High-speed comparator architectures for fast binary comparison

Conference Paper · November 2012

DOI: 10.1109/EAIT.2012.6408016

CITATIONS
7

READS
3,402

2 authors:

Suman Deb
Nanyang Technological University
2 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Saurabh Chaudhury
National Institute of Technology, Silchar
150 PUBLICATIONS **1,742** CITATIONS
SEE PROFILE

2012 Third International Conference on Emerging Applications of Information Technology (EAIT)

High-Speed Comparator Architectures for Fast Binary Comparison

Suman Deb Saurabh Chaudhury
Department of Electrical Engineering
National Institute of Technology Silchar,
Silchar, India
Email: saurabh1971@gmail.com

Abstract- **This paper proposes the design of digital comparator with two different parallel architectures. These comparators are first realized in Verilog and simulated with *Xilinx ISE 8.2i* platform and then compared with the traditional design. Simulation results show that the first proposed architecture has 23.769 % less combinational delay (logic + interconnect) and the second proposed architecture is even much faster and has a combinational delay of 35.218 % less compared to the traditional design.**

Keywords- **binary comparator, sorting networks, parallel architecture, compare look ahead logic, tree comparator, combinational path delay.**

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 [1, 2]. Comparators are also used in sorting networks which play an important role in areas such as parallel computing, multi-access memories and multiprocessing [3, 4, 5, 6].

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. Previously published comparator implementations having serial and parallel architecture can both be found in literature. 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.

The rest of the paper is organized as follows: Section II briefly describes the design of traditional comparator. The proposed architectures and their 32-bit level implementation are presented in section III. Section IV illustrates the simulation results and their comparison. The final conclusive remarks about the proposed architectures are made in section V.

II. TRADITIONAL ARCHITECTURE

Traditional magnitude comparator [7] is shown in Fig.1. The design is a parallel architecture. The circuit has three output bits: *A>B*, *A<B*, *A=B*. In many applications, only two output signals *A>B*, *A<B* are sufficient. The output bit (*A=B)* goes *high* if all the bits of *A* are equal to the corresponding bits of *B*. The two output signals *A>B* and *A<B*, are determined based on the following two conditions.

- If MSBs of the two numbers are unequal, i.e when A_i=1, B_i=0 then *A>B* or A_i=0, B_i=1 for *A<B*.
- OR if the pair of bits in the significant bit positions are equal, and LSBs are different i.e. A_i=*Bi* and A_{i-1} = 1, B_{i-1}=0 then *A>B* or A_i=*Bi* and A_{i-1}=0, B_{i-1}=1 then *A<B*.

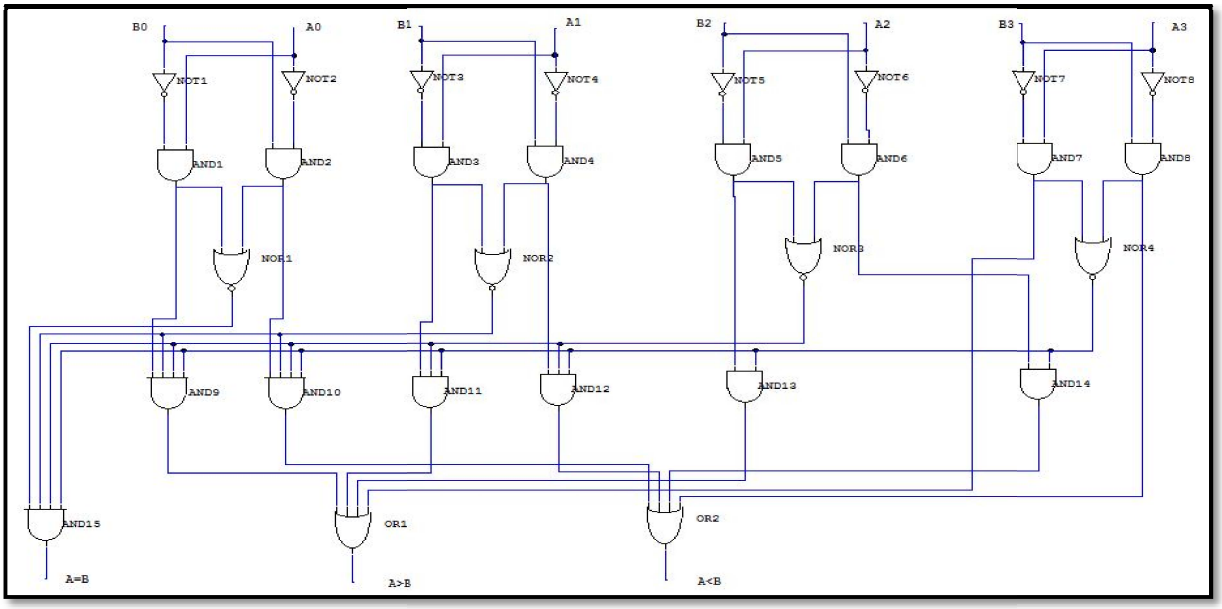

Fig. 1: Traditional Binary Comparator

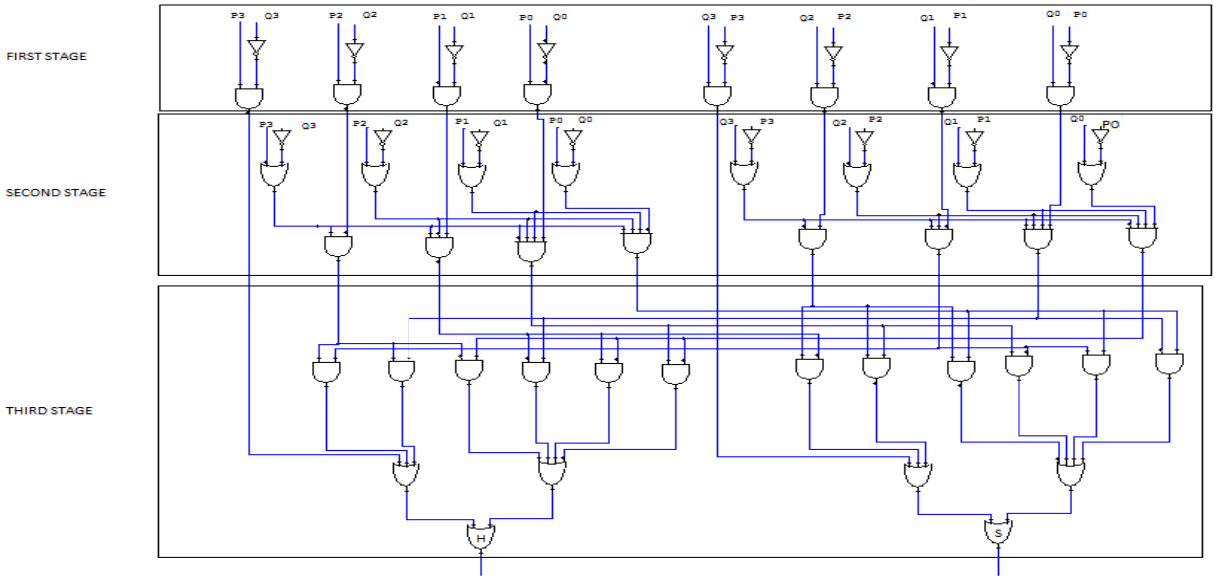

Fig. 2: Proposed Architecture-I

III. PROPOSED ARCHITECTURES

A. Proposed Architecture-I

The first proposed architecture presented in this paper is based on parallel approach and has two output bits: H(i.e. $A>B$), S(i.e. $A<B$). The circuit for the 4-bit design of this comparator is displayed in Fig. 2 and is slightly a modified version of the traditional comparator (which works on bit-weight comparison of two numbers from LSB to MSB).

To understand the logic for the proposed architecture, let us consider an example for the comparison of A=1011_2 and B =1100_2. In the first stage, we identify and extract the 1s of first number which have a 0 in the corresponding position of the second number and are allowed to remain. The basic idea behind this is that only such 1s of a number make it greater than the other number. All other bit positions which have a 1 in the corresponding position of the other number, are made 0. This is done for both the numbers in parallel, that is, A with respect to B (i.e. $A_i\overline{B_i}$) and B with respect to A (i.e.

$B_i\overline{A_i}$) , thereby forming two numbers *A'* and *B'* as shown below.

A = 1 0 1 1	*B* = 1 1 0 0
B = 1 1 0 0	*A* = 1 0 1 1
A' = 0 0 1 1	*B'* = 0 1 0 0

In the second stage, only the most significant 1s of *A'* and *B'* are extracted by giving it higher priority. Other 1s are made 0. This stage incorporates logic similar to the *priority* logic of a priority encoder. This way two new numbers, *A''* and *B''* are formed as shown below. Due to the *priority* logic incorporated, the number of 1s in *A''* and *B''* is either one or zero.

A' = 0 0 1 1	*B'* = 0 1 0 0
A'' = 0 0 1 0	*B''* = 0 1 0 0

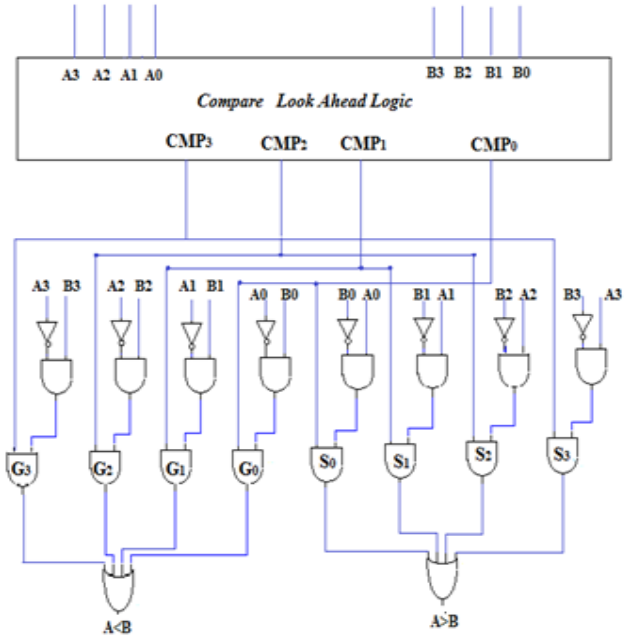

Fig. 3**:** Proposed Architecture II

In the final stage, from *A''* and *B''* two new signals are extracted. These are H (i.e. *A*>*B*) and S (i.e. *A*<*B*), both are of single bit, obtained by extracting the most significant bit (1) from *A''* and *B''*. If the 1 of *A''* is in a more significant position than that of *B''* or if *B''* has all 0s but *A''* has a 1, then this 1 is used to form output bit H. Similarly, if the 1 of *B''* is in a more significant position than that of *A''* or if *A''* has all 0s but *B''* has a 1, then this 1 is used to form output bit S as follows.

A'' = 0 0 1 0	*B''* = 0 1 0 0
B'' = 0 1 0 0	*A''* = 0 0 1 0
H = 0	S = 1

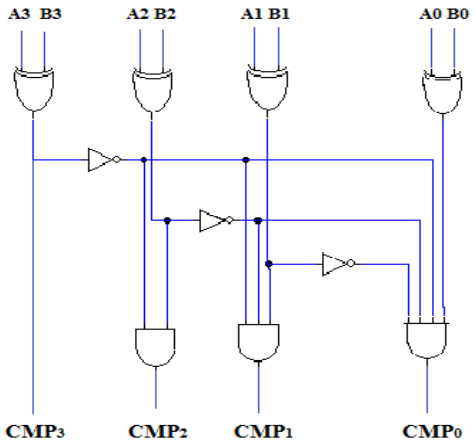

Fig. 4: Compare Look Ahead Logic

B. Proposed Architecture-II

The second architecture proposed in this paper can be called a *look-ahead* comparator and has a parallel approach. It has two output bits: *A*>*B* and *A*<*B*. The circuit of the 4-bit design of this comparator is displayed in Figure 3.

The first stage of this architecture employs a *compare look-ahead* logic as shown in Fig. 4. This *look-ahead* section generates *compare* signals *CMPi* for each bit position, *i'* (*i*= 0, 1, 2, 3). The *CMPi* signal goes *high* if:

- The i^{th} bits of *A* and *B* are unequal or
- The MSBs are equal and more significant positions of *A* and B are unequal.

For a given pair of numbers, only one of the *compare* signals will be *high*. A high *CMPi* will activate *Gi* and *Si* gates. If *Ai =1 and Bi* = 0, output of *Gi* and hence, (*A*>*B*) goes *high*. On the other hand, if *Ai =0 and B*i =1, output of *Si* and hence, (*A*<*B*) goes *high*.

C. Modified comparator module for 32-bit tree Structure

The schematic for 32-bit level implementation of the traditional and proposed comparators is shown in Figure 5. The blocks of the first stage compute the comparison result for every 4 bits of the input numbers. The blocks in the second stage take the result of four sets of 4-bit numbers and compute the result for the two 16-bit numbers which are obtained when the four sets of 4-bit numbers are concatenated. This logic is repeated in the third stage where the 2-bit block takes the results of two sets of 16-bit numbers and computes the result for the two 32-bit numbers.

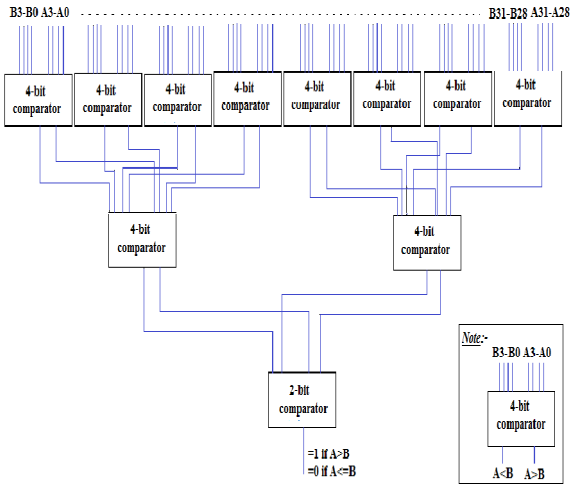

Fig. 5: 32-bit tree structure comparator

In the 32-bit level implementation of both the proposed comparators, a modified 2-bit comparator module has been utilized. Since the numbers input to the 2-bit comparator module are the outputs of 4-bit comparators, certain pairs of numbers can never be the input combinations: (10,10), (10,11), (11,10), (11,01), (01,11), (01,01). This is because the (A>B) and (A<B) output bits of the 4- bit comparator module can never be 1 at the same time. As a result, the Boolean expression for the (A>B) output of 2-bit comparator module becomes:

$$(A>B) = A_{01} + \overline{A}_{01} A_{00} \overline{B}_{01} \overline{B}_{00}$$

The logic-level circuit diagram of the modified 2-bit comparator module is shown in Fig. 6.

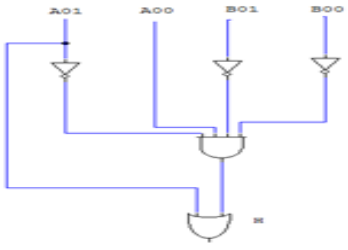

Fig. 6: Modified 2-bit comparator

IV. Simulation And Results

All the simulations of the 32-bit level implementation of the traditional and proposed comparators have been carried out using *Verilog HDL* programming in *Xilinx ISE 8.2i* platform. Adequate testing of each design was done to verify correct operation. *Xilinx* chip series details used for simulation and comprehensive analysis are mentioned as under:

Family: *VirtexE*

Device: *XCV200E*

Package: *FPG456*

The results obtained after the simulation of the traditional and the proposed comparators are summarized in Table 1.

TABLE 1: Delay Comparison of Comparator Designs for 32-bit operation

Architecture	Logic Delay (ns)	Routing Delay (ns)	Maximum Delay (ns)
Traditional	9.924	8.280	18.204
First Proposed	7.805	6.072	13.877
Second Proposed	7.009	4.784	11.793

Referring to the above table, it can be seen that there is a decrease in routing delay as well as logic delay of both the proposed comparators in comparison to the traditional comparator. This leads to an overall reduction in their combinational path delay.

V. CONCLUSION

The proposed comparators have been discussed, simulated and compared with the traditional one. Simulation results show 23.77 % decrease in path delay in case of the proposed architecture-I and 35.22% decrease in path delay in case of proposed architecture-II over traditional architecture.

REFERENCES

[1] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers, 2002.

[2] J. Eyre and J. Bier, "DSP processors hit the mainstream," IEEE Computer, pp. 51–59, 1999.

[3] Shun-Wen Cheng, "Arbitrary Long Digit Sorter HWISW Co-Design," in Proc. Asia and South Pacific Design Automation Conj, ASPDAC,, 03, pp. 538-543, Jan. 2003.

[4] D. E. Knuth, Sorting and Searching. Reading: Addison-Wesley, 1973.

[5] D. Norris, "Comparator circuit," U.S. Patent 5,534,844, April 3, 1995.

[6] Shun-Wen Cheng, "A high-speed magnitude comparator with small transistor count" in Proceedings of the 2003 10th IEEE International Conference on Electronics, Circuits and Systems (ICECS), pp. 1168- 1171, 2003

[7] M. Morris Mano, Digital Logic and Computer Design.