

Design of High Speed Digital CMOS Comparator Using Parallel Prefix Tree

N. Prasanna¹, H. Sumitha²

¹Department Of ECE, Srinivasa Ramanujan Institute Of Technology : Ananthapuramu

²Assistantprofessor, Department Of ECE, Srinivasa Ramanujan Institute Of Technology : Ananthapuramu

Abstract: *This paper Presents a new comparator design is proposed by using parallel prefix tree. Energy efficient and high speed operation of comparators is needed for high speed digital circuits. The comparison outcome of the most significant bit, proceeding bitwise toward the least significant bit only when the compared bits are equal. In existing system, the parallel prefix structure is designed for 16, 32 and 64 bit architectures and the reports from the Xilinx tool concludes that for every bit range doubles the delay, memory, LUT and power has not doubled up to the mark. But In the proposed design of my project, each and every element in the parallel prefix structure will be replaced by universal logic (multiplexer). By performing this modification in the architecture will leads to reduction in POWER CONSUMPTION and DELAY.*

Keywords: Parallel prefix tree structure high fan in, high fan out, Bitwise competition logic(BCL).

1. Introduction

A high speed comparator is a very basic and useful arithmetic component of digital systems. Comparator is a major fundamental element in most digital circuits. The main advantages of this design are high speed and power efficiency, maintained over a wide range. Comparators are key design element for a wide range of applications like parallel testing, signature analyzer, built-in self-test circuits, graphics and image/signal processing[1]-[3]. The design of high-speed, low power, and area-efficient comparators. Comparison is a fundamental operation digital processors..

Comparator designs improve scalability and reduce comparison delays using a hierarchical prefix tree structure composed of 2-b comparators [17]. These structures require $\log_2 N$ comparison levels, with each level consisting of several cascaded logic gates. However, the delay and area of these designs may be prohibitive for comparing bit operands. The prefix tree structure's area and power consumption can be improved by leveraging two-input multiplexers (instead of 2-b comparator cells) at each level and generate-propagate logic cells on the first level (instead of 2-b adder cells)[18].

Furthermore, the structure can perform only "greater-than" or "less-than" comparisons and not equality. To improve the speed and reduce power consumption[20],[21] several designs rely on pipelining and power-down mechanisms to reduce switching activity with respect to the actual input operands' bit values. A 64-b comparator requires only three pipeline cycles using a multiphase clocking scheme[23]. However, such a clocking scheme may be unsuitable for high-speed single-cycle processors because of several heavily loaded global clock signals that have high-power transition activity. Additionally, race conditions and a heavily constrained clock jitter margin may make this design unsuitable for wide-range comparators. An alternative architecture leverages priority encoder magnitude decision logic with two pipelined operations that are triggered at both

the falling and rising clock edges[24] to improve operating speed and eliminate long dynamic logic chains. Other architectures use a multiplexer-based structure to split a 64-b comparator into two comparator stages[25]: the first stage consists of eight modules performing 8-b comparisons and the modules' outputs are input into a priority encoder and the second stage uses an 8-to-1 multiplexer to select the appropriate result from the eight modules in the first stage.

Similarly, other energy-efficient designs leverage schemes to reduce switching activity. Compute-on demand comparators compare two binary numbers one bit at a time, rippling from the most significant bit (MSB) to the least significant bit (LSB). The outcome of each bit comparison either enables the comparison of the next bit if the bits are equal, or represents the final comparison decision if the bits are different. Thus, a comparison cell is activated only if all bits of greater significance are equal. This scheme detects the larger operand by determining which operand possesses the leftmost 1 bit after pre-encoding before supplying the operands to bitwise competition logic (BCL) structure. The BCL structure partitions the operands into 8-b blocks and the result for each block is input into a multiplexer to determine the final comparison decision.

This structure consists of two basic modules: Comparison resolution module and decision module. The comparison resolution module divides two input N-bit arrays to be compared into two busses namely left bus and right bus each of N bits wide respectively. The decision module in turn decides whether equal, less than or greater than relationship exists between applied inputs for comparison. To overcome some of the drawback present in the above designs (such as higher power consumption, multi cycle computation, unsuitable custom structures for scaling, irregular VLSI structures, and irregular transistors), parallel prefix structure based comparator design provides fast, scalable, wide range, and power efficient algorithm.

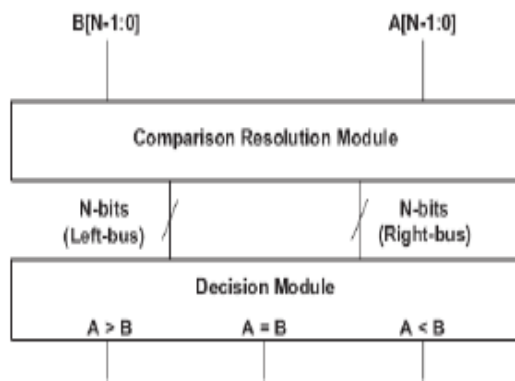


Figure 1: Block diagram of our comparator architecture, consisting of a comparison resolution module connected to a decision module.

The comparison resolution module is a novel MSB-to-LSB parallel prefix tree structure that performs the bitwise comparison of two N bit operands (A & B) entered into the comparator. The parallel structure encodes the bitwise comparison results to two N bit buses called left bus and right bus. The bitwise comparison of equal bits sets $_0$ in both the buses. If the bitwise comparison of unequal bits occur, any of the buses (A or B) sets to $_1$ and the bitwise comparison stops immediately by setting $_0$ in the remaining bits present in the buses. The decision module produces the result of comparison of the input operands based on the signals from the left and right buses. The possible results from the decision module are $A=B, A>B, A<B$.

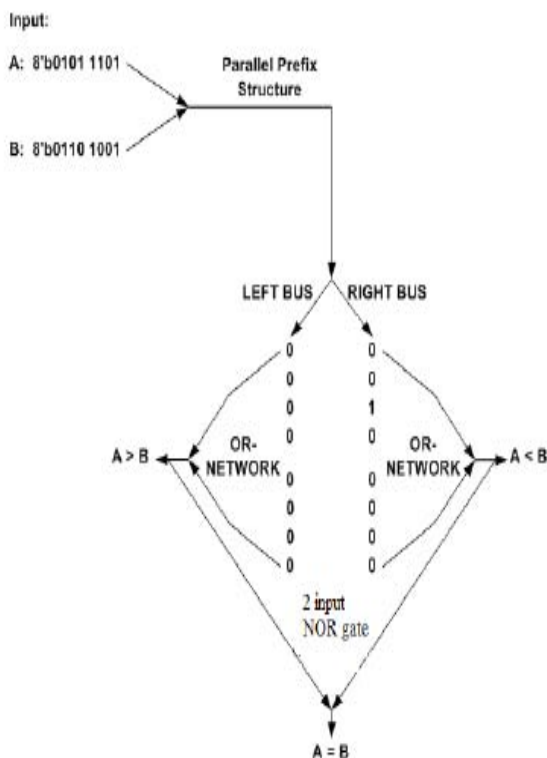


Figure 2: Example 8-bit Comparison

Let the two 8-bit binary numbers be A and B. $A = 0101\ 1101$ and $B = 0110\ 1001$. In the first step, a parallel prefix tree structure generates the encoded data on the left bus and right bus for each pair of corresponding bits from A and B. In this example, $A_7 = 0$ and $B_7 = 0$ encodes as $left_7 = right_7$

$= 0, A_6 = 1, \text{ and } B_6 = 1$ encodes as $left_6 = right_6 = 0$, and $A_5 = 0$ and $B_5 = 1$ encodes $left_5 = 0$ and $right_5 = 1$. At this point, since the bits are unequal, the comparison terminates and a final comparison decision can be made based on the first three bits evaluated. The parallel prefix structure forces all bits of lesser significance on each bus to 0, regardless of the remaining bit values in the operands. In the second step, the OR-networks perform the bus OR scans, resulting in 0 and 1, respectively.

2. Comparator Design Details

We partition the structure into five hierarchical prefixing sets, where as each set performs a exact function whose output serves as input to the next set, in hope of the fifth set produces the output on the left bus and the right bus Every part of cells components within each set operate in parallel were as it's a key feature to increase operating speed while minimizing the transitions to a minimal set of left most bits needed for a correct decision. This prefixing set structure bounds the components fan-in and fan-out regardless of comparator bit-width and eliminates heavily loaded global signals with parasitic components, thus improving the operating speed and reducing power consumption.

3. Architecture of 16-Bit Comparator Using Parallel Prefix Tree

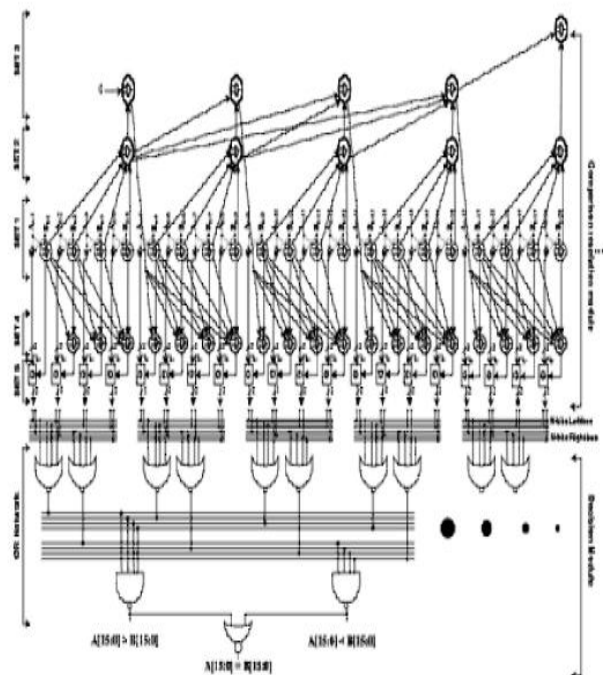


Figure 3: Implementation details for the comparison resolution module (sets 1 through 5) and the decision module

In this project the switching logic and the main block design is carried out by using mux logic to perform low power operations because , In electronics, a multiplexer is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. A multiplexer of 2n inputs has n select lines, which are used to select which input line to send to the output. A multiplexer is also

called a data selector. An electronic multiplexer can be considered as a multiple-input single output.

4. Advantages by Using Multiplexr Based in Parallel Prefix Tree

Low power consumption by replacing the needed logics by multiplexer, because multiplexer operates at very low power switching transitions compared to another logical gates. Low delay compared to normal based comparator, less area and less LUT compared to existing system.

5. Conclusion

In this project a Design Of High Speed CMOS Comparator Using Parallel Prefix Tree using regular digital hardware structures consisting of two modules: the comparison resolution module and the decision module. This regularity allows simple prediction of comparator characteristics for arbitrary bit widths and is attractive for continued technology Scaling and logic synthesis. These modules are structured as parallel prefix trees by using a normal flow.

Family	Existed one using logic gates 16 bit	Proposed one using mux 16bit
Spartan=3 (XC3S50) (PQ208) Speed=4	LUT=57 Time Delay=18.774ns	LUT=47 Time Delay=16.739ns

Future work will include additional circuit optimizations to further reduce the power dissipation by adapting dynamic and analog implementations for the comparator resolution module and a high-speed zero-detector circuit for the decision module. Given that our comparator is composed of two balanced timing modules, the structure can be divided into two or more pipeline stages with balanced delays, based on a set structure, to effectively increase the comparison throughput.

References

- [1] H. J. R. Liu and H. Yao, *High Performance VLSI Signal Processing Innovative Architectures and Algorithms*, vol. 2. Piscataway, NJ: IEEE Press, 1998.
- [2] Y. Sheng and W. Wang, "Design and implementation of compression algorithm comparator for digital image processing on component," in *Proc. 9th Int. Conf. Young Comput. Sci.*, Nov. 2008, pp. 1337–1341.
- [3] B. Parhami, "Efficient hamming weight comparators for binary vectors based on accumulative and up/down parallel counters," *IEEE TransCircuits Syst.*, vol. 56, no. 2, pp. 167–171, Feb. 2009.
- [4] A. H. Chan and G. W. Roberts, "A jitter characterization system using a component-invariant Vernier delay line," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 12, no. 1, pp. 79–95, Jan. 2004.
- [5] M. Abramovici, M. A. Breuer, and A. D. Friedman, *Digital Systems Testing and Testable Design*, Piscataway, NJ: IEEE Press, 1990.
- [6] H. Suzuki, C. H. Kim, and K. Roy, "Fast tag comparator using diode partitioned domino for 64-bit

- microprocessor," *IEEE Trans. Circuits Syst. I*, vol. 54, no. 2, pp. 322–328, Feb. 2007.
- [7] D. V. Ponomarev, G. Kucuk, O. Ergin, and K. Ghose, "Energy efficient comparators for superscalar datapaths," *IEEE Trans. Comput.*, vol. 53, no. 7, pp. 892–904, Jul. 2004.
- [8] V. G. Oklobdzija, "A algorithmic and novel design of a leading zero detector circuit: Comparison with logic synthesis," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 1, pp. 124–128, Mar. 1994.
- [9] H. L. Helms, *High Speed (HC/HCT) CMOS Guide*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [10] SN7485 *4-bit Magnitude Comparators*, Texas Instruments, Dallas, TX, 1999.
- [11] K. W. Glass, "Digital comparator circuit," U.S. Patent 5 260 680, Feb. 13, 1992.
- [12] D. norris, "Comparator circuit," U.S. Patent 5 534 844, Apr. 3, 1995.
- [13] W. Guangjie, S. Shimin, and J. Lijiu, "New efficient design of digital comparator," in *Proc. 2nd Int. Conf. Appl. Specific Integr. Circuits*, 1996, pp. 263–266.
- [14] S. Abdel-Hafeez, "Single rail domino logic for four-phase clocking scheme," U.S. Patent 6 265 899, Oct. 20, 2001.
- [15] M. D. Ercegovic and T. Lang, *Digital Arithmetic*, San Mateo, CA: Morgan Kaufmann, 2004.
- [16] J. P. Uyemura, *CMOS Logic Circuit Design*, Norwood, MA: Kluwer, 1999.
- [17] J. E. Stine and M. J. Schulte, "A combined two's complement and floating-point comparator," in *Proc. Int. Symp. Circuits Syst.*, vol. 1. 2005, pp. 89–92.
- [18] S.-W. Cheng, "A high-speed magnitude comparator with small transistor count," in *Proc. IEEE Int. Conf. Electron., Circuits, Syst.*, vol. 3. Dec. 2003, pp. 1168–1171.
- [19] A. Bellaour and M. I. Elmasry, *Low-Power Digital VLSI Design Circuits and Systems*. Norwood, MA: Kluwer, 1995.
- [20] W. Belluomini, D. Jamsek, A. K. Nartin, C. McDowell, R. K. Montoye, H. C. Ngo, and J. Sawada, "Limited switch dynamic logic circuits for high-speed low-power circuit design," *IBM J. Res. Develop.*, vol. 50, nos. 2–3, pp. 277–286, Mar.–May 2006.
- [21] C.-C. Wang, C.-F. Wu, and K.-C. Tsai, "1GHz 64-bit high-speed comparator using ANT dynamic logic with two-phase clocking," in *IEE Proc.-Comput. Digit. Tech.*, vol. 145, no. 6, pp. 433–436, Nov. 1998.
- [22] C.-C. Wang, P.-M. Lee, C.-F. Wu, and H.-L. Wu, "High fan-in dynamic CMOS comparators with low transistor count," *IEEE Trans. Circuits Syst. I*, vol. 50, no. 9, pp. 1216–1220, Sep. 2003.
- [23] N. Maheshwari and S. S. Sapatnekar, "Optimizing large multiphase level-clocked circuits," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 18, no. 9, pp. 1249–1264, Nov. 1999.
- [24] C.-H. Huang and J.-S. Wang, "High-performance and power-efficient CMOS comparators," *IEEE J. Solid-State Circuits*, vol. 38, no. 2, pp. 254–262, Feb. 2003.
- [25] H.-M. Lam and C.-Y. Tsui, "A mux-based high-performance single-cycle CMOS comparator," *IEEE Trans. Circuits Syst. II*, vol. 54, no. 7, pp. 591–595, Jul. 2007.

- [26] F. Frustaci, S. Perri, M. Lanuzza, and P. Corsonello, "Energy-efficient single-clock-cycle binary comparator," *Int. J. Circuit Theory Appl.*, vol. 40, no. 3, pp. 237–246, Mar. 2012.
- [27] P. Coussy and A. Morawiec, *High-Level Synthesis: From Algorithm to Digital Circuit*. New York: Springer-Verlag, 2008.
- [28] S. Perri and P. Corsonello, "Fast low-cost implementation of single clock-cycle binary comparator," *IEEE Trans. Circuits Syst. II*, vol. 55, no. 12, pp. 1239–1243, Dec. 2008.
- [29] M. D. Ercegovac and T. Lang, "Sign detection and comparison networks with a small number of transitions," in *Proc. 12th IEEE Symp. Comput. Arithmetic*, Jul. 1995, pp. 59–66.
- [30] J. D. Bruguera and T. Lang, "Multilevel reverse most-significant carry computation," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 9, no. 6, pp. 959–962, Dec. 2001.
- [31] D. R. Lutz and D. N. Jayasimha, "The half-adder form and early branch condition resolution," in *Proc. 13th IEEE Symp. Comput. Arithmetic*, Jul. 1997, pp. 266–273.
- [32] J. Hensley, M. Singh, and A. Lastra, "A fast, energy-efficient z-comparator," in *Proc. ACM Conf. Graph. Hardw.*, 2005, pp. 41–44.
- [33] V. N. Ekanayake, I. K. Clinton, and R. Manohar, "Dynamic significance compression for a low-energy sensor network asynchronous processor," in *Proc. 11th IEEE Int. Symp. Asynchronous Circuits Syst.*, Mar. 2005, pp. 144–154.
- [34] H.-M. Lam and C.-Y. Tsui, "High-performance single clock cycle CMOS comparator," *Electron. Lett.*, vol. 42, no. 2, pp. 75–77, Jan. 2006.
- [35] J.-Y. Kim and H.-J. Yoo, "Bitwise competition logic for compact digital comparator," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2007, pp. 59–62.
- [36] M. S. Schmookler and K. J. Nowka, "Leading zero anticipation and detection—a comparison of methods," in *Proc. 15th IEEE Symp. Comput. Arithmetic*, Sep. 2001, pp. 7–12.