JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

LOW ELABORATION TREE ARCHITECTURE FOR FINDING THE INITIAL MINIMA

KEERTHIKA NIMMAKAYALA,
Mr.B.T.S.S. SRI HARI KRISHNA GOPAL, Professor, Department of Electronics & Communication Engineering.
International School of Technology & Sciences (Approved by AICTE & Affiliated to JNTUK Kakinada), Rajahmundry, India

Abstract:

This summary presents an area-efficient tree architecture for finding the first two minima and the indices of the first minima, essential for the design of low-density parity-check decoders based on min-sum algorithms.This proposed architecture decreases the usage of the number of comparators by using the intervening comparison results again and again which are computed for the first minimum to collect second minimum candidates.As a result, the proposed tree architecture significantly improves the spatial-temporal complexity. In this paper, we propose the design of a digital comparator with two different parallel architectures. These comparators are first implemented in Verilog, simulated on the Xilinx ISE 14.6 platform, and then compared to conventional designs. Simulation results shows that the initially proposed architecture decreases the combinatorial delay (logic + interconnect) and the second proposed architecture is even much faster, with a very small combinatorial delay compared to the conventional design. Here in this project, we have extended our project to 128 bit architecture by using 4-bit comparator.

Introduction:

A digital Comparator is a combinational circuit that **compares two digital or binary numbers** in order to find out whether one binary number is equal, less than, or greater than the other binary number. We logically design a circuit for which we will have two inputs one for A and the other for B and have three output terminals, one for A > B condition,

One for A = B condition, and one for A < B condition. It is an important data path component for all basic intent structures and is also an important device for specific applications and transmission control architectures. It is a critical data path component for all basic intent structures and a critical device for application-specific and transport control architectures. The consortium also has comparators, which play an important role in parallel research, multiple access memory, multiple processing, etc. Comparator is an important part of processors along with digital camera devices. When it comes to processors, these types of units need low latency to deliver high throughput along with fast clock speeds. Planning with excessive pace comparison constructs/designs is therefore another important analysis. Pre-printed implementations of comparators that are serial in a parallel structure may find their way into the literature as well. Serial structures work for limited advice (eg lizabeth). Advice, on the other hand, has fewer bits). For more time constraints (e.g. 33 tad, 64 tad inputs) the business complexity and combinatorial latency increases significantly. Therefore, concurrent tactics are often chosen for comparators along with more time recommendations. The comparator type shown in this document offers a parallel tactic.

II.TRADITIONAL ARCHITECTURE

Traditional magnitude comparator has a parallel architecture design which is shown in Fig.1. The circuit has 3 Output bits: A>B, A<B, A=B.

But in many applications, only two output signals are enough. They are A>B, and A<B. The output bit (A=B) goes high if all the bits of A are equal to the corresponding bit s of B. The two outputs A>B and A<B, are determined/identified based upon the following conditions, which are

• If MSBs of the two numbers are unequal, i.e. when Ai=0, Bi =1 then A<B else A>B.

• OR If MSBs of the two numbers are equal, and LSBs are different i.e. Ai=Bi and Ai-1 = 0, Bi-1 =1 then A<B else A>B.

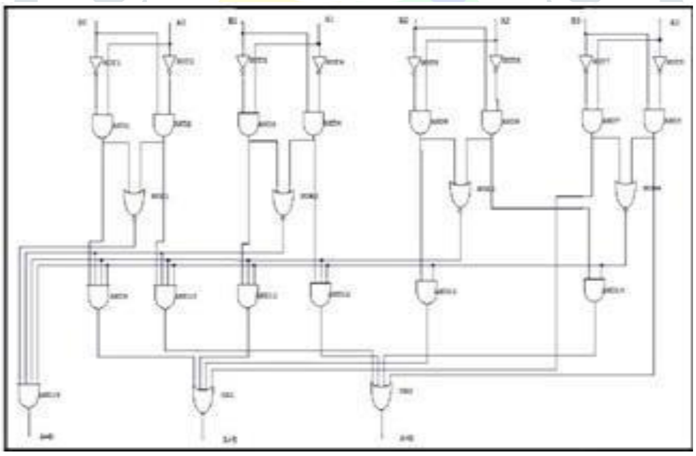


Fig.1. Traditional Binary Comparator

III.PROPOSED ARCHITECTURES

A. Proposed Architecture-I

Since these particular papers rely on a parallel approach, the first suggested structure provided is the papers H (i.e. Age. A > B), S (i.e. Age. A < B). There are pairs of conforming sections available. This circuit for a 4-bit pattern using this comparator can be represented as with only two mappings, and is also a slightly modified version of the traditional comparator (which is 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. The first stage identifies extracts of the first number 1 that have a 0 in the position corresponding to the second number and are allowed to remain.

The basic idea behind this is that only 1 of the numbers is greater than any other number. All other's will be 0's in positions where there is a 1 in the corresponding position's of the other numbers. This is done in parallel for both

numbers. That is, A to B (that is) and AB to A is done by 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=10 11	B=1 100
B=1100	A=1011
A'=0011	B'=0100

The second stage preferentially extracts only the top 1's of A' and B' by giving it the highest priority. This stage contains logic similar to the Priority Logic in the Priority Encoder. Thus, here two new numbers are formed, A" and B", as shown below. Due to the built-in priority logic, the number of 1's in A" and B" is either 1 or zero.

A'=00 11	B'=0 100
A''=0010	B''=0100

In the final stage, two new signals are extracted from A" and B". These are H (that is, A>B) and S (that is, A<B) of which both are single bits obtained by extracting the most significant bits from A" and B". If the 1 in A" is in a significantly higher position than the position of B", or if B" is all zeros and A" is with 1s, then that 1 is used to form the output bit H. Similarly, if the 1 in B" is in a more significant position than A", or if A" are all zeros and B" is 1, then this 1 is used to set the output S which looks like this:

A''=0010	B''=0 100
B''=0100	A''=00 10
H =0	S = 1

B. Proposed Architecture-II

The second structure proposed in this report, called arbitrary look-ahead comparators, employs a parallel approach. The two main points are: A> B and A<B. Theworld for a 4-bit design using Comparator is shown in set 3.

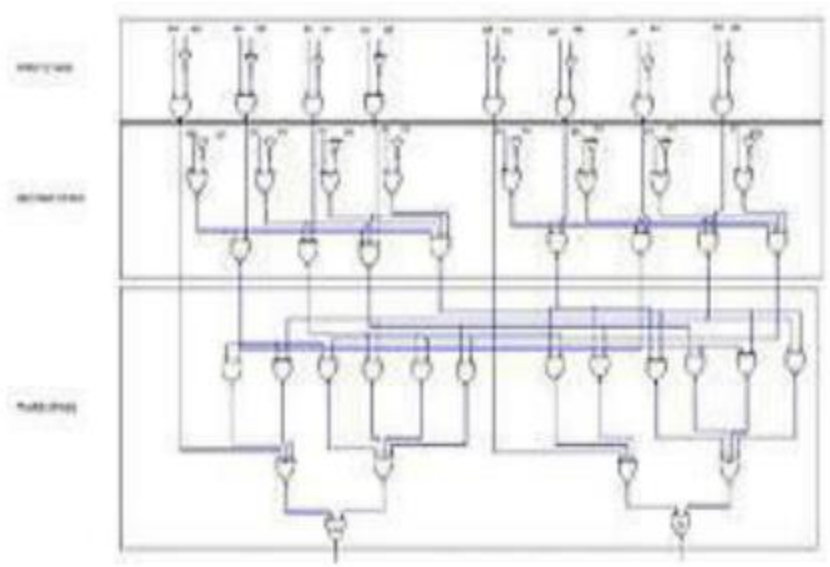


Fig.2. Proposed Architecture-I

In the early stages of these structures, uses any forward thinking as shown in the diagram above. This look-ahead section produces an evaluation signal CMPi for each small bit state(i=0, 1, 2, 3). In that case, the CMPi indicator will be high. For, given only two numbers, only one of the evaluated signals is high. Higher CMPi initializes Giand Cuando inputs. If Ai = 1 and Bi = 0, the final result is high with respect to Gi, hence (A > B). Alternatively, if Ai = 0 and Bi = 1, then, the final result for Cuando and, so (A < B) is higher.

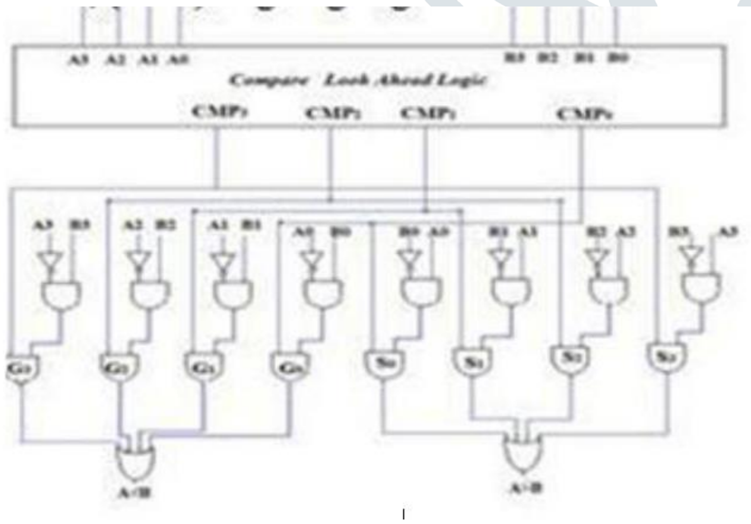


Fig.3. Proposed Architecture II

C. Modified Comparator Module for 32-Bit Tree Structure

The scheme of the conventional comparator and the 32-bit implementation of the proposed comparator is shown in Figure 4. The initial stage block computes the comparison result of all 4 bits of input elements. The second stage block takes four sets of 4-bit number results and computes the two 16-bit number results obtained by concatenating the four sets of 4-bit number. This logic is repeated at in the 3rd stage. Here a 2-bit block takes the result of two sets of 16-bit numbers and computes the result of two32-bit numbers.

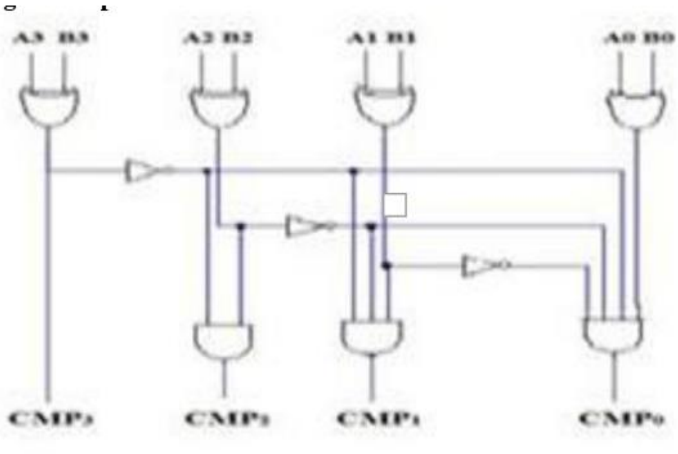


Fig.4. 32-Bit Tree Structure Comparator.

A modified 2-bitcomparator module was used in the 32-bit level implementation of both proposed comparators. Since the inputs to 2-bit comparator module are outputs of the 4-bit comparators, no particular pair of numbers is a combination of inputs: (11,01), (01,11), (01,01),(10,10), (10,11), (11,10).

This is due to that the output bits (A>B) and (A<B) of a 4-bit comparator module are never 1 at the same instance. As a final result, the Boolean expression for the output (A>B) of this 2-bit comparator module is:

- A logic diagram of the altered2-bit comparator module is shown in Figure 5.

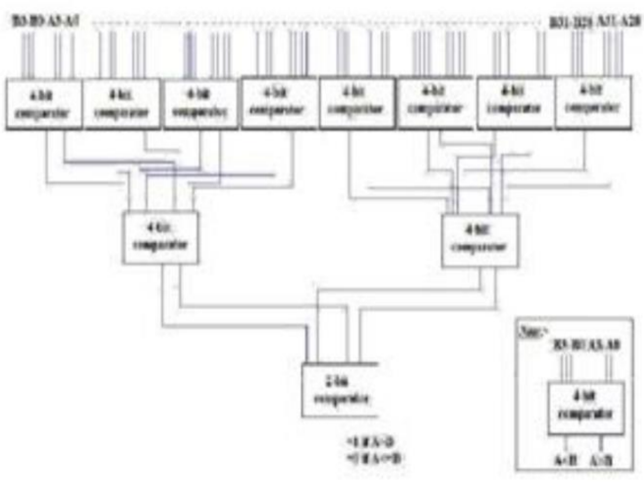


Fig.5. Modified 2-Bit Comparator

IV. SIMULATION AND RESULTS

All simulations of the conventional and proposed 32-bit level implementation of the comparator were performed using Verilog HDL programming on the Xilinx ISE 14.6 platform. Appropriate testing of each design has been performed to ensure that it works correctly. Details of the Xilinx chip series used for simulation and comprehensive analysis are referenced as at the following URL:

Fig.6. 32-Bit Comparator Simulation Waveform.

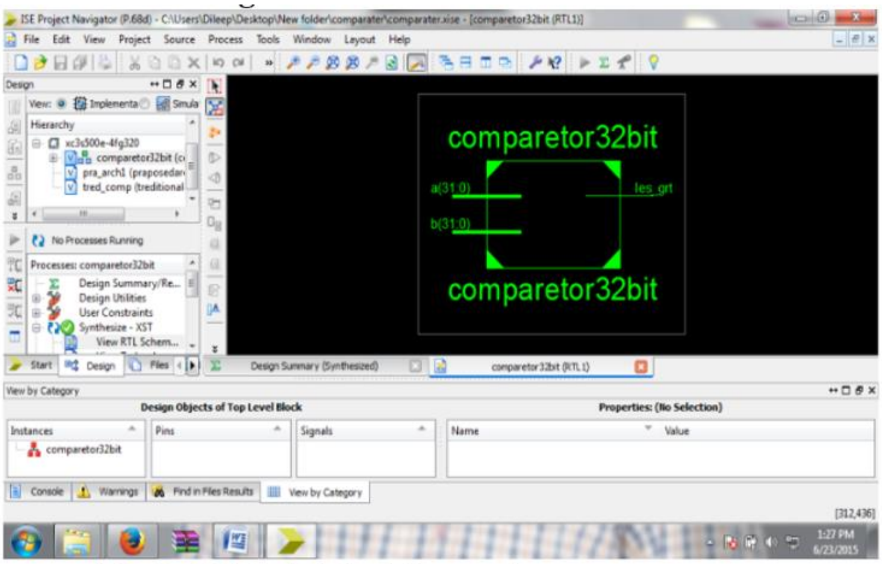


Fig.7. 32 Bit RTL Schematic.

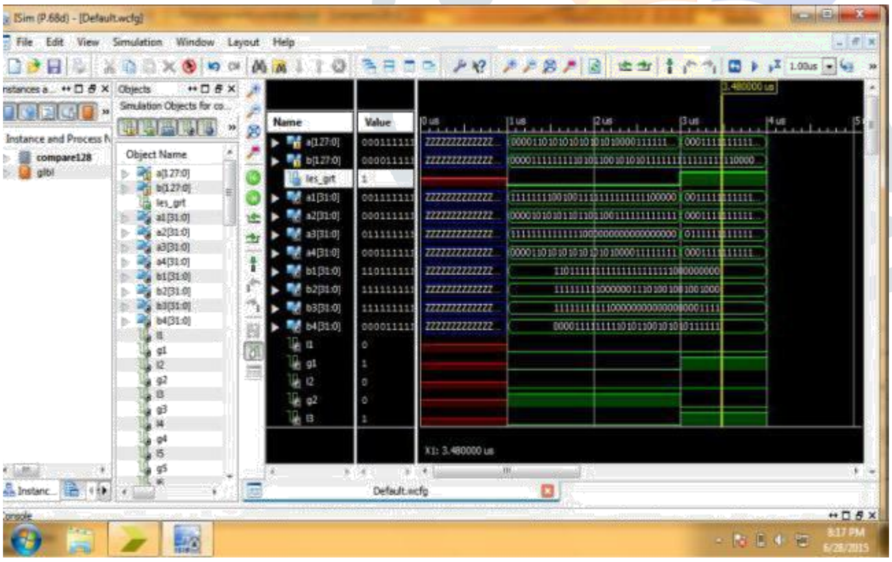


Fig.8. Proposed 128 Bit Comparator Waveform.

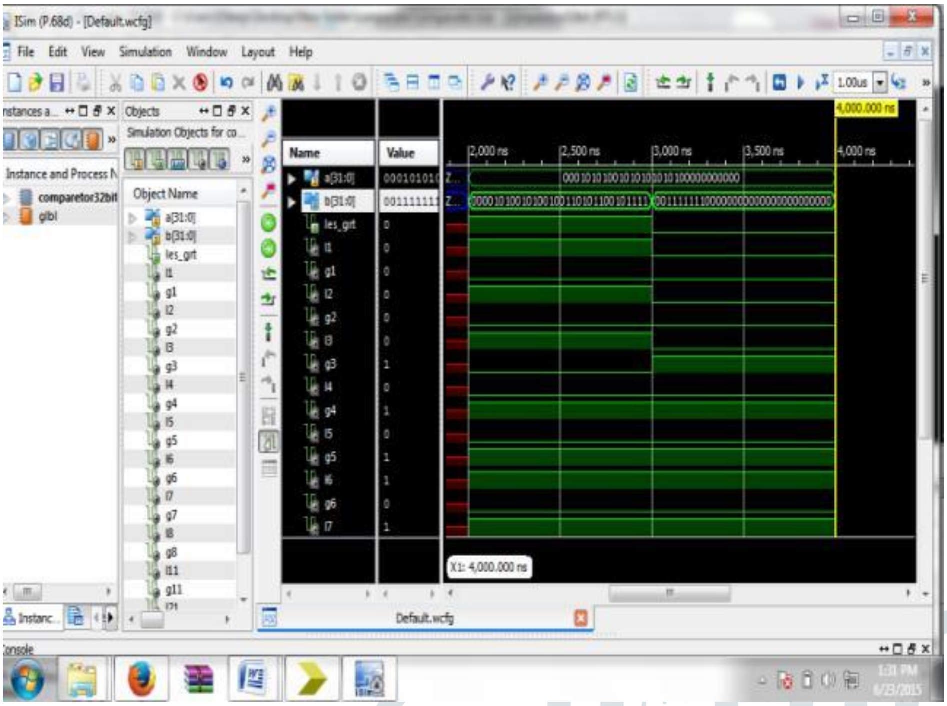
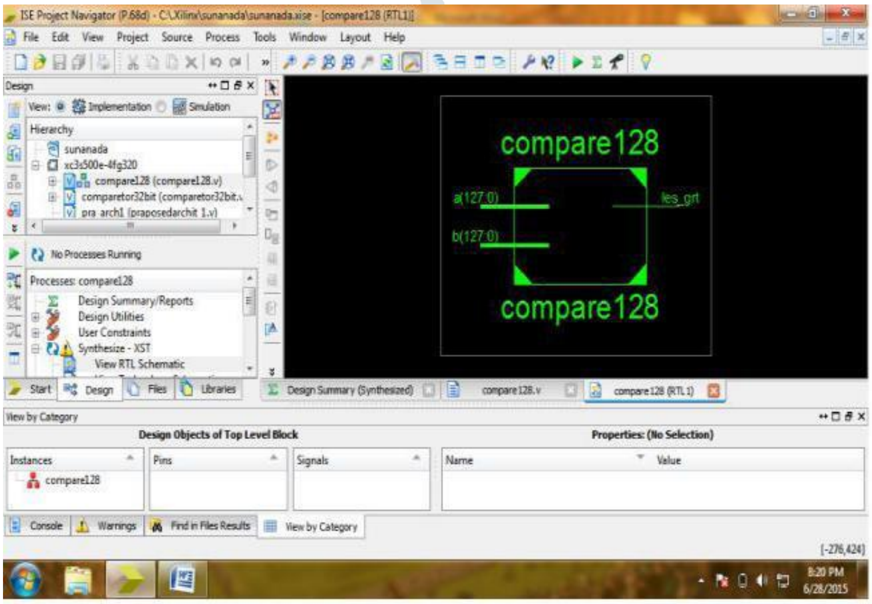


Fig.9. 128-Bit Comparator RTL Schematic.



V. CONCLUSION

The presented comparators are debated, simulated and differentiated with conventional ones. The simulation results in Section IV show the path delay reduction for the proposed architecture-I, and we also designed a 128-bitcomparator for comparing large amounts of data. These designs are simulated and synthesized by the Xilinx ISE 14.6 tools.

VI. REFERENCES

1. Verilog hdl by padmanabam
2. www.asicworld.com
3. J.L.Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach. Morgan Kaufmann Publishers, 2002.

4. J. Eyre and J. Bier, “DSP processors hit the mainstream,” IEEE Computer, pp. 51–59, 2016.

5. 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. 2015.