IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

N-BIT CMOS Comparator with Zero Crossing Detector Using Parallel Prefix Tree

V. Sidharthan [*1], Dr. K. Gopalakrishnan [2]

[*1] Asst. Professor, Dept of Electronics, S.N.R. Sons College, Coimbatore-641006, India

[2] Asso. Professor, Dept of Electronics, S.N.R. Sons College, Coimbatore-641006, India

n.v.sidharth@gmail.com

Abstract

This paper provides an experience of new comparator model gives large range, with faster operation by converting n-bit CMOS cells. This comparator make use of novel scalable parallel prefix constructs strategic advantage by comparing Most Significant Bit (MSB) outcomes which is scheduled bit wise towards the Least Significant Bit (LSB). By comparing as the bits are equal and high speed zero detector circuit is used for decision module to reduce dynamic power wastage by eliminating unnecessary conversions in parallel prefix that render N-bit compression result following [log 4 N] + [log 16 N] + 4 CMOS cells. Core lead of this model is high speed and power effectiveness is maintained over a wide range. More than this, the design uses a standard reconfigurable VLSI topology that permits logical derivation of the input-output delay as a role of bandwidth. HSPICE form used in 32 bit comparator shows a defective case input output delay of 0.86ns and at most power consumption of 7.7mW using 0.15-µm TSMC technology at 1GHz.

Keywords: High-speed Arithmetic unit, Wide bit Comparator Architecture, Parallel prefixes tree structure, Zero Crossing detector.

Introduction

The Comparators are key design elements for a wide range of applications scientific computation, test circuit applications, and optimized equality only comparators for general purpose processor components. Even though comparator logic design is straightforward, the wide use of comparators in high-performance systems places a great importance on performance and power consumption optimizations. Some state, the comparator designs use dynamic gate logic circuit structures to enhance performance, while others leverage specialized arithmetic units for wide comparisons, along with custom logic circuits.

The prefix tree structure area and power consumption can be improved by leveraging two input multiplexers at each level and generate propagate logic cells on the first level, that take advantages of one's complement addition. Using this logic composition, a prefix tree requires six levels for the most common comparison bit-width of 32 bits, but suffers from high power consumption due to every cell in the structure being active, regardless of the input operands values. Furthermore, the structure can perform only "greater-than" or "less-than" comparisons and not equality

To improve the speed and reduce power utilization, numerous designs rely on pipelining and power down mechanisms to reduce switching activity, with respect to the actual input operands' bit values. One design uses all N transistor circuits to compensate for high fan-in with high pipeline throughput. A 32 bit comparator requires only three pipeline cycles using a multiphase clocking scheme.

An alternative architecture leverages priority encoder magnitude decision logic with two pipelined operations that are triggered at both the falling and rising clock edges to improve operating speed and eliminate long dynamic logic chains. This structure leads to a large overall conductive resistance, with heavily loaded parasitic components on the clock signal, that strictly limits the clock speed. Other architectures use a multiplexer-based structure to split a 32 bit comparator into two comparator stages. The first stage consists of eight modules performing 8 bit 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. This architecture uses two phases domino clocking to perform both stages in a single clock series. From the time once operation occurs on the increasing and falling clock edges, this additionally confines the operating speed and jitter margin and makes the design highly susceptible to race conditions. Some comparators combine a tree structure with a two phase domino clocking structure for speed enrichment. These architectures add two inputs, after negating one input via two's complement, using the carry-out signal as the "greater-than" or "less-than" indicator

The comparators compare two binary numbers one bit at a time, rippling from the MSB to the 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 unlike. As a result, a comparison cell is activated only if all bits of greater significance are equal. To reduce the long delays suffered by bitwise ripple designs, an enhanced architecture incorporates an algorithm that uses no arithmetic operations. 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 bit blocks and the result for each block is input into a multiplexer to determine the final comparison ruling. Appropriate to the BCL based designs low transistor count, this design has the potential for low power utilization, but the pre-encoder logic modules prior to the BCL modules limit the maximum achievable operating frequency.

In addition, special control logic is needed to enable the BCL units to switch dynamically in a synchronized fashion, thus raising the power consumption and dropping the operating frequency. Use of reconfigurable arithmetic algorithms, with total (input-to-output) hardware realization for both fully- custom and standard-cell approaches, improves the longevity of our design and makes our design ideal for technology scaling and short time to market. A novel MSB to LSB parallel prefix tree structure, based on a reduced switching paradigm and using parallelism at each level, contributes to the speed and energy efficiency of our design.

Use of complement logic, with neither clock gating nor latency delay, enables global partitioning into two major pipelined stages or locally into several pipelined stages based on the number of levels. This flexibility provides area versus performance tradeoffs.

Comparator architectural overview

The comparison resolution module in Fig. 1 is a novel MSB to LSB parallel prefix tree structure that performs bit wise comparison of two N -bit operands A and B, denoted as A_{N-1}, A_{N-2}, . . ., A_0 and B_{N-1}, B_{N-2}, . . ., B_0, where the subscripts range from $N-1$ for the MSB to 0 for the LSB. The comparison resolution modules perform the bitwise comparison asynchronously from left to right, were as the comparison logic's computation is triggered only if all bits of greater significance are equal. The parallel structure encodes the bit wise comparison results into two N-bit buses, the left bus and the right bus, each of those stores partial comparison result as each bit position is evaluated.

An 8-b comparison of input operands A = 01011101 & B = 01101001 is illustrated in Fig. 2. 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.

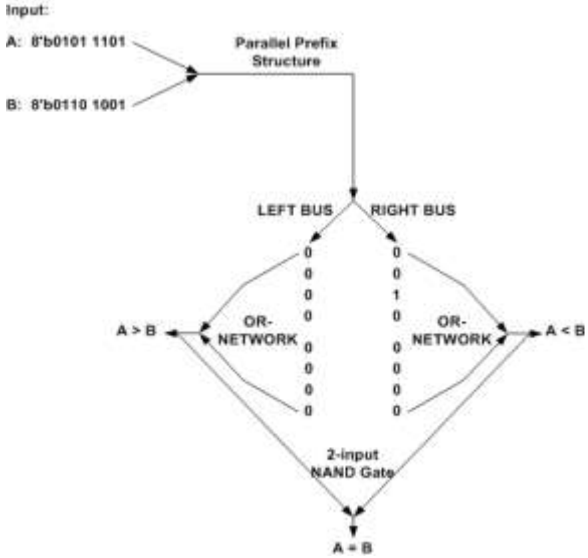


Fig. 2: Example 8-bit comparison

In this example, $A_7 = 0$ and $B_7 = 0$ encodes as $left_7 = right_7 = 0$, $A_6 = 1$, 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 arrangement forces all bits of lesser significance on each bus to 0, apart from the remaining bit values in the operands. In the second step, the OR-networks perform the bus OR-scans, resulting in 0 and 1, correspondingly, and the final comparison decision.

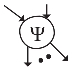
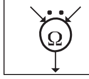
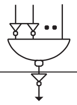
We partition the structure into five hierarchical prefixing sets, as depicted in Fig. 3,

Table I: Symbols Notation and Definitions

Symbol (Cells)	Definition

N	Operand bitwidth
A	First input operand
B	Second input operand
R	Right bus result bit
L	Left bus result bit
&	Bitwise AND
|	Bitwise OR
T{*}	Logic function of cell type*
COMP{*}	Complement function of set*

Table II: Logic Gate representations

Symbols (Cells)	**Logic Gate**	**Maximum Fan-in/Fan-out**
Ak Bk  	Ak Bk 	2 / 4 (12) 4 / 4 (8)

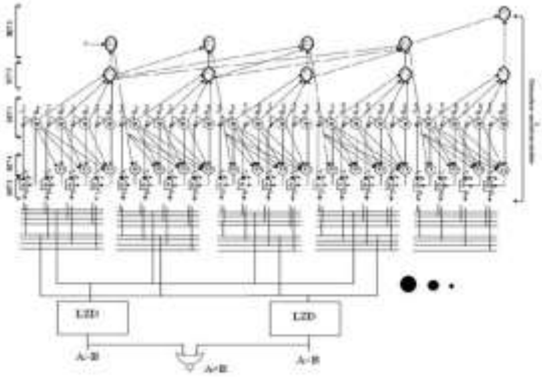
with the associated symbol representations in Tables I and II, where as each set performs a exact function whose output serves as input to the next set, in hope of the fifth set produces the output on the left bus and the right bus.

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.

Comparator design details

In this section, we detail our comparator's design Figure 3, which is based on using a novel parallel prefix tree Tables I and II contain symbols and definitions. Each set or groups of cells that produces output and serve as inputs to the next set in the hierarchy, with the exception of set 1, the outputs serve as inputs to several sets. Set 1 compares the N -bit operands A and B bit-by-bit, using a single level of N-Type cell. Those cells compute (where $0 \le k \le N - 1$). Set 2 consists of 2 N-type cells, which combine the termination flags for each of the four N - type cells from set. Set 3 provides functionality similar to set 2 using the same NOR- logic to continue or terminate the bitwise comparison activity. If the comparison is terminated, then the set 3 signals set 4 to the set left bus and right bus bits to 0 for all bits of lower significance. Set 4 consists of Q-type cells, whose outputs control the select inputs of Q–type cells in set 5, which in turn drive both the left bus and the right bus. For an Q-type cell and the 4 bit partition to which the cell belongs, bitwise comparison outcomes from set 1 provide information about the MSB in the cell's Q-type cells, which



Compute $(0 \le k \le N - 1)$ The number of inputs in the Q-type cells increases from left to right in each partition. Thus, the Q-type cells in set 4 determine whether set 5 propagates the bitwise comparison codes. The superscripts "1" and "0" in (8) and (9) denote the summation of the left and right bits, respectively, and the subscript "1" denotes the first level of OR-logic in the decision module that receives data directly from set 5.

-Type Cell	Input Driving -Type Cell Output
Y_1	D_{15}
Y_1	$D_{15} D_{14}$

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

Y_1	$D_{15}\, D_{14}\, D_{13}$
Y_1	$D_{15}\, D_{14}\, D_{13}\, D_{12}$
Y_1	$C_{3,0}\, D_{11}$
Y_1	$C_{3,0}\, D_{11}\, D_{10}$
Y_9	$C_{3,0}\, D_{11}\, D_{10}\, D_9$
Y_8	$C_{3,0}\, D_{11}\, D_{10}\, D_9\, D_8$
Y_7	$C_{3,1}\, D_7$
Y_6	$C_{3,1}\, D_7\, D_6$
Y_5	$C_{3,1}\, D_7\, D_6\, D_5$
Y_4	$C_{3,1}\, D_7\, D_6\, D_5\, D_4$
Y_3	$C_{3,2}\, D_3$
Y_2	$C_{3,2}\, D_3\, D_2$
Y_1	$C_{3,2}\, D_3\, D_2\, D_1$
Y_0	$C_{3,2}\, D_3\, D_2\, D_1\, D_0$

Area, speed and power evaluations

The area, operating speed, and power requirements of proposed comparator architecture and calculate the number of logic levels required for an N -bit comparator based on simple CMOS logic gates. The deriving the total number of cells required and use Table IV to translate the cell counts into transistors for an N -bit comparator. Table IV shows the total number of cells and the required number of levels per set for various comparator bit-widths.The critical path delay of our proposed comparators with N bit Inputs, the total delay of 16 puts our design among the fastest comparators reported in based on a basic CMOS gate circuit without any circuit level modifications. Minimizing the switching activity reduces the average power dissipation and is considered a key enabling technique for modern low-power design. The operands activate all cells in set 1 in parallel, thus set 1 provides no power savings. Table V shows that set 1 account for 25% of the total transistors, and thus power dissipation, for an arbitrary comparator size.

Table VI: Leakage Power for CMOS NAND with 4 Transistors at Different node factors

	0.18 μm **1.95 V**	**0.15 μm** **1.65 V**	**0.13 μm** **1.5 V**	**0.09 μm** **1 V**
NAND CMOS 4 Transistors	11.58 nW	33.3 nW	657.3 nW	984.2 nW

Table VII: Leakage Power for Comparator with 32 bit at different node factors

	0.18 μm **1.95 V**	**0.15 μm** **1.65 V**	**0.13 μm** **1.5 V**	**0.09 μm** **1 V**
32 bit comparator 2000 transistors	0.0116 mW	0.0534 mW	0.626 mW	0.8619 mW

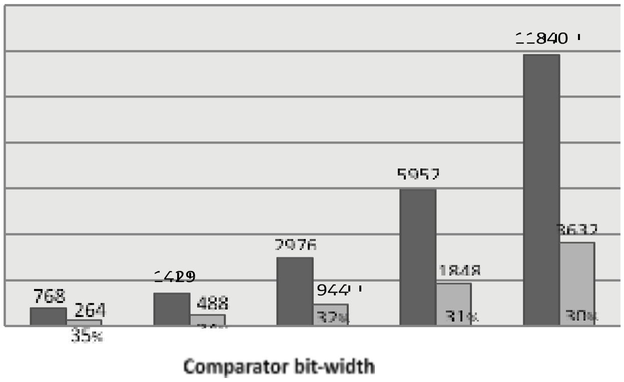


Fig.4. Total number of transistors and number of active transistors

ISSN: 2277-9655
Scientific Journal Impact Factor: 3.449
(ISRA), Impact Factor: 1.852

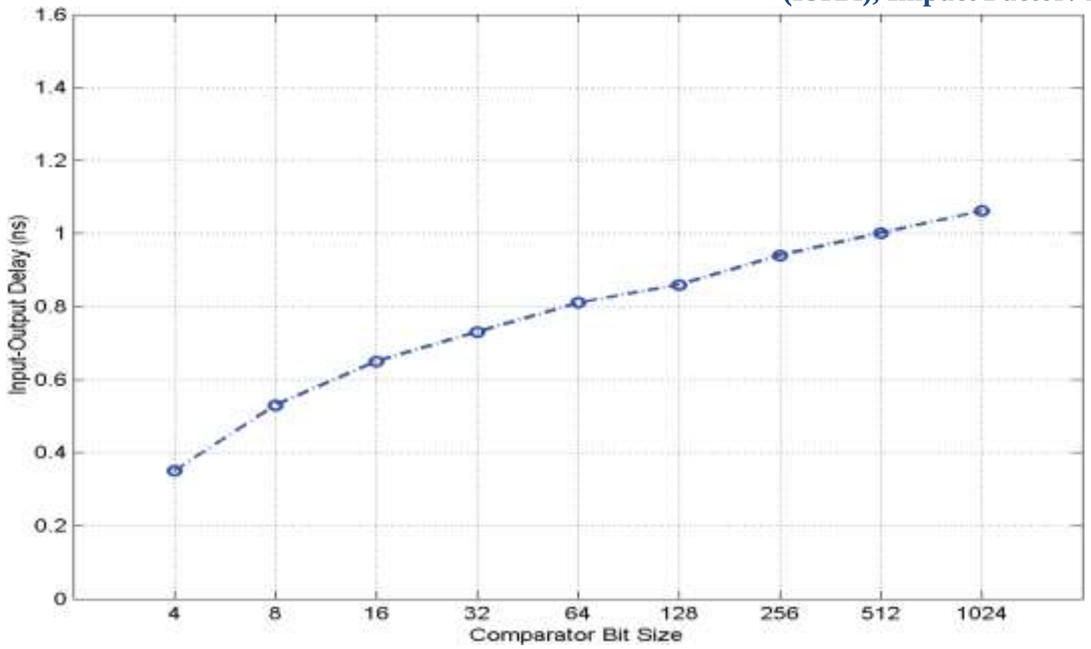


Fig.5. Maximum input-output delay versus input bit-width for our proposed comparator design

Table IV: Total number of Cells and Circuit Levels in Each Set for Comparator band width

Comparator Bit	Set- 1		Set- 2		Set- 3		Set- 4		Set- 5	
	Cells	Levels	Cells	Levels	Cells	Levels	Cells	Levels	Cells	Levels
16	16 '¥	1	4	1	4	1	16 *Q*	1	16	1
32	32 '¥	1	8	1	8	1	32 *Q*	1	32	1

Table V: Total Number of Transistors for Comparator Band width

Comparator Bit	Transistor					
	Set -1	Set 2	Set- 3	Set- 4	Set- 5	Total
16	16 × 12	4 × 8	4 × 8	16 × 20	16 × 12	768
32	32 × 12	8 × 8	4 × 8	32 × 20	32 × 12	1424

Conclusion

In this paper, we presented a n bit high-speed low-power comparator using regular digital hardware structures consisting of two modules: the comparison resolution module and decision module. The structured modules are parallel prefix trees with repeated cells in the form of simple stages that are one gate level deep. Leveraging the parallel prefix tree structure for our comparator design is the design that performs the comparison process from the most significant bit to the least significant bit, using the parallel operation, rather than rippling. Regardless of the comparator bit-width, our structure guarantees that less than 35% of all of the transistors used in the design are active during operation. Additionally, all the cells are locally interconnected, that avoid the requirements for large cell drivers, thus balancing all cells to a uniform transistor size. Simulation results with standard CMOS transistor cells revealed operating speeds of 1.2 and 1 GHz for 16 bit and 32 bit comparators, correspondingly, in a 0.15-μm CMOS process and worst case operands. These results translate to a 40% speed advantage over state-of-the-art fast comparators. Furthermore, simulation results confirmed our comparator's power effectiveness, with a power dissipation of 0.9 μ W/MHz on average and 4.12 μ W/MHz in the worst case when 32 bits or more of the inputs must be evaluated.

References

1. *H. L. Helms, High Speed (HC/HCT) CMOS Guide. Englewood Cliffs, NJ: Prentice-Hall, 1989.*
2. *SN7485 4-bit Magnitude Comparators, Texas Instruments, Dallas, TX, 1999.*
3. *K. W. Glass, "Digital comparator circuit," U.S. Patent 5 260 680, Feb 13, 1992*

4. *D. Norris, "Comparator circuit," U.S. Patent 5 534 844, Apr. 3, 1995.*
5. *M. D. Ercegovac and T. Lang, Digital Arithmetic, San Mateo, CA: Morgan Kaufmann, 2004.*
6. *J. P. Uyemura, CMOS Logic Circuit Design, Norwood, MA: Kluwer, 1999.*