e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 05 Issue 04
February 2018

Design of Low Power & High Speed Parallel Prefix Comparator

V.Venkata Arjun & Lakshmi Prasad.Ch
[1]Student of VLSI, [2]Assistant Professor
E.C.E Department, Universal College of Engineering&Technology, Dokiparru
Email: [1] venkata.arjun98@gmail.com [2]lakshmiprasad0637@gmail.com

Abstract—*In this paper we proposes a comparator design using digital CMOS cells featuring wide-range and high-speed operation. The Comparison is most basic arithmetic operation that determines whether one number is greater than, less than or equal to the other number. Our comparator uses a novel scalable parallel prefix structure that leverages the comparison outcome of the MSB, proceeding bitwise towards LSB only when the comparison bits are equal. This comparator is composed of locally interconnected CMOS gates with a maximum fan-in of five and fan-out of four, independent of comparator bandwidth. Comparator is most fundamental component that performs comparison operation. Comparison between modified and existing 8-bit binary comparator using parallel prefix designs is calculated by simulation performed at 90nm technology in DSCH, Microwind Tool and simulated with Xilinx ISE13.1. The main advantages of our proposed design are high speed and power efficiency, maintained over a wide range.*

Index Terms—High-speed arithmetic, high-speed wide-bit comparator architecture, parallel prefix tree structure.

I. INTRODUCTION

Comparator is a basic arithmetic unit that compares the magnitude of two binary numbers, say A and B, and produces output bits: A>B or A<B or A=B. It is an important data-path element for any general purpose architecture as well as an essential device for application-specific and signal processing architectures. Comparators are also used in sorting networks which play an important role in areas such as parallel computing, multi-access memories and multiprocessing.

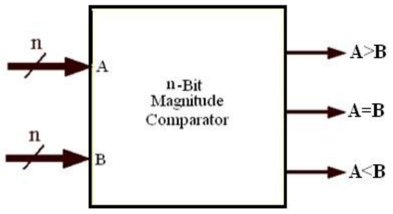


Fig 1. Block Diagram of n-Bit Magnitude Comparator

Comparator forms a fundamental component of processors and digital systems. For processors, in order to achieve high throughput with fast clock rates, it is necessary that such devices have less delay. Consequently, the designing of high speed comparator architecture becomes a relevant and essential research topic. The serial architecture is suitable for short inputs (i.e. when both the inputs have lesser number of bits). For longer inputs (say, 32 bit, 64 bit inputs), the circuit complexity and the combinational delay increase drastically. As a result, parallel approach is generally preferred for comparators with longer inputs. The comparator designs presented in this paper are based on parallel approach.

II. ARCHITECTURE OVERVIEW

The comparison resolution module in Fig. 1 (which depicts the high-level architecture of our proposed design) is a novel MSB-to-LSB parallel-prefix tree structure that performs bitwise 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 module performs the bitwise comparison asynchronously from left to right, such that the comparison logic's computation is triggered only if all bits of greater significance are equal. The bitwise comparison results are stored into two Nbit

buses using parallel prefix structure. The two buses are the left bus and the right bus, each of which store the partial comparison result as each bit position is evaluated,such that

ifAx>Bx, then leftx= 1 and rightx= 0.

ifAx<Bx , then leftx= 0 and rightx= 1.

ifAx= Bx , then leftx= 0 and rightx= 0.

To reduce switching activities, if the bitwise comparison is not equal, the bitwise comparison of all the bits of lower significance is terminated and all such positions are set to zero on both buses. Hence, there is never more than one high bit on either bus.The decision module uses NOR-NAND networks to output the final comparison decision based on all of the bits on the left bus producing the Lbbit and all of the bits on the right bus producing the Rbbit. If LbR b= 00, then A = B, if LbRb= 10 then A > B, if LbRb= 01 then A < B, and LbRb= 11 is not possible.

A 4-b comparison of input operands A = 1000 and B =0101 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. In this example, A3= 1 and B3= 0 encodes as Lb3=1 and Rb3= 0. At this point, wherethe bits are unequal, the comparison terminates and a final comparison decision can be made based on the first bit evaluated.

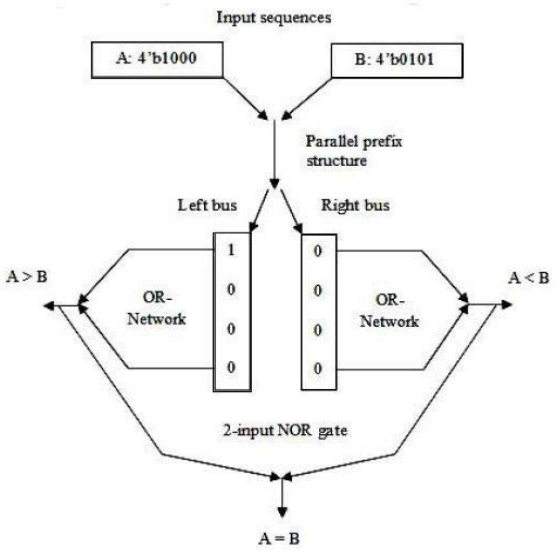


Fig. 2. Example 4-b comparison

The parallel prefix structure fixes all bits of lower significance of left and right 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, and the final comparison decision is A> B. The structure is sectioned into five hierarchical prefixing sets, as shown in Fig. 3, with the associated symbolic representations in Tables I and II. Here each group performs a specific function. The output of each group is given as an input to the next group.At last the fifth group produces the output on the left bus and the right bus.

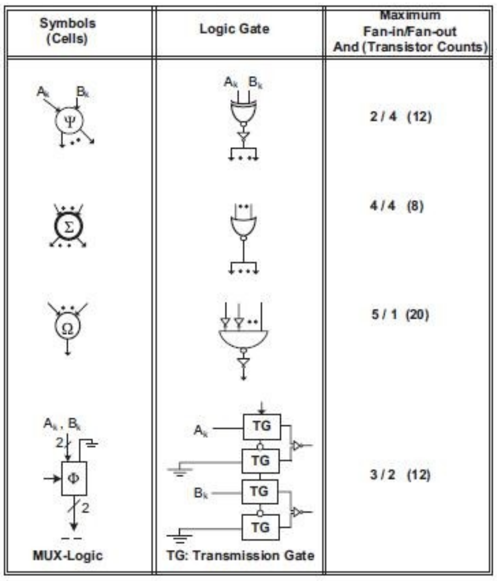
TABLE I

SYMBOL 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

TABLE II
LOGIC GATE REPRESENTATIONS FOR SYMBOLS USED IN FIG. 3

	Scalable 8 bit Comparator			**Proposed 8 bit Comparator**		
Technology	0.18 µm 1.95 V	0.120 µm 1.95 V	90µm 1.95 V	0.18 µm 1.95 V	0.120 µm 1.95 V	90µm 1.95 V
Area	38211	8796.5	6108.7	12322	2838.8	1971.4
Power µW	1.342 mW	0.193 mW	93.67 W	0.217 mW	20.633 µW	16.716 µW



Symbols (Cells)	Logic Gate	Maximum Fan-in/Fan-out And (Transistor Counts)
		2 / 4 (12)
		4 / 4 (8)
		5 / 1 (20)
MUX-Logic	TG: Transmission Gate	3 / 2 (12)

A_k
B_k
Ψ
Σ
Ω
A_k, B_k
2
Φ
2
TG
TG
TG
TG

This prefixing set structure bounds the components' fan-in and fan-out regardless of comparator bitwidth and eliminates heavily loaded global signals with parasitic components, thus improving the operating speed and reducing power consumption. Additionally, the OR-network's fan-in and fan-out is limited by partitioning the buses into 4-b groupings of the input operands, thus reducing the capacitive load of each bus.

III. COMPARATOR DESIGN DETAILS.

In this section, we detail our comparator's design (Fig.3), which is based on using a novel parallel prefix tree (Tables I and II contain symbols and definitions). Each set or group of cells produces outputs that serve as inputs to the next set in the hierarchy, with the exception of set 1, whose outputs serve as inputs to several sets

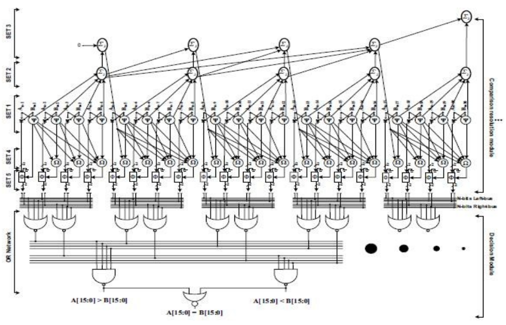


Fig.3 Implementation details for the comparison resolution module (sets 1 through 5) and the decision module.

Set 1 compares the N-bit operands A and B bit-by-bit, using a single level of N ψ-type cells. The N-type cells provide a termination flag D_k to cells in sets 2 and 4, indicating whether the computation should terminate.

Set 2 consists of $\sum_2$-type cells, which combine the termination flags for each of the four Ψ-type cells from set 1 (each $\sum_2$-type cell combines the termination flags of one 4-b partition) using NOR-logic to limit the fan-in and fan-out to a maximum of four. The $\sum_2$-type cells either continue the comparison for bits of lesser significance if all four inputs are 0s, or terminate the comparison if a final decision can be made.

Set 3 consists of $\sum_3$-type cells, which are similar to $\sum_2$-type cells, but can have more logic levels, different inputs, and carry different triggering points. A $\sum_3$-type cell provides no comparison functionality; the cell's sole purpose is to limit the fan-in and fan-out regardless of operand bit width. To limit the $\sum_3$-type cell's local interconnect to four, the number of levels in set 3 increases if the fan-in exceeds four. 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, set 3 signals set 4 to set the left bus and right bus bits to 0 for all bits of lower significance.

From left to right, the first four $\sum_3$-type cells in set 3 combine the 4-b partition comparison outcomes from the one, two, three, and four 4-b partitions of set 2. Since the fourth $\sum_3$-type cell has a fan-in of four, the number of levels in set 3 increases and set 3's fifth $\sum_3$-type cell combines the comparison outcomes of the first 16 MSBs with a fan-in of only two and a fan-out of one.

Set 4 consists of Ω-type cells, whose outputs control the select inputs of ϕ -type cells (two-input multiplexors) in set 5, which in turn drive both the left bus and the right bus. For an Ω-type cell and the 4-b partition to which the cell belongs, bitwise comparison outcomes from set 1 provide information about the more significant bits in the cell's Ω type cells.

The number of inputs in the Ω-type cells increases from left to right in each partition, ending with a fan-in of five. Thus, the Ω type cells in set 4 determine whether set 5 propagates the bitwise comparison codes.

Set 5 consists of N ϕ -type cells (two-input, 2-b-wide multiplexers). One input is (A_K, B_k) and the other is hardwired to "00." The select control input is based on the Ω-type cell output from set 4. We define the 2-b as the left-bit code (A_K) and the right-bit code (B_k), where all left-bit codes and all right-bit codes combine to form the left bus and the right bus, respectively. The output $F_k^{1,0}$ denotes the "greater-than," "less-than," or "equal to" final comparison decision Essentially, the 2-b code F1,0 k can be realized by OR-ing all left bits and all right bits separately as shown in figs 2 and 3.

IV. PROPOSED 8 BIT COMPARATOR:

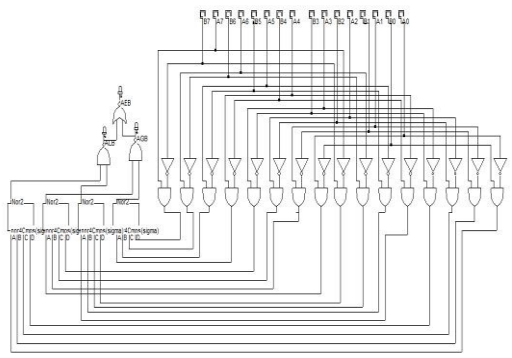


Fig. 4 Proposed 8 Bit Comparator

In this section, The Proposed comparator design has the two modules one is comparison resolution module and another one is decision module. In this only comparison resolution module is modified and decision module is same as of the conventional comparator design modules. This is shown in fig 5.1

Comparison resolution module is reducing the No. of Transistors. This comparator Comparison resolution module using only one AND gate and NOT gate. The AND gate input is connected to the NOT gate output. This is Two sets, one is A0 to A3 and B0 to B3 and another one is A4 to A7and B4 to B7. The A0 to A3 complement of A4 to A7 and B0 to B3 complement of B4 to B7 is connected.

V. SCHEMATIC REPRESENTATIONS OF CONVENTIONAL DIGITAL COMPARATOR.

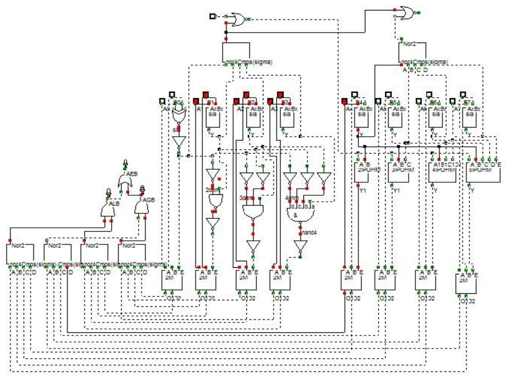
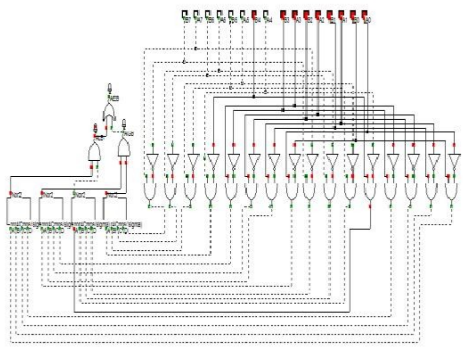


Fig. 5.1 Design of 8 Bit Comparator Using a Parallel Prefix Tree using DSCH Tool



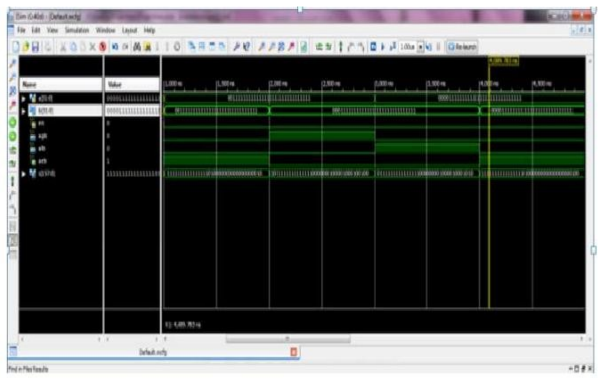


Fig. 5.2 Design of Proposed 8 Bit Comparator using DSCH Tool

VI. SIMULATION RESULTS

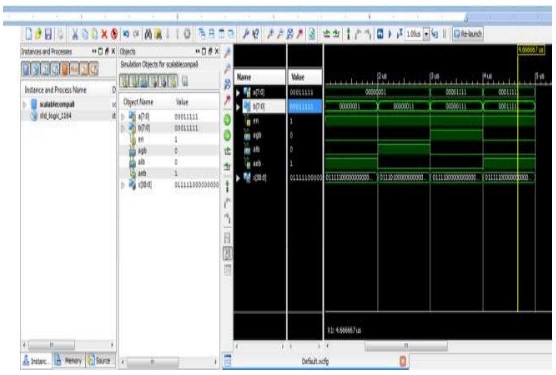


Fig. 6.1 Simulation output of 8 Bit Parallel Prefix Tree comparator using in Xilinx.

TABLE I

COMPARATOR WITH 8 BITS AT DIFFERENT TECHNOLOGY

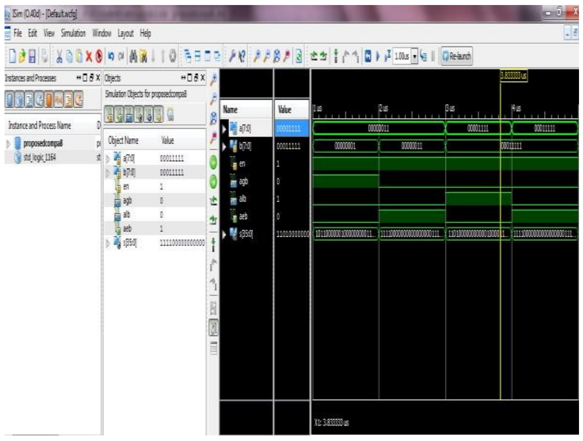


Fig.6.2 Simulation output of proposed 8 Bit Comparator using in Xilinx.

Fig. 6.3 Simulation Results of 32 Bit Comparator Using a Parallel Prefix Tree using in Xilinx Tool.

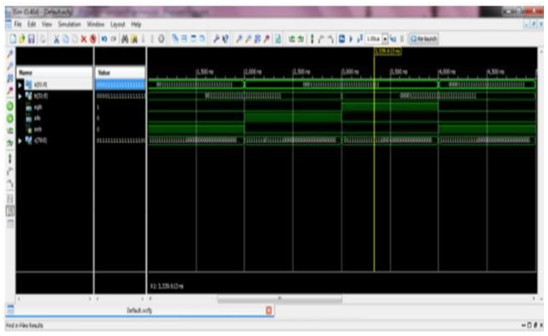


Fig. 6.4 Simulation Results of Proposed 32 Bit Comparator using in Xilinx Tool

VII. CONCLUSION

In this paper, we presented a scalable low power comparator with speed using regular digital structures consisting of two modules: the comparison module and the decision module. These modules are structured as parallel prefix trees with repeated cells in the form of simple gates that are one gate level deep with maximum fan-in and fan-out of four and five respectively, independent of input bitwidth.

The parallel prefix structure of comparator design performs the comparison operation from the MSB to LSB, using parallel operation rather than rippling.

We further modified the comparison module in the comparator design and proposed a new comparator which reduce the No. of transistors then compare the parallel prefix structure. Simulation results for 32bit comparator using Xilinx ISE 13.1.Our simulation analysis showed that power reduction in proposed comparator using parallel prefix structure over existed comparator.

VIII. 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 Trans. Circuits 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.