 (2021), 2649- 2661
Research Article

High-Speed And Area-Efficient 16, 64 -Bit Digital Comparator

[1.] Dr. K. Ilamathi, [2.] Dr. Bennila Thangammal, [3]. P. Santhoshini, [4.] Raya Hemesh [5]. Tirumalasetty Sri Sai Ram Charan [6.] Yanamalamanda Vignesh

[1]ASSOCIATE PROFFESSOR, Dept of Electronics and Communication Engineering, R.M.D. Engineering College, Chennai
2ASSOCIATE PROFFESSOR, Dept of Electronics and Communication Engineering, R.M.D. Engineering College, Chennai
3ASSISSTANT PROFFESSOR, Dept of Electronics and Communication Engineering, R.M.D. Engineering College, Chennai
[4, 5,6] Dept of Electronics and Communication Engineering, R.M.D. Engineering College, Chennai

Article History: Received: 11 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 10 May 2021

ABSTRACT: An area-efficient N-bit digital comparator with high operating speed and low-power dissipation is presented in this work. The proposed comparator structure consists of two separate modules. The first module is the comparison evaluation module (CEM) and the second module is the final module (FM). Independent from the input operand bit widths, stages present in CEM involve the regular structure of repeated logic cells used for implementing parallel prefix tree structure. The FM validates the final comparison based on results obtained from the CEM. The presence of regular very large-scale integration topology in the proposed structure allows the analytical derivation of the area in terms of total number of transistors present in the design and total delay encountered in input–output flow as the function of input operand bit width. The proposed comparator is designed using 180nm technology in tanner tool and the results are observed.

Keywords **–** Tanner EDA Tool, Comparison evaluation module, Final Module, (VHSIC) Hardware Description Language, Central Processing Unit, Error Redundancy Check, Multiple Input Multiple Output, Microcontroller Units.

1. Introduction

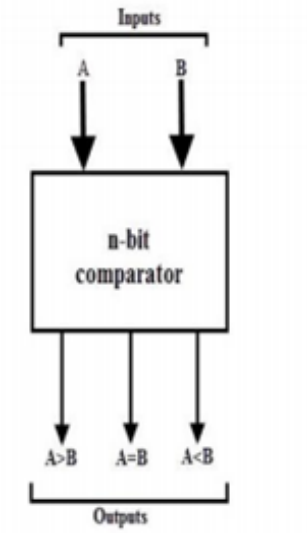
Digital comparator is the fundamental design element used for the applications, in which the final results are based on the output obtained from the computation involving comparison as an activity. There are wide range of applications, which involve scientific computations (digital image processing, pattern recognition/ matching, arithmetic sorting, data compression and digital neural network [1–3]) and test circuit applications (built-in self-test circuits, signature analysers and jitter measurement [4–5]) consisting of comparator as the basic design element. The optimized design of comparator is used as the key component in the general-purpose computer architecture for developing the memory addressing logic, queue buffers, test circuits etc. [6–8]. Extensive use of comparator logic in various computation-based designs necessitates optimization in terms of area, power and speed. Some of the comparator designs use dynamic logic to achieve low-power consumption but limitations of low-speed and poor-noise margin make the dynamic design rather challenging. The other designs use subtractors in the form of flat adder components along with custom logic circuits [9–13] to implement comparison process for wider bit operands but these designs give slower response and area intensive arrangement [14– 16]. The improvement in the scalability and reduction in the comparison delay has been achieved in hierarchical prefix tree structure-based comparator that composed of 2-bit comparators at each level [17]. However, for the wide input operands, these structures maybe prohibitive due to prolonged delay and power consumption arising from log2 N comparison levels. Improvement in some of the limiting factors of the parallel prefix tree structure such as area and power consumption can be achieved by using two input multiplexers at each level and generate–propagate logic at the first level. However, the comparator structure has very high power consumption since every cell remains in active state irrespective of the applied operand values.

A digital comparator or magnitude comparator is a hardware electronic device that has two binary inputs, and determines whether one number is greater than, less than or equal to the other number. The comparators are widely used in Central Processing Units (CPUs), Micro Controller Units (MCUs) which is a crucial data path element of image and signal processing architectures.

In the last few years, the design of high-speed and low power binary comparators has received a great deal of attention. Data comparison is needed in digital systems while performing arithmetic or logical operations. This comparison determines whether one number is greater than, equal, or less than the other number. A digital comparator is widely used in combinational system and is specially designed to compare the relative magnitudes of binary numbers.

These are also available in IC form with different bit comparing configurations such as 4-bit, 8-bit, etc. More than one comparator can also be connected in cascade arrangement to perform comparison of numbers of longer lengths. Whenever we want to compare the two binary numbers, first we have to compare the most significant bits. If these MSBs are equal, then only we need to compare the next significant bits. But if the MSBs are not equal, then it would be clear that either A is greater than or less than B and the process of comparison ceases. For example, the two 2-bit number are A = A1A0 and B=B1B0. If A1 is not equal to B1, then it is clear that A is greater than B for A1 =1 & B1= 0 or else A is less than B for A0= 0 & B0 =1. At this stage the process of comparison ceases. If the MSBs are equal, i.e., A1=B1 only then we need to compare the next significant bits A0 and B0 and decide whether the number is greater than, less than or equal. So, the comparator produces three outputs as L, E and G corresponds to less than, equal and greater than comparisons.

A magnitude digital comparator is a combinational circuit that compares two digital or binary numbers (consider A and B) and determines their relative magnitudes in order to find out whether one number is equal, less than or greater than the other digital number. Three binary variables are used to indicate the outcome of the comparison as A>B, A<B, or A=B. The below figure shows the block diagram of a n-bit comparator which compares the two numbers of n-bit length and generates their relation between themselves.



Nbit comparator

II RELATED WORK

Magnitude comparators are mostly utilized in microcontrollers and CPUs to address data comparison, register and perform all other arithmetic operations. Magnitude comparators are implemented in many devices and every auto-turn-off device is surely designed using a comparator.

A comparator is a decision-making tool and it holds the ability to be executed in numerous control devices. Accepting two binary numbers as input (A and B), data comparison through magnitude comparators produces the output to indicate equality (A=B), logic 1 in two conditions when (A>B or A<B).

Binary comparators are found in a wide variety of circuits, such as microprocessors, communications systems, encryption devices, image processing, 3D graphics and many others. A faster, more power efficient, or more compact comparator would be an advantage in any of these circuits. use of comparators in high-performance systems places a great importance on the extensive performance and power consumption optimizations. A comparator basically involves comparison of two n-bit bit numbers is a critical operation for almost all digital systems. A comparator compares two n-bit values to determine which is greater, or if they are equal. In general, it is used to compare two inputs. Comparators are broadly classified into Analog and Digital comparators. However, in this brief what is concerned is about the digital comparator. The digital comparator is further classified into Total (Full) comparators and Equality comparators. In full comparators, given two n-bit binary numbers A and B, they are able to separately recognize the three possible conditions i.e. A > B, A < B and A = B. In equality comparators, as the name suggests, they only indicate equality when both the inputs are equal. Comparators find their applications in many Digital Signal Processors. It has been an important logic block in an ALU and have extensive applications such as decoding of x86 instructions [1]. It also finds applications in MIMO (Multiple Input Multiple Output) decoding algorithms require

extensive iterations of binary number comparison. In recent year, high speed & low power device designs have emerged as principal theme in electronic industry due to increasing demand of portable devices. This tremendous demand is due to popularity of battery-operated portable equipment such as personal computing devices, wireless communication, medical applications etc. Demand & popularity of portable electronic devices are driving the designers to strive for higher speed, smaller power consumption and smaller area. The logic style used in logic gates basically influences the speed, size, power dissipation, and the wiring complexity of a circuit. Circuit size depends on the number of transistors and their sizes and on the wiring complexity. The wiring complexity is determined by the number of connections and their lengths. All these characteristics may vary considerably from one logic style to another and thus proper choice of logic style is very important for circuit performance.

Recently, the market demand for mainly three things: low power, high speed and less area for designing the circuit So intensified research is going on low power, high speed embedded systems used in mobiles, laptops etc has led to scale down technology to nano regimes, allow to implement more functionality on single chip,

The magnitude comparator is a fundamental arithmetic component of digital system, Now-a-days, most important problem in computer science is sorting. Sorting is useful in fundamental process such as in communication and computing systems. Sorting of data problem can be solved by comparator, where comparator play an important role in the areas of parallel computing, multiprocessing and multi-access memories, A magnitude comparator is also used in Digital Signal Processors (DSP) for data processing, microprocessor for decoding instruction and microcontroller for controlling temperature of furnace in industry. The traditional method to implement the comparator is by flattering the logic function directly, but this method is only suitable for the comparator with a smaller number of inputs [1], When large number of inputs is applied, circuit complexity increases drastically and the operating speed is degraded accordingly. Alternative way to implement the comparator is by using a parallel adder. In this method, the adder has become the major factor for reducing the operating speed.

Contemporary microprocessors, particularly superscalar CPUs, rely on the use of a significant amount of associative matching logic to support register renaming, out–of–order execution and virtual memory mechanisms. The traditional comparator circuit (also known as a pull–down comparator), dissipates energy on a mismatch in the input arguments (comparands). The output is precharged, and pulled down on a mismatch in any bit position during the evaluation phase, causing energy dissipation. Notice that the effective output loading of traditional comparators is high: this is equal to the diffusion capacitances of 2C n–transistors plus the load capacitance, where C is the number of bits compared. This results in considerable power dissipation in the case of a mismatch. As mismatches are much more frequent than matches in some components of superscalar paths that make use of associative addressing, the use of traditional comparators is not an energy–efficient solution.

This CMOS architectures for electronic circuit design has dominated the field of circuit design for quite some time. CMOS circuits are not only power efficient but also have good switching characteristics making them a great choice for combinational and sequential circuit designing. Moreover, the fabrication techniques of CMOS transistors have evolved over the period of time and matured into an efficient process.

But there is an inherent problem with CMOS logic design. For every logic to be implemented in CMOS architecture it needs large number of transistors which not only increases the chip area but also increases power consumption and power dissipation [1, 2, 3]. Therefore, a need has aroused to reduce the transistor count without affecting the logic functionality and Pass transistor logic (PTL) has to be a great tool in materializing this effort. Moreover, PTL utilizes NMOS and PMOS transistor for logic implementation whose fabrication technology is even more matured than CMOS fabrication technology.

The evolution of CMOS technology has laid down the design of Universal logic gates at physical level of abstraction.

III METHODOLOGY

The Demand and popularity of portable electronics is driving designers to strive for smaller silicon area, higher speeds, longer battery life, and more reliability. Power is one of the premium resources a designer tries to save when designing a system. Full adders are fundamental units in various circuits, especially in circuits used for performing arithmetic operations. such as compressors, comparators, parity checkers, and so on. Full adders are often in the critical paths of complex arithmetic circuits for multiplication and division. These in turn form the core of any system and thereby influence the overall performance of the entire system. Enhancing the performance of the full adder can significantly affect the system performance.

The DataPath consumes roughly 30% of the total power of the system. Adders are an extensively used component in data paths and, therefore, careful design and analysis is required for these units to obtain optimum performance. At the circuit level, an optimized design is desired to avoid any degradation in the output voltage, consume less power, have less delay in critical path, and be reliable even at low supply voltage as we scale towards deep sub micrometre. Good driving capability under different load conditions and balanced output to avoid glitches is also an important

virtue. Since the full-adder cells are duplicated in large numbers, layout regularity, and interconnect complexity are also important. Several logic styles have been used in the past to design full adder cells. Each design style has its own merits and demerits. Classical designs of full adders normally use only one logic style for the whole full-adder design. One example of such design is the standard static CMOS full adder. This full adder is based on regular CMOS structure with conventional pull-up and pull-down transistors providing full-swing output and good driving capabilities. The main drawback of static CMOS circuits is the existence of the PMOS block, because of its low mobility compared to the NMOS devices. Therefore, the PMOS devices need to be sized up to attain the desired performance. The input capacitance of a static CMOS gate is large because each input is connected to the gate of at least a PMOS and a NMOS device. The evolution of CMOS technology has laid down the design of Universal logic gates at physical level of abstraction. With up gradation to CMOS; Pass transistor logic, Transmission gates, Domino logic, Ratioed and Dynamic logic supports any design with comparable Power consumption, Area that comprises of number of transistors and Power supply requirements. With decrease in technology node, the power supply requirements also reduce to greater extent and thereby logic design with low power availability becomes need of time. Raising in number of transistors on chip power consumption in VLSI (very large scale integration) systems are also raised, because of Demand and vogue of portable consumer electronics are making the IC designers to aspire for high speed, high battery durability, low power chips and more reliability. If low power techniques are not there in the circuit means it will suffer from low battery life. Power consumption can be reduced by minimizing the supply voltage; the designer may able to attain low power consumption but has to face trade-off between other requirements like speed and reliability. Low power consumption is one of the important design criteria for all IC designers at all levels of design along with delay and area considerations. The exclusive-OR(XOR) and exclusive-NOR(XNOR) gates are popular in microprocessors and are the basic building block of many arithmetic and encryption circuits especially circuits used for performing arithmetic operations, like full adders, compressors, comparators, parity checkers ,error detectors and correctors, and adders, The main concern is to design a circuit with XOR/XNOR gates is to obtain low power consumption and delay in the critical path and full output swing with less number of transistors. An eight transistor XOR-XNOR (XE) with cross-coupled p-MOS, is used to design the proposed comparator sub-block. This design is good for low power and small area. 64-bit comparator utilizes 64 XE blocks to drive next stage comparator sub-block logic. We propose a novel XOR– XNOR circuit using eight transistors that generates XOR and XNOR outputs simultaneously. The proposed XOR–XNOR circuit is based on complementary pass-transistor logic using only one static inverter instead of two static inverters as in the regular CPL style XOR circuit. The proposed circuit is shown in Fig. 2. The first half of the circuit utilizes only nMOS pass transistors for the generation of the outputs. The cross-coupled PMOS transistors guarantee full swing operation for all possible input combinations and reduce short-circuit power dissipation. The circuit is inherently fast due to the high mobility NMOS transistors and the fast differential stage of cross-coupled PMOS transistors. This is another reason for speed degradation of static CMOS gates. Another conventional adder is the complementary pass-transistor logic (CPL). It provides high-speed, full-swing operation and good driving capability due to the output static inverters and the fast differential stage of cross-coupled pMOS transistors. But due to the presence of a lot of internal nodes and static inverters, there is large power dissipation. The layout of a CPL cell is also not as straightforward as a static CMOS cell due to its irregular transistor arrangement. The dynamic CMOS logic style provides a high speed of operation because the logic is constructed with only high mobility Nmos transistors. Also, due to the absence of the pMOS transistors, the input capacitance is also low, thus enhancing the speed of operation. However, it has several inherent problems such as charge sharing and high clock load. It has higher switching activity and lower noise immunity. It consumes a large portion of the power in driving the clock lines. Moreover, dynamic logic style is more susceptible to leakage. Due to these reasons, we do not include dynamic logic style in our discussions in this paper. Some other full-adder designs include transmission-function full adder (TFA) and transmission gate full adder. These designs are based on transmission-function theory and transmission gates, respectively. These adders are inherently low power consuming. These logic styles are good for designing XOR or XNOR gates. The main disadvantage of these logic styles is that they lack driving capability. This is attributed to the fact that the inputs are coupled to the outputs. When TGA or TFA are cascaded, their performance degrades significantly. The remaining adder designs use more than one logic style for their implementation. We call this the hybrid-CMOS logic design style. Examples of adders built with this design style are DB cell, NEW14-T adder and hybrid pass logic with static CMOS output drive full adder and new-HPSC adder. These designs exploit the features of different logic styles to improve upon the performance of the designs using single logic style. All hybrid designs use the best available modules implemented using different logic styles or enhance the available modules in an attempt to build a low power full-adder cell. Generally, the main focus in such attempts is to reduce the numbers of transistors in the adder cell and, consequently, reduce the number of power dissipating nodes. This is achieved by utilizing intrinsically low power consuming logic styles like TFA or TGA or 23 simply pass transistors. In doing so, the designers often trade off other vital requirements such as driving capability, noise immunity, and layout

complexity. Most of these adders lack driving capabilities as the inputs are coupled to the outputs. Their performance as a single unit or in small chains is good but when large adders are built by cascading these 1-b full-adder cells, the performance degrades drastically. The performance degradation can be handled by inserting buffers in between stages to enhance the delay characteristics. However, this leads to an extra overhead and the initial advantage of having a lesser number of transistors is lost. TableI indicates the functioning of the proposed circuit more clearly. For any input vector, the PMOS transistors are switched ON by a good and, therefore, avoid any static power dissipation. Owing to the lower Vthn and high electron mobility of the NMOS transistors, the circuit has a faster response as compared to the previous circuit.

IV PROPOSED SYSTEM

The working principle of conventional comparison is shown in Fig. 1, where the operands A and B have unequal most significant bit (MSB) bits. Since the first unequal bits of operands A and B encountered is well-sufficient to decide the outcome of the comparison between the two operands, remaining bit positions are ignored for comparison. The comparison process used for comparing N-bit operands starts comparison from (N–1)th bit (or MSB bit) and proceeds toward the comparison of (N–2)th bit (or least significant bit (LSB)) if and only if the MSB bits of the two operands are equal.

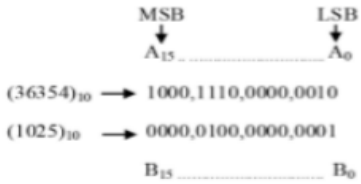


Fig. 1 Comparison between two 16-bit operands

As shown in Fig. 2, the comparison process continues to compare the bit pairs obtained from the operands until it gets an unequal pair of bits on its way toward the LSB bit position.

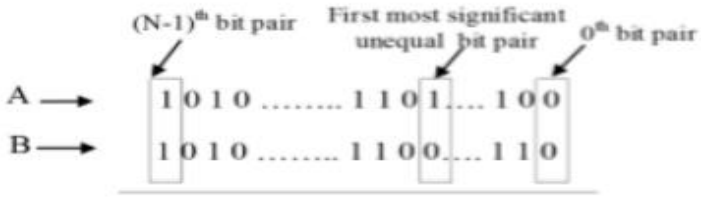


Fig. 2 Comparison between two N-bit operands

The unequal bit pair (X) and equal bit pair (E) are realised as

The flowchart of the algorithm used for the implementation of the proposed N-bit digital binary comparator is shown in Fig. 3. The symbols and their descriptions used in the proposed design are listed in Table 1. The two N-bit input operands A and B are selected for the comparison and are checked if the operands are equal or not equal by performing the bitwise comparison. If the result of comparison comes out as ‘equal’, then the proposed comparator drives the output logic AEB to logic 1. If the comparison result of the operands comes out as ‘unequal’, then the pre encoder output bits are checked from MSB to LSB. The output logic AGB or ALB goes to logic 1 based on the results of pre encoder. The proposed algorithm reduces the superfluous switching activities occurred during comparison operation, which further limits the dynamic power consumption of the proposed comparator.

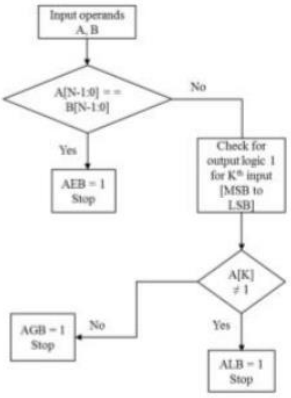


Fig. 3 Flowchart of the proposed N-bit digital comparator

The proposed N-bit digital comparator is shown in Fig. 4. The two N-bit input operands A and B are selected for the comparison and are checked if the operands are equal or not equal by performing the bitwise comparison.

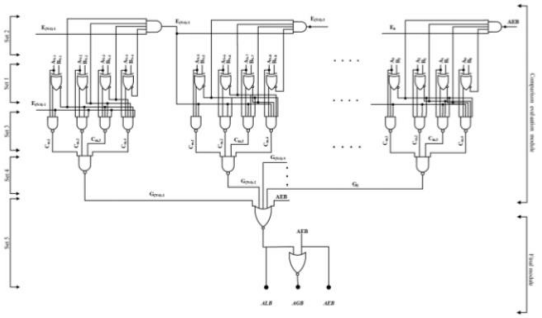


Fig. 4 Proposed digital N-bit comparator

For performing a comparison between two N-bit binary operands, the proposed structure is divided into the comparison evaluation module (CEM) and final module (FM). These modules serve as a high-level and low-level architectures. The CEM incorporates parallel prefix tree structure that is intended for performing a bitwise comparison of two N-bit operands A and B depicted by AN −1AN−2, …, A0 and BN−1BN−2, …, B0. To explore the regularity of the proposed comparator for arbitrary bit widths, two operands A and B are applied into 4-bit partitions as AN−1AN−2AN−3AN−4, …, A3A2A1A0 and BN−1BN−2BN−3BN−4, …, B3B2B1B0. The complete process of comparison is divided into five sets, in which CEM contains sets 1–4 and FM contain only set 5. All the sets in the design are placed in four hierarchal prefix orders according to their functionality; therefore, the output of each set in this approach serves as the input of another set with an exclusion of set 1, whose outputs act as the inputs of sets 2 and 3. In set 1, bitwise comparison of two N-bit binary operands is carried out by the novel EX–OR–NOR cell. The proposed structure of EX–OR–NOR cell shown in Fig. 5 is based on the pass transistor logic and CMOS logic.

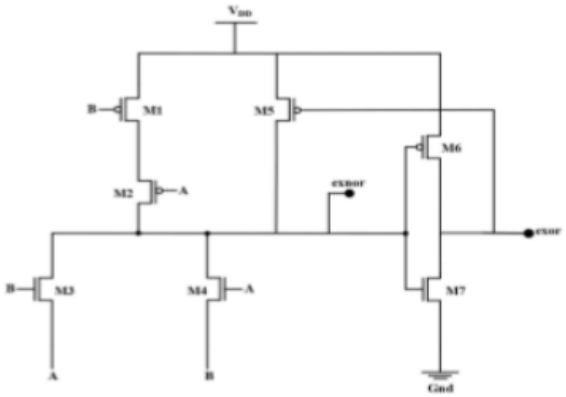


Fig. 5 Proposed XOR Gate

It uses seven transistors for EX– OR and EX–NOR operations as compared with the conventional eight transistors model [32]. The transistor M5 is used to obtain full output voltage swing of EX–NOR operation as shown in Fig. 6. The six transistors model has also been reported in [39] but it gives limited output voltage swing when applied input operands are (0, 0) or (1, 1). Optimum aspect ratios of the seven transistors (M1−M7) consisting of four P-channel MOS (PMOS) and three N channel MOS transistors of the proposed EX–OR–NOR cell are carried out to avoid the universal drive constraint faced by the pass transistor logic. The novel structure uses a PMOS transistor in the feedback to maintain the logic level on the EX–NOR output terminal and the CMOS logic to boost up the output for achieving the full voltage swing on the EX–OR output terminal. The outputs of novel EX–OR–NOR cells provide the termination and comparison bits intended for sets 2 and 3 structures.

The operation of the novel EX–OR–NOR cell is described as
set1: TK = AK ⊙ BK (3)
set1:DK = AK ⊕BK (4)

where TK indicates equal bit pair, DK indicates unequal bit pair of operands A and B and K is an integer, which varies in the range of $0 \leq K \leq N–1$.
Set 2 comprises of cells, which operates on the termination bits (TK) obtained from set 1. The logic cells present in set 2 combine the termination bits obtained from the nibble partitions (partition used for the comparison of every 4 bit of the operands starting from the MSB) present in set 1 and the outputs obtained from the preceding AND-type logic cells present in the same level of set 2. Equal flags E[(N/4) −1] to E0 generated from set 2 control the switching activities of the next subsequent partitions of set 3.
Comparison request from set 2 generates if and only if all the results obtained from the bitwise comparison performed by preceding cells of set 1 are ‘equal’; otherwise, termination bits as logic 0 will be generated. The operation of set 2 is expressed as
set2: Em−1 = ∏ m=1 N/4 −1 T4m+3T4m+2T4m+1T4mEm (5)

set2: AEB (when m=0) =∏T3T2T1T0E0 (6)
where Em−1, for m = 1 to [(N/4) −1] represent the equal flags of set 2. Set 3 includes cells, which combine the outputs obtained from sets 1 and 2. The number of inputs increases in the ascending order from left to right for each cell in their respective partition and ending with the maximum fan-in of six. The combination of sets 1 and 3 architectures forms the pre-encoder structure. If most significant unequal bits are received in the comparison process of two operands, then the output bits obtained from sets 1 and 2 allow the termination of the subsequent bitwise comparison activity of the logic cells present in set 3. Computation process of the cells present in each partition of set 3 can be written as

Cm, 1 =COMP ∏ m=0 N/4 −1 EmA4m+3D4m+3 (7)
Cm, 2 =COMP ∏ m=0 N/4 −1 EmA4m+2D4m+2T4m+3 (8)
Cm, 3 =COMP ∏ m=0 N/4 −1 EmA4m+1D4m+1T4m+3T4m+2 (9)
Cm, 4 =COMP ∏ m=0 N/4 −1 EmA4mD4mT4m+3T4m+2T4m+1 (10)

where Cm,1, Cm,2, Cm,3 and Cm,4 {for m = [(N/4) −1] to 0} represent outputs of NAND-type logic cells for the m^{th} partition of set 3. Set 4 contains NAND-type logic cells, which receive the inputs from set 3 and set 4 requires (N/4) cells to combine the outputs from each partition of set 3. The complete operation can be written as

set4: Gm =COMP ∏ m=0 N/4 −1 Cm, 1Cm, 2Cm, 3Cm, 4 (11)

where Gm {for m = [(N/4) −1] to 0} represent the outputs of the mth logic cell. Set 5 contains two NOR-type logic cells to decide the final results of the proposed digital comparator in terms of ‘ALB’ and ‘AGB’. First NOR gate uses outputs of set 4 and ‘AEB’ as inputs to decide ‘ALB’, whereas second NOR gate uses the output of first NOR gate and ‘AEB’ as inputs to decide ‘AGB’. The computation process of set 5 is given by

set5:ALB=COMP ∑GN/4 −1…G0(AEB) (12)
set5: AGB=COMP ∑ ALB (AEB) (13)

To explain the process of the proposed methodology, two input operands A = 1010101010101010 and B = 1001100110011001 are chosen for 16-bit comparison and the pictorial view of the process is illustrated in Fig. 7.

The complete process is divided into five sets. Set 1 includes a bitwise comparison of input operands for the examination of equal and unequal bit pairs. The outputs of set 1 are '1100110011001100' and '0011001100110011' using novel EX– OR–NOR cells. From the output, it is clear that the first two bit pairs A15 B15 = '11' and A14 B14 = '00' are equal bit pairs, whereas the third bit pair A13 B13 = '10' is the unequal most significant bit pair. The set 2 examines the presence of equal bit pairs but due to the presence of unequal most significant bit pair, the outputs of set 2 are E3 = '1', E2 = '0', E1 = '0', E0 = '0' and AEB = '0' due to logical AND operation. Since, unequal most significant bit pair A5B5 = '10' is encountered during the comparison process, the output of set 3 is obtained as C3,1C3,2C3,3C3,4,C2,1C2,2C2,3C2,4,C1,1C1,2C1,3C1,4,C0,1C0,2C0,3C0,4 = '1101,1111,111 1,1111'. Set 4 combines four nibbles obtained from the four partitions of set 3 into 4-bit data as '1000'. Finally, set 5 acquires the 4-bit input pattern from set 4 and output bit 'AEB' from set 2 to give the final decision. Since A is greater than B, the proposed comparator structure provides the outputs AGB = '1', ALB = '0' and AEB = '0'.

V CIRCUIT DIAGRAM

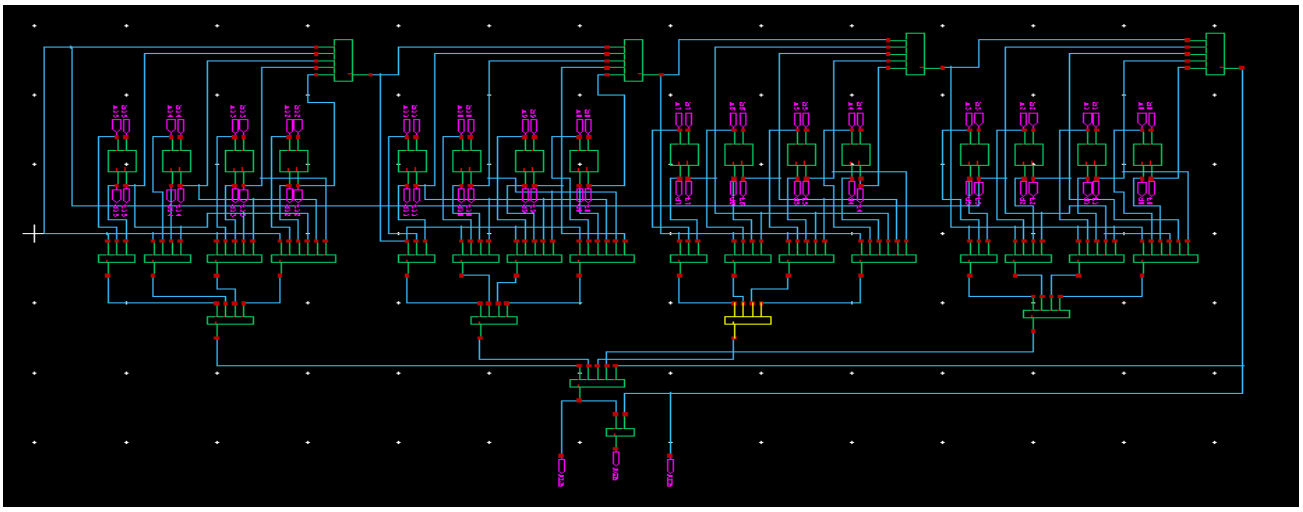


Fig 6: 16 bit proposed comparator circuit diagram

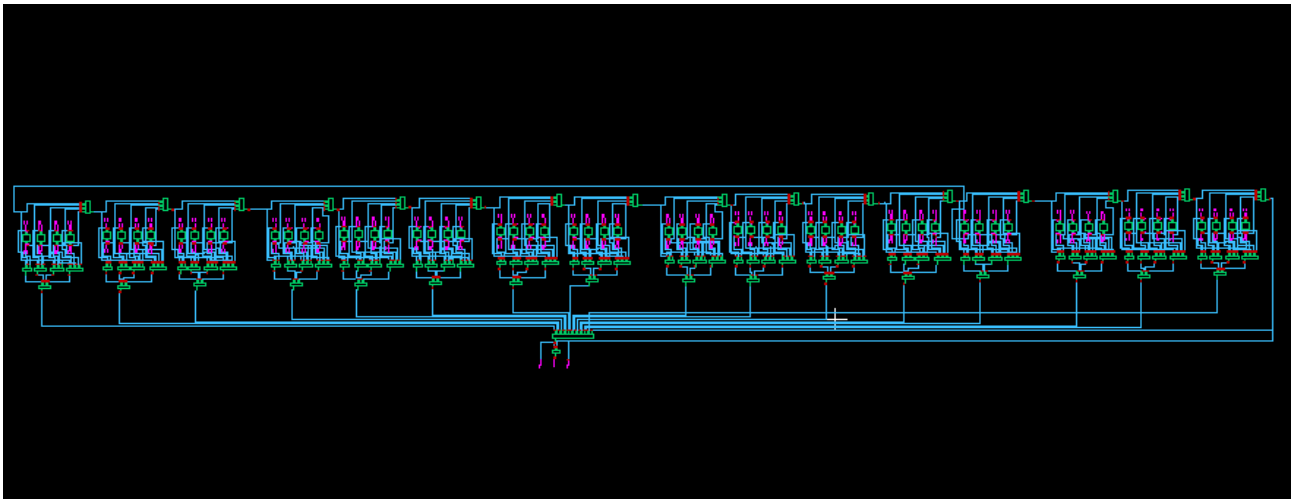


Fig 7:64 bit proposed comparator circuit diagram

VI RESULTS

a) Waveforms

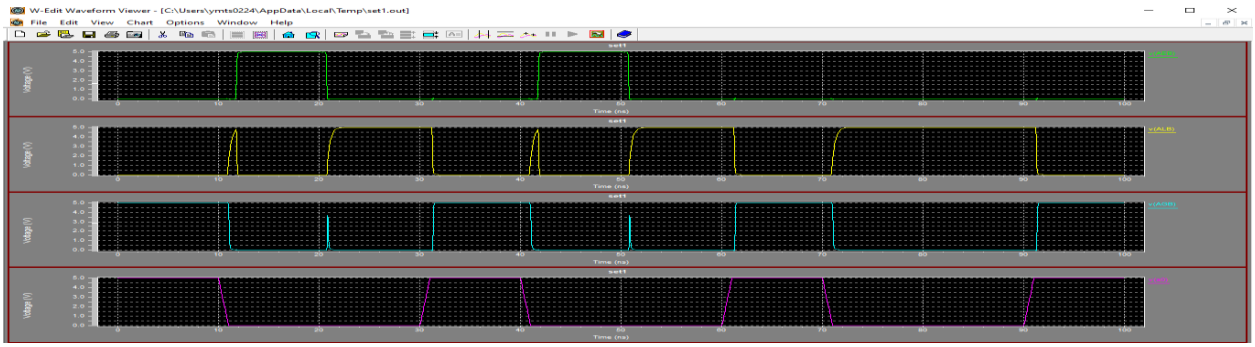


Fig 8: 16 bit proposed comparator wave forms

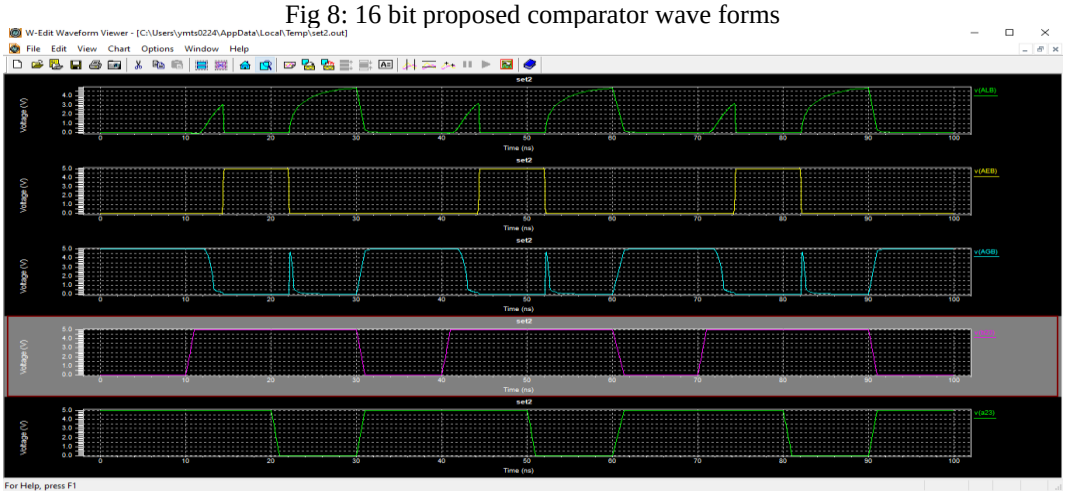


Fig 8: 64 bit proposed comparator circuit diagram

b) Delay report

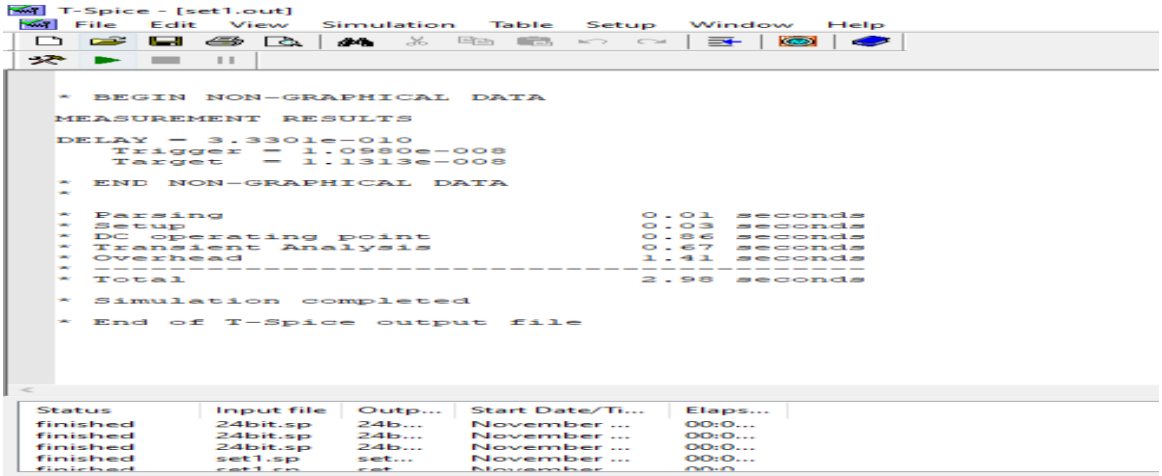


Fig 9: 16 bit comparator delay report

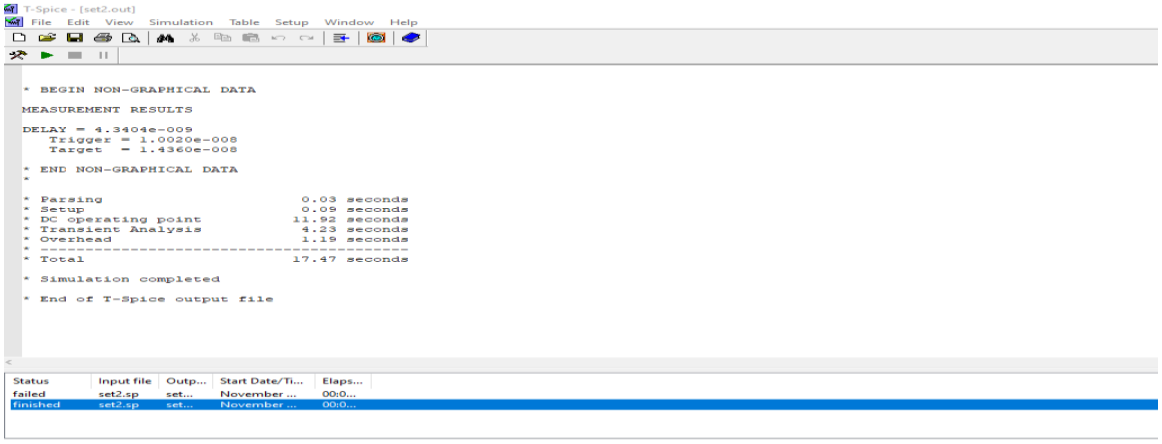


Fig 10: 64bit comparator delay report

c) Power report

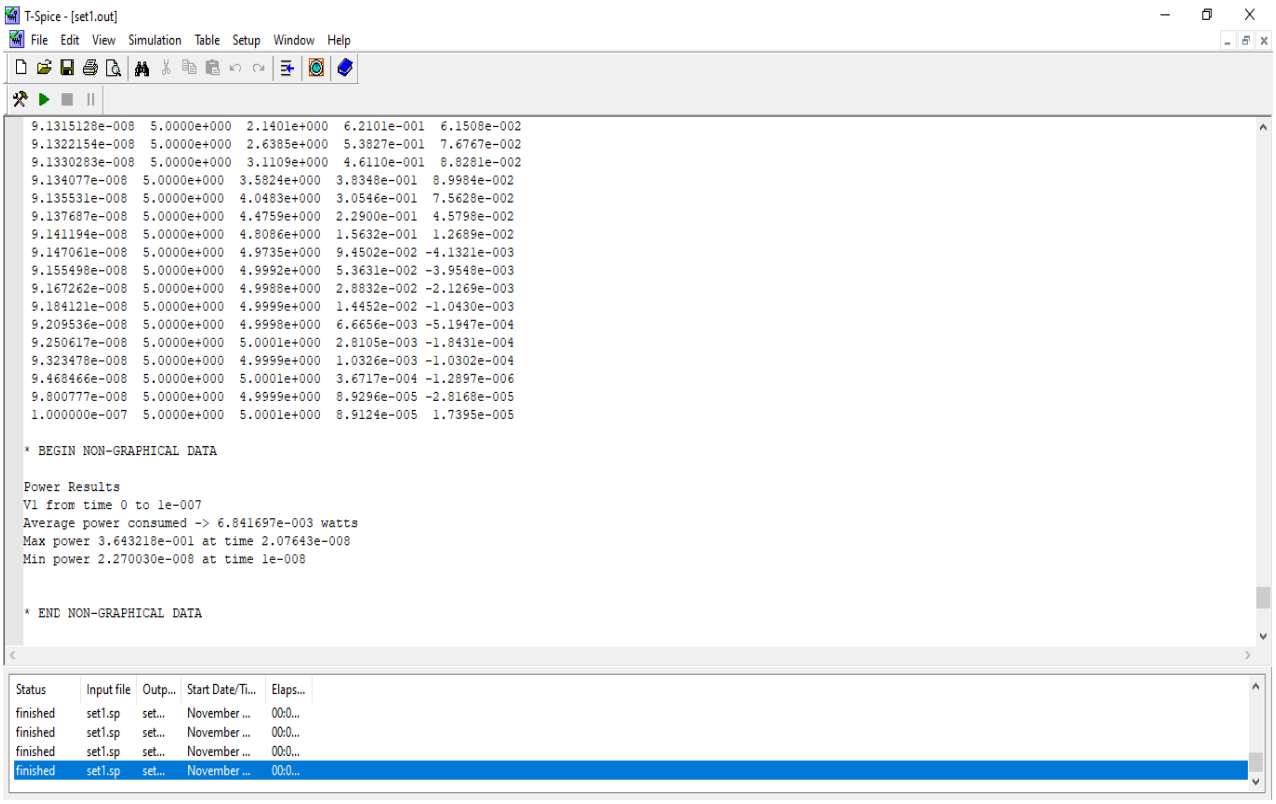


Fig 11: 16bit comparator power report

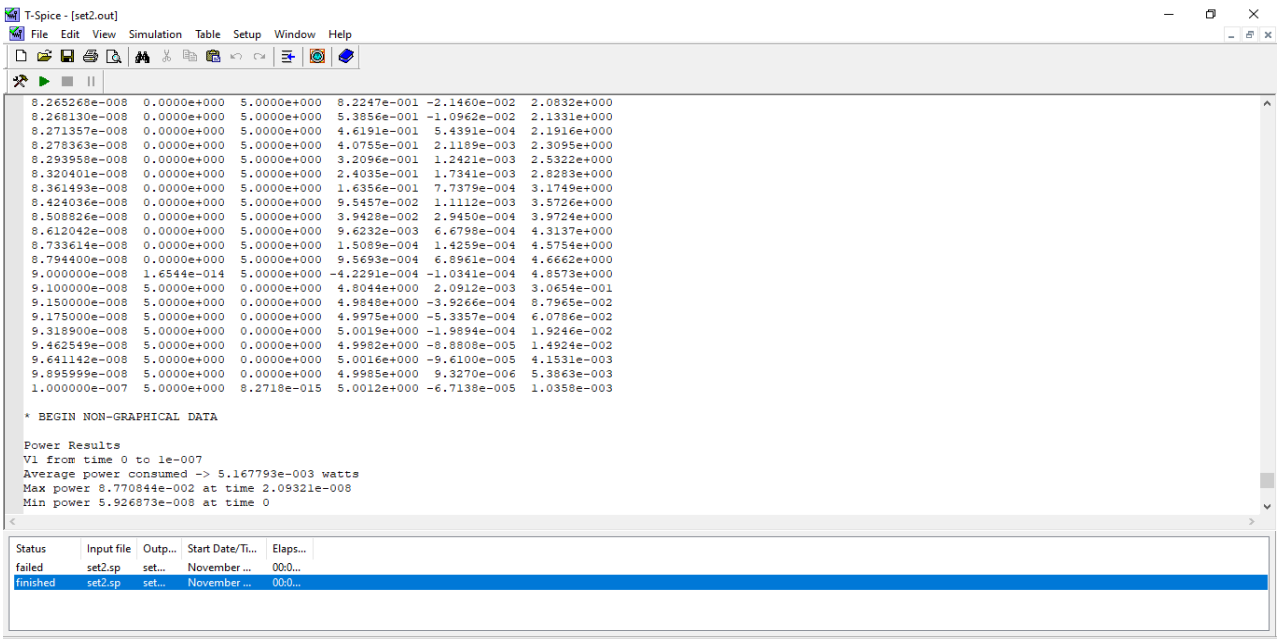


Fig 12: 64bit comparator power report

d) Area report

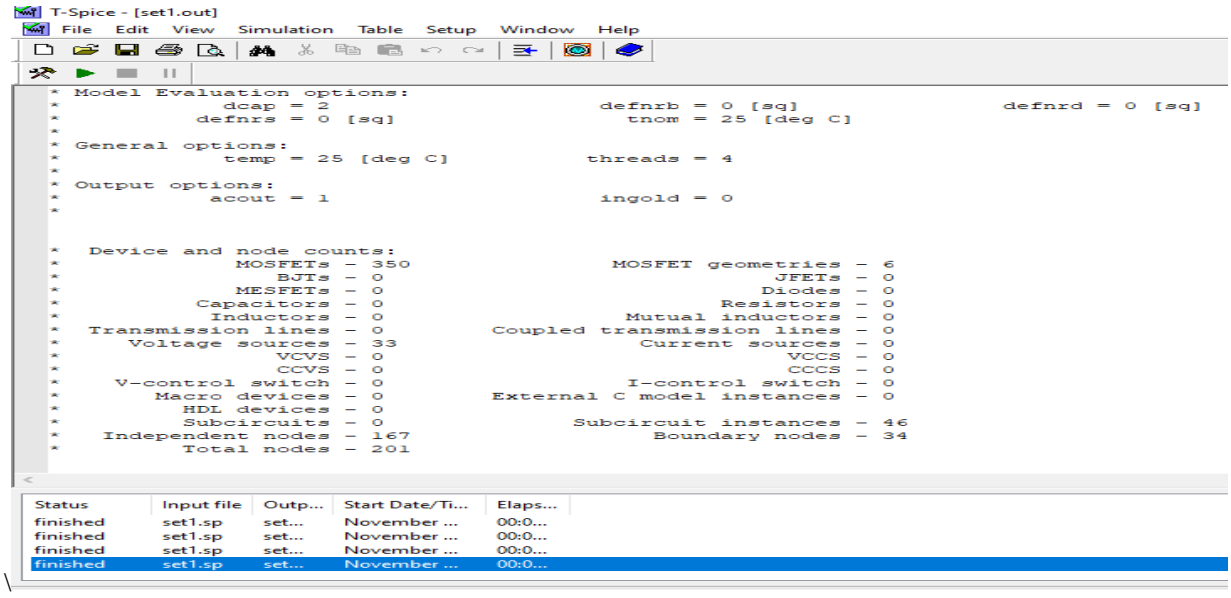


Fig 13: 16 bit comparator area report

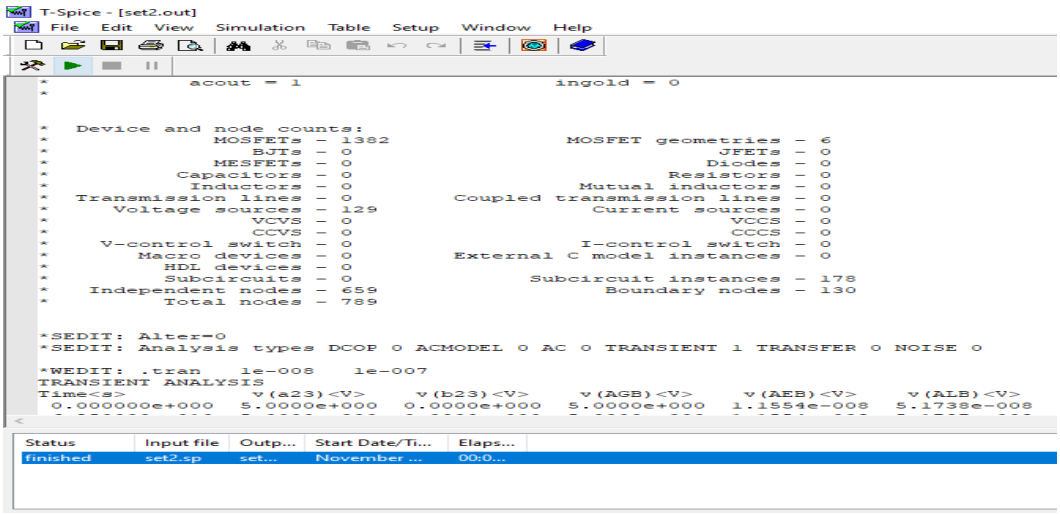


Fig 14: 64 bit comparator area report

VII CONCLUSION

In this paper, a novel scalable comparator using CEM and FM structures is proposed. The CEM comprises of the regular structure of repeated logic cells used for implementing parallel prefix tree structure. This regular structure can be used to predict the characteristics of the proposed comparator for arbitrary bit widths. The proposed comparator has a maximum operating frequency, low-power dissipation and minimum delay as compared with existing comparators designed using 0.18 µm CMOS technology. These advantages of the proposed comparator make it suitable for various applications such as scientific computations, test circuits, memory addressing logic etc.

VIII FUTURE WORK

The proposed high speed area efficient comparator has wide range of applications, which involve scientific computations (digital image processing, pattern recognition/ matching, arithmetic sorting, data compression and digital neural network and test circuit applications (built-in self-test circuits, signature analysers and jitter measurement). Its power consumption can be further improved by using adiabatic logic instead of CMOS logic.

REFERENCES

1. Parhami, B.: ‘Efficient hamming weight comparators for binary vectors based on accumulative and up/down parallel counters’, IEEE Trans. Circuits Syst., 2009, 56, (2), pp. 167– 171
2. Liu, H.J.R., Yao, H.: ‘High-performance VLSI signal processing innovative architectures and algorithms’ (IEEE Press, Piscataway, NJ, 1998)
3. Sheng, Y., Wang, W.: ‘Design and implementation of compression algorithm comparator for digital image processing on component’. Proc. Ninth Int. Conf. Young Computer Scientists, Hunan, China, November 2008, pp. 1337– 1341
4. Abramovici, M., Breuer, M.A., Friedman, A.D., et al.: ‘Digital systems testing and testable design’ (IEEE Press, Piscataway, NJ, 1990)
5. Chan, A., Roberts, G.W.: ‘A jitter characterization system using a component invariant Vernier delay line’, IEEE Trans. Very Large Scale Integer. (VLSI) Syst., 2004, 12, (1), pp. 79–95
6. Oklobdzija, V.G.: ‘An algorithmic and novel design of a leading zero detector circuit: comparison with logic synthesis’, IEEE Trans. Very Large Scale Integer. (VLSI) Syst., 1994, 2, (1), pp. 124–128
7. Suzuki, H., Kim, C.H., Roy, K., et al.: ‘Fast tag comparator using diode partitioned domino for 64-bit microprocessor’, IEEE Trans. Circuits Syst. I, 2007, 54, (2), pp. 322–328 [8] Ponomarev, D., Kucuk, G., Ergin, O., et al.: ‘Energy-efficient comparators for superscalar data paths’, IEEE Trans. Comput., 2004, 53, (7), pp. 892–904
8. Guangjie, W., Shimin, S., Lijiu, J., et al.: ‘New efficient design of digital comparator’. Proc. Second Int. Conf. Applications Specific Integrated Circuits, Shanghai, China, 1996, pp. 263–266
9. Abdel-Hafeez, S.: ‘Single rail domino logic for four-phase clocking scheme’, U.S. Patent 6265899, October

2001
10. Ercegovac, M.D., Lang, T.: 'Digital arithmetic' (Morgan Kaufmann, San Mateo, CA, 2004)
11. Stine, J.E., Schulte, M.J.: 'A combined two's complement and floating-point comparator'. Proc. Int. Symp. Circuits Systems, 2005, pp. 89–92
12. Cheng, S.: 'A high-speed magnitude comparator with small transistor count'. Proc. IEEE Int. Conf. Electronics, Circuits, Systems, Sharjah, United Arab Emirates, December 2003, pp. 1168–1171
13. Bellaour, A., Elmasry, M.I.: 'Low-power digital VLSI design circuits and systems' (Kluwer, Norwood, MA, 1995)
14. Wang, C.C., Wu, C.F., Tsai, K.C., et al.: '1 GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking', IEE Proc.-Comput. Digit. Tech., 1998, 145, (6), pp. 433–436
15. Belluomini, W., Jamsek, D., Nartin, A.K., et al.: 'Limited switch dynamic logic circuits for high-speed low-power circuit design', IBM J. Res. Dev., 2006, 50, (2–3), pp. 277–286
16. Wang, C., Lee, P., Wu, C., et al.: 'High fan-in dynamic CMOS comparators with low transistor count', IEEE Trans. Circuits Syst. I, 2003, 50, (9), pp. 1216–1220
17. Huang, C.H., Wang, J.S.: 'High-performance and power-efficient CMOS comparators', IEEE J. Solid-State Circuits, 2003, 38, (2), pp. 254–262
18. Lam, H.M., Tsui, C.Y.: 'A mux-based high-performance single-cycle CMOS comparator', IEEE Trans. Circuits Syst. II, 2007, 54, (7), pp. 591–595
19. Maheshwari, N., Sapatnekar, S.S.: 'Optimizing large multiphase level-clocked circuits', IEEE Trans. Comput. Aided Des. Integr. Circuits Syst., 1999, 18, (9), pp. 1249–1264
20. Frustaci, F., Perri, S., Lanuzza, M., et al.: 'Energy-efficient single-clock-cycle binary comparator', Int. J. Circuit Theory Appl., 2012, 40, (3), pp. 237–246
21. Coussy, P., Morawiec, A.: 'High-level synthesis: from algorithm to digital circuit' (Springer-Verlag, New York, 2008)
22. Perri, S., Corsonello, P.: 'Fast low-cost implementation of single-clock-cycle binary comparator', IEEE Trans. Circuits Syst. II, 2008, 55, (12), pp. 1239– 1243
23. Lutz, D.R., Jayasimha, D.N.: 'The half-adder form and early branch condition resolution'. Proc. 13th IEEE Symp. Computer Arithmetic, Asilomar, CA, USA, July 1997, pp. 266–273
24. Ercegovac, M.D., Lang, T.: 'Sign detection and comparison networks with a small number of transitions. Proc. 12th IEEE Symp. Computer Arithmetic, Bath, UK, July 1995, pp. 59–66
25. Bruguera, J.D., Lang, T.: 'Multilevel reverse most-significant carry computation', IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2001, 9, (6), pp. 959–
26. Hensley, J., Singh, M., Lastra, A., et al.: 'A fast, energy-efficient zcomparator'. Proc. ACM Conf. Graphics Hardware, Los Angeles, California, 2005, pp. 41–44
27. Ekanayake, V.N., Clinton, I.K., Manohar, R., et al.: 'Dynamic significance compression for a low-energy sensor network asynchronous processor'. Proc. 11th IEEE Int. Symp. Asynchronous Circuits Systems, New York City, NY, USA, March 2005, pp. 144–154
28. Lam, H.M., Tsui, C.Y.: 'High-performance single clock cycle CMOS comparator', Electron. Lett., 2006, 42, (2), pp. 75–77
29. Kim, J.Y., Yoo, H.J.: 'Bitwise competition logic for compact digital comparator'. Proc. IEEE Asian Solid-State Circuits Conf., Jeju, South Korea, November 2007, pp. 59–62
30. Hafeez, A., Ross, A., Parhami, B., et al.: 'Scalable digital CMOS comparator using a parallel prefix tree', IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2013, 21, (11), pp. 1989–1998
31. Boppana, N., Ren, S.: 'A low-power and area-efficient 64-bit digital comparator', J. Circuits Syst. Comput., 2016, 25, (2), pp. 1650148–1650163
32. Chua, C., Kumar, R., Sireesha, B., et al.: 'Design and analysis of low-power and area efficient N-bit parallel binary comparator', Analog Integr. Circuits Signal Process., 2017, 92, (2), pp. 225–231
33. Chua, C., Kumar, R.: 'An improved design and simulation of low-power and area-efficient parallel binary comparator', Microelectron. J., 2017, 66, pp. 84– 88
34. Abdel-Hafeez, S., Gordon-Ross, A., & Parhami, B. (2013). Scalable Digital CMOS Comparator Using a Parallel Prefix Tree. IEEE Transactions on Very Large Scale Integration (VLSI) Systems.
35. Lam, H.-M., & Tsui, C.-Y. (2007). A mux-based High-Performance Single-Cycle CMOS Comparator. IEEE Transactions on Circuits and Systems II
36. Vijaya Krishna Boppana, N. V., & Ren, S. (2016). A Low-Power and Area-Efficient 64-Bit Digital Comparator. Journal of Circuits, Systems and Computers, 25(12), 1650148.