INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER SCIENCE AND ENGINEERING TECHNOLOGIES

ISSN : 2454-9924

Design of Area efficient comparator architecture using 6T XOR Gate

V.Umadevi [1], C.Geetha[2].

[1]M.tech student, Dept of ECE (VLSI Design), YITS Tirupati.

Gmail: m.umaramesh@gmail.com

[2]Associate Professor, Dept of ECE, YITS Tirupati.

Gmail: geetharaj202@gmail.com

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: N-bit digital comparator, Low-power dissipation.

I.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. Numerous applications, like 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, etc.) signature analyzers and jitter measurement [4–5]) consist 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]. The comprehensive need for comparator logic in numerous computation-based designs necessarily involves efficiency in terms of area, power, and delay. Few comparator designs use dynamic logic to achieve power efficiency, however the constraints of higher speeds and poor noise margin (robustness) challenges modeling difficult. 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 scalability and comparison delay have been optimized in a hierarchical prefix tree structure-based comparator consisting of two bit comparators at each step [17]. However, for a wide range of input operands, these architectures may be prohibited due to the more delay and power usage caused by log2 N comparison levels. By using two input multiplexers at each level and generate–propagate logic at the first level, some of the limiting factors of the parallel prefix tree structure, such as area and power usage, can be improved. However, the comparator structure has very high power consumption since every cell remains in active state irrespective of the applied operand values.

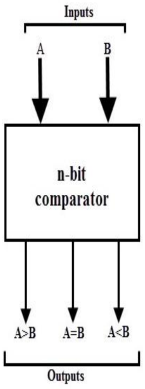
A digital comparator, also known as a magnitude comparator, is a hardware electronic instrument which decides whether one number is greater, less than, or equal to the other among the given two in-out numbers. The so called magnitude comparators are extensively found in Central Processing Units (CPUs), Micro Controller Units (MCUs), which are significant data path elements in image and signal processing applications.

The development of high-speed, low-power digital comparators has gained significant attention in recent years. In digital structures, data comparison is necessary when performing arithmetic or logical operations. This comparison decides whether a number is greater, equal, or less than another. A digital comparator, which is specifically designed to compare the relative magnitudes of binary numbers, is commonly used in combinational frameworks.

These are also available in Integrated Circuit (IC) form with various bit comparing combinations such as 4-bit, 8-bit, and so on. More than one comparator may also be cascaded to compare numbers of larger bit widths. When comparing two binary numbers, we must first compare the most significant bits(MSB) of the given two numbers Based on MSB Bit, it can be identified whether the number is smaller or greater . If these Most Significant bit (MSB)s are equal, the next relevant bits just need to be compared. However, if the MSBs are not identical, it is obvious that either A is greater than or less than B, and the comparing phase is terminated. Consider an example, say A = A1A0 and B = B1B0 are two 2-bit numbers. If A1(MSB bit of A) is not equal to B1(MSB bit of B), it is obvious that A is greater than B if A1 = 1 and B1 = 0, or that A is less than B if A0 = 0 and B0 =1. At this stage, the comparative process comes to an end. If the MSBs are equal, i.e., A1=B1 only then we need If the MSBs are equal, i.e., A1=B1, comparison of the next relevant bits A0 and B0 is required to determine if the number is greater than, less than, or equal to the other. As a result, the comparator generates three outputs: L, E, and G, which correspond to less than, equal to, and greater than comparisons.

A combinational circuit that compares two digital or binary numbers (say A and B) is known as magnitude comparator. It calculates their relative magnitudes to determine whether one number is equal to, less than, or greater than the other. To show the result of the comparison, three binary variables are used: A>B, A<B, or A=B. The diagram below depicts the block diagram of an n-bit comparator, which contrasts two numbers of n-bit length and generates the desired outputs according to the given inputs.

For comparing the address data, register, and perform all other arithmetic operations, the Magnitude comparators are primarily used in microcontrollers and CPU. Magnitude comparators are used in many applications, and any auto-turn-off system is almost certainly programmed with a comparator.



N-bit comparator

II. EXISTING METHOD

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 most significant bits (MSB) of the two operands are equal.

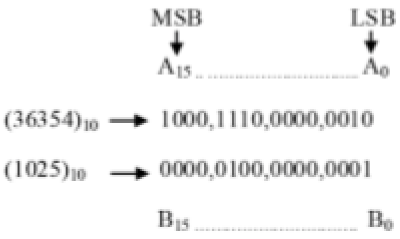


Fig. 1 Comparison between two 16 bit operands

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.

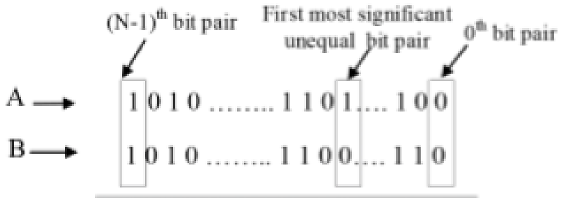


Fig. 2 Comparison between two N-bit operands

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

$$X = A \oplus B \quad (1)$$

$$E = A \odot B \quad (2)$$

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 eliminates unnecessary switching operations that occur during comparison process, thus limiting the proposed comparator's dynamic power consumption. The N-bit digital comparator is demonstrated in figure 3.

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 comparison evaluation module (CEM) incorporates parallel prefix tree structure that is included for executing a bitwise comparison of two N-bit operands A and B represented by $A_{N-1}A_{N-2}$, …, A_0 and B_{N-1} B_{N-2}, …, B_0.

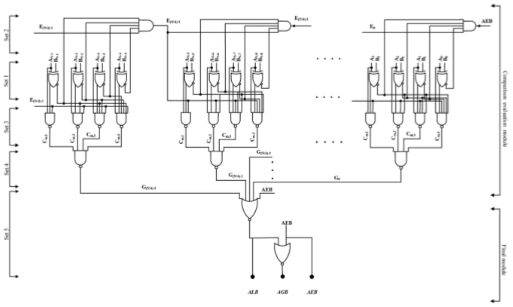


Fig.3 N-bit digital comparator

To investigate the regularity of the proposed magnitude comparator for specific bit widths, two operands A and B are applied into 4 bit partitions as $A_{N-1}A_{N-2}A_{N-3}A_{N-4}$,………$A_3A_2A_1A_0$ and $B_{N-1}B_{N-2}B_{N-3}B_{N-4}$,..………$B_3B_2B_1B_0$. The complete process of comparison is divided into five sets, in which CEM contains sets 1–4 and FM contains 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 structure of EX–OR–NOR cell shown in Fig. 4 is based on the pass transistor logic and CMOS logic.

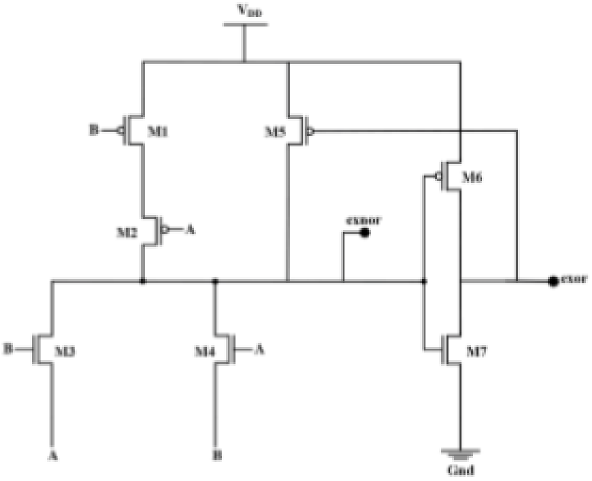


Fig. 4 circuit of EX-OR-NOR cell

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 Nchannel 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: $TK = AK \odot BK$ (3)

$$\text{set1:DK} = \text{AK} \oplus \text{BK} \ (4)$$

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 ≤ K ≤ 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

$$\text{set2: Em}-1 = \left(\prod_{m=0}^{N/4-1} T_{4m+3} + T_{4m+2} + T_{4m+1} + T_{4m}E_m\right) \qquad (5)$$

$$\text{set2: AEB(when m=0)}=\prod \mathrm{T}_3\mathrm{T}_2\mathrm{T}_1\mathrm{T}_0\mathrm{E}_0 \ (6)$$

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. For each cell in their respective partition, the number of inputs increases in ascending order from left to right, ending with the full 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

$$\text{Cm,1} = COMP\ (\prod_{m=0}^{N/4-1} E_m A_{4m+3} D_{4m+3})\ (7)$$

$$\text{Cm, 2} = COMP\ (\prod_{m=0}^{N/4-1} E_{\mathrm{m}}\mathrm{A}_{4\mathrm{m}+2}\mathrm{D}_{4\mathrm{m}+2}\mathrm{T}_{4\mathrm{m}+3}\ (8)$$

$$\text{Cm,3}=COMP\ (\prod_{m=0}^{N/4-1} E_{\mathrm{m}}\mathrm{A4}_{\mathrm{m}+1}\mathrm{D}_{4\mathrm{m}+1}\mathrm{T}_{4\mathrm{m}+3}\mathrm{T}_{4\mathrm{m}+2}\ (9)$$

$$\text{Cm,4}= COMP\ (\prod_{m=0}^{N/4-1} E_{\mathrm{m}}\mathrm{A}_{4\mathrm{m}}\mathrm{D}_{4\mathrm{m}}\mathrm{T}_{4\mathrm{m}+3}\mathrm{T}_{4\mathrm{m}+2}\mathrm{T}_{4\mathrm{m}+1}\ (10)$$

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 mth 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

$$\text{set4 :Gm} = COMP\ (\text{Cm,1 Cm, 2 Cm, 3 Cm, 4 })(11)$$

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

$$\text{set5:ALB=COMP}\ (\sum \mathrm{G}_{\mathrm{N}/4\,-1}\ldots\mathrm{G}_0(\text{AEB})\ (12)$$

$$\text{set5:AGB=COMP} \sum \text{ALB (AEB)}\ (13)$$

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. 3. 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 obtainedasC3,1C3,2C3,3C3,4,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'.

III. PROPOSED METHOD

Due to the extreme growth and demand for portable electronics, manufacturers are striving for shrinking silicon areas, higher performance, better battery life, and greater durability. When designing a device, one of the premium resources that a designer wants to conserve is power. XOR gates are fundamental units in wide variety of digital circuits that perform arithmetic computations. Some commonly used digital circuits the require XOR gate are Compressors, comparators, parity checkers. XOR gates are often found in the critical paths of complex arithmetic circuits used for multiplication and division. This, in particular, xor gate forms the basis of every digital framework and thus have an impact on the overall performance of the system.

In the present scenario, VLSI design demands hardware efficient and low power designs. In this paper, a low power hardware efficient comparator is presented using 6T XNOR. The XOR gate contains several inputs although only one output. It is the essential module of some combinational circuits, comparators. Also used in encryption and arithmetic circuits as a combination of XNOR and XOR circuit for low power consumption. The 6T XOR gate, which consists of 6 transistors aim at providing the utmost output voltage. It has less area with less power and parameters. It attain low power by reducing parasitic effect due to decrease in area. The main target of this proposed work is to be set to a low power area efficient comparator circuit using proposed 6T XOR gate with the least area by maintain a symmetry between power and speed at lesser voltage levels and highest output voltage levels.

A digital comparator is a piece of hardware that takes two numbers as input and decides whether given number is greater than, less than, or equal to the other. Comparators are integral part of CPU, error detection circuits and microcontrollers. In this paper, the XOR gate is replaced by 6T XOR thereby, we can reduce the power and hardware Utilization.

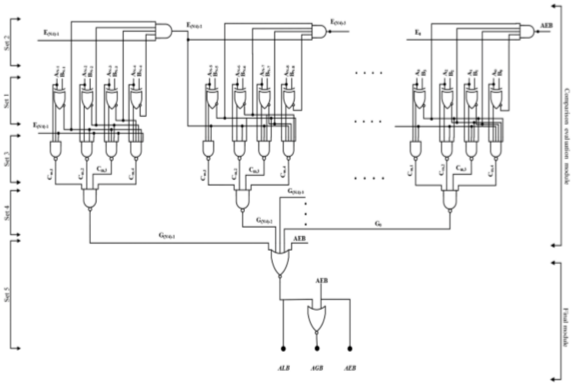


Fig.5 Block diagram of N- bit magnitude comparator

If we observe from the figure, we can find that for an n-bit comparator it requires n XOR gates. That means the size of comparator is directly proportional to the XOR gate count. Since the XOR gate is the major source of power consumption it is necessary to design a low power EXOR gate.

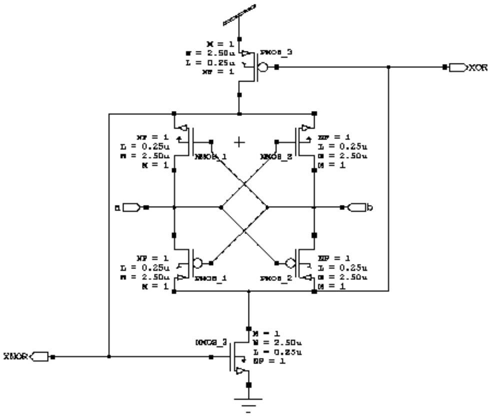


Fig.6 Circuit diagram of XOR gate.

The proposed XOR gate not only provides better power and area it also provides full swing voltage.it provides good driving capability. It can be cascaded to any number of stages because of its good driving capability.

IV. RESULTS AND DISCUSSION

The proposed design is hardware efficient, unlike the existing design, without degrading the other full swing output voltage. Simulation results show that the proposed comparator using 6T XOR occupies less area and power, improving the comparator architecture's efficiency.

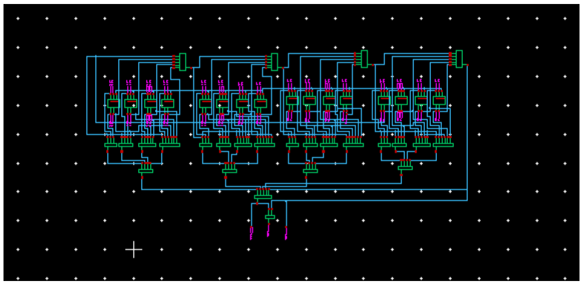


Fig.7 Schematic of Proposed 16 bit Magnitude Comparator

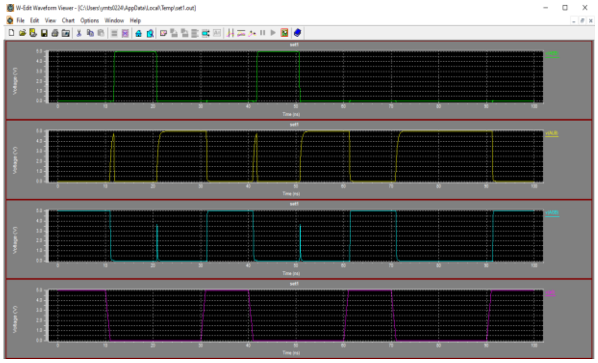


Fig.8 Waveform of Proposed 16 bit Magnitude comparator

Table: 1 Evaluation of Area, Delay and Power report

	Power(mw)	Delay(ns)	Area
Proposed 64 bit comparator	4.6	4.3	1318
Exist 64bit comparator	5.6	4.3	1382
proposed 24bit comparator	1.7	1.2	498
Exist 24bit comparator	1.9	1.2	522

Proposed 16bit comparator	4.9	0.6	334
Exist 16bit comparator	6.8	0.3	350

V. CONCLUSION

In this paper, a novel 6TXOR Gate is designed which is used in the proposed comparator. Comparator using CEM and FM structures is proposed. The (comparison evaluation module) CEM is made up of a regular structure of replicated logic cells that are used to enforce a parallel prefix tree structure. For specific bit widths, this regular structure may be used to predict the characteristics of the proposed magnitude comparator. All the designed are implemented using 0.18 µm CMOS Technology. The proposed comparator has a maximum operating frequency, lower power dissipation, and a smaller area. These advantages of the proposed comparator make it suitable for various applications such as scientific computations, test circuits, memory addressing logic etc.

REFERENCES

[1] Abdel-Hafez, “Single rail domino logic for four-phase clocking scheme”, U.S. Patent 6265899, October 2001

[2] Liu, H.J.R “High-performance VLSI signal processing innovative architectures and algorithms” (IEEE Press, Piscataway, NJ, 1998)

[3] Sheng, Y, “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, et al. “Digital systems testing and testable design” (IEEE Press, Piscataway, NJ, 1990)

[5] Chan, A, et al. “A jitter characterization system using a componentinvariant Vernier delay line”, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 2004, 12, (1), pp. 79–95

[6] Oklobdzija, ‘An algorithmic and novel design of a leading zero detector circuit: comparison with logic synthesis’, IEEE Trans. Very Large Scale Integr. (VLSI) Syst., 1994, 2, (1), pp. 124–128

[7] Suzuki 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 datapaths’, IEEE Trans. Comput., 2004, 53, (7), pp. 892–904

[9] 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

[13] Helms, “High-speed (HC/HCT) CMOS guide” (Prentice-Hall, Englewood Cliffs, NJ, 1989)

[14] Uyemura, J.P.: ‘CMOS logic circuit design’ (Kluwer, Norwood, MA, 1999)

[15] Parhami, ‘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

[16] Ercegovac, et al. ‘Digital arithmetic’ (Morgan Kaufmann, San Mateo, CA, 2004)

[17] Stine, J.E., Schulte, M.J, “A combined two's complement and floating-point comparator”. Proc. Int. Symp. Circuits Systems, 2005, pp. 89–92

[18] 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