

PARALLEL PREFIX TREE 32-BIT COMPARATOR AND ADDER BY USING SCALABLE DIGITAL CMOS

Tulluri.Jyotsna, Mrs.J.Mrudula

ABSTRACT

Comparators and adders 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 (associative memories, load-store queue buffers, translation look-aside buffers, branch target buffers), and many other CPU argument comparison blocks. Even though comparator logic design is straight forward, the extensive use of 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. These structures require $\log_2 N$ comparison levels, with each level consisting of several cascaded logic gate but with prefix tree structures area and power consumption can be improved by leveraging two-input multiplexers at each level and generate-propagate logic cells on the first level which takes advantage of one's complement addition. This design uses all N Transistor (ANT) circuits to compensate for high fan-in with high pipeline throughput

Key words- Binary to excess -1 converter (BEC), Variable stage Carry Select, Adder, Clock Select Adder with Sharing (CSAS), Logic design procedure.

I. INTRODUCTION

Comparators design improve scalability and reduce comparison delays using a hierarchical prefix tree structure composed of 2-b comparators. 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 wide 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), which takes advantage of one's complement addition. Using this logic composition, a prefix tree requires six levels for the most common comparison bit width of 64 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 consumption, several 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 (ANT) circuits to compensate for high fan-in with high pipeline throughput.

A multiphase 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 to improve operating speed and eliminate long dynamic logic chains. However, 64-b and wider comparators require an multilevel cascade structure, with each logic level consisting of seven nMos transistors connected in

series that behave in saturating mode during operation. This structure leads to a large overall conductive resistance, with heavily loaded parasitic components on the clock signal, which severely limits the clock speed and jitter margin. Other architectures use a multiplexer-based structure to split a 64-b comparator into two comparator stages: 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.

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-b blocks and the result for each block is input into a multiplexer to determine the final comparison decision. Due to this BCL-based design's low transistor count, this design has the potential for low power consumption, but the pre-encoder logic modules preceding 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 to increasing the power consumption and reducing the operation frequency. To alleviate some of the drawbacks of previous designs (such as high power consumption, multi cycle computation, custom structures unsuitable for continued technology scaling, long time to market due to irregular VLSI structures, and irregular transistor geometry sizes), in this paper we leverage standard CMOS-cells to architect fast, scalable, wide range, and power-efficient algorithmic comparators with the following key features

1. 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.

2. A novel MSB-to-LSB parallel-prefix tree structure, based on a reduce switching paradigm and using parallelism at each level (as opposed to a sequential approach), contributes to the speed and energy efficiency of our design.

3. Use of components built from simple single-gate-level logic, with maximum fan-in and fan-out of five and four, respectively, regardless of the comparator bit-width, makes it easy to characterize and accurately model our comparator for arbitrary bit-widths.

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

Some of the applications are USB Transmitter has been developed into a common code (Generalized USB Transmitter) which can be reused for developing the complete USB device stack.

Some of the Low speed and High-speed USB devices, which are presently in the market are (Optical Mouse, Keyboard, Printer, Scanner, Joy Stick, Memory stick, FlashMemor, Mobiles, Video Cameras).

II. PARALLEL PREFIX TREE COMPARATOR

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 parallel structure encodes the bitwise comparison results into two N-Bit buses, the left bus and the right bus, each of which store the partial comparison result as each of which store the partial comparison result as each bit position is evaluated. The comparison resolution module in Fig (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 AN-1, AN-

$2, \dots, A_0$ and $B_{N-1}, B_{N-2}, \dots, B_0$, where the subscripts range from N-1 for the MSB to 0 for the LSB.

$$\begin{cases} \text{if } A_k > B_k, & \text{then left}_k = 1 \text{ and right}_k = 0 \\ \text{if } A_k < B_k, & \text{then left}_k = 0 \text{ and right}_k = 1 \\ \text{if } A_k = B_k, & \text{then left}_k = 0 \text{ and right}_k = 0. \end{cases}$$

In addition, to reduce switching activities, as soon as a bitwise comparison is not equal, the bitwise comparison of every bit of lower significance is terminated and all such positions are set to zero on both buses, thus, there is never more than one high bit on either bus. The decision module uses two OR-networks to output the final comparison decision based on separate OR-scans of all of the bits on the left bus (producing the R bit). If LR=00, then A=b, if LR=10 then A>b, if LR=01 then A<B, and LR = 11 is not possible. An 8-b comparison of input operands A=01011101 and B=01101001 is illustrated in fig .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.

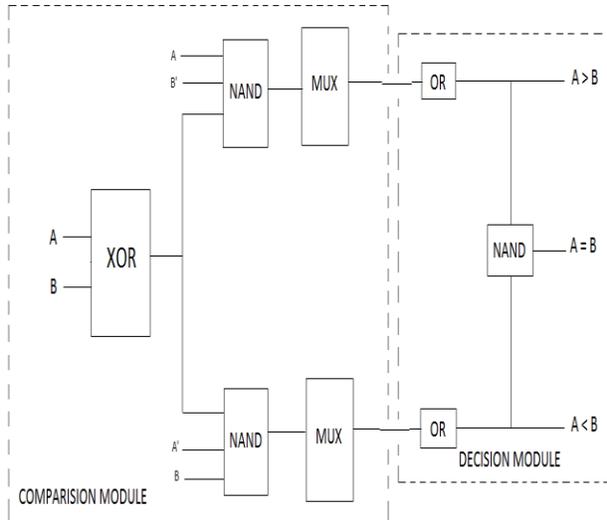


Fig 1. Block diagram of parallel prefix tree comparator,

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 as $left_5=0$ 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, and the final comparison decision is A>B. We partition the structure into five hierarchical prefixing sets, as depicted.

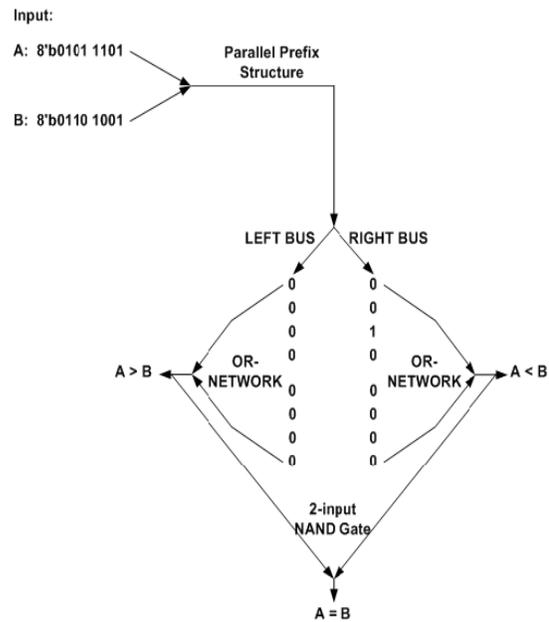


Fig 2. Algorithm for 8-bit comparison

III.PREFIX TREE ADDERS

Parallel-prefix adders, also known as carry-tree adders, pre-compute the propagate and generate signals. These signals are variously combined using the fundamental carry operator

$$(fco). (g_L, p_L) \circ (g_R, p_R) = (g_L + p_L \cdot g_R, p_L \cdot p_R) \quad (1)$$

Due to associative property of the fco, this operator can be combined in different ways to form various adder structures. For, example the four-bit carry-look ahead generator is given by:

$$c_4 = (g_4, p_4) \circ [(g_3, p_3) \circ [(g_2, p_2) \circ (g_1, p_1)]] \quad (2)$$

A simple rearrangement of the order of operations allows parallel operation, resulting in a more efficient tree structure for this four bit example:

$$c4 = [(g4, p4) \circ (g3, p3)] \circ [(g2, p2) \circ (g1, p1)] \quad (3)$$

It is readily apparent that a key advantage of the tree structured adder is that the critical path due to the carry delay is on the order of $\log_2 N$ for an N-bit wide adder. The arrangement of the prefix network give s rise to various families of adders. For a discussion of the various carry –tree structures. For this study, the focus is on the Kogge-stone adder ,known for having minimal logic depth and fan out (see fig) Here we designate BC as the black cell which generates the ordered pair in equation (1) ;the grey cell (GC)generates the left signal only, following. The interconnect area is known to be high, but for an FPGA with large routing overhead to begin with, this is not as important as in a VLSI implementation. The regularity of the Kogge-stone prefix network has built in redundancy which has implications for fault-tolerant designs. The sparse Kogge-stone adder, shown in Fig 1(b), is also studied. This hybrid design completes the summation process with a 4bit RCA allowing the carry prefix network to be simplified.

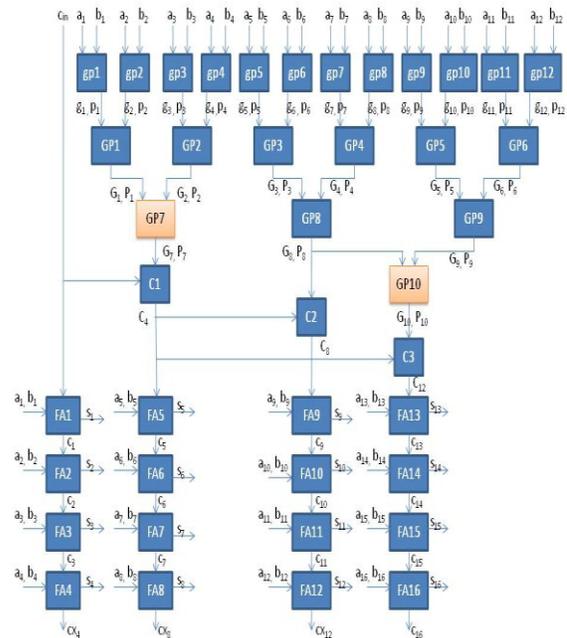


Fig 4. Spanning tree carry-Lookahead Adder

IV.SIMULATION AND SYNTHESIS REPORT FOR COMPARATOR

Simulation Results:

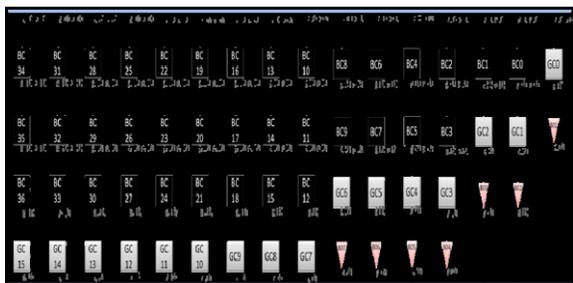
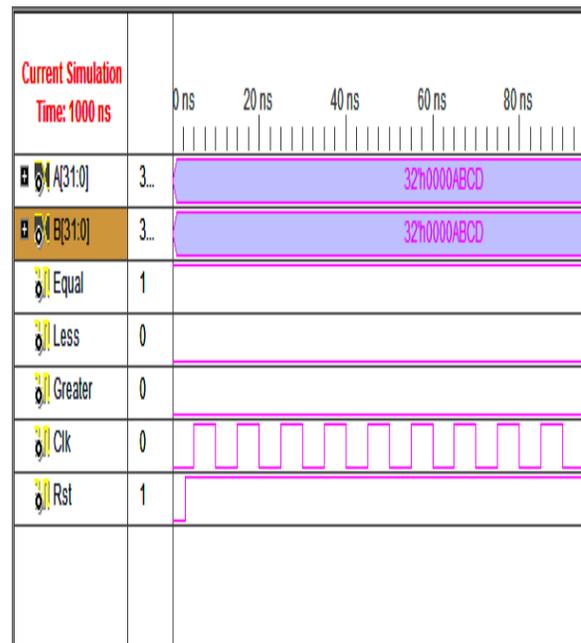


Fig 3. Parallel prefix Carry tree adder design

Another carry-tree adder known as the spanning tree carry-look ahead (CLA) adder is also examined. Like the sparse Kogge-stone adder, this design terminates with a 4-bit RCA.As the FPGA uses a fast carry-chain for the RCA, it is interesting to compare the performance of this adder with the sparse Kogge-Stone and regular Kogge-Stone adders. Also of interest for the spanning –tree CLA is its testability features.

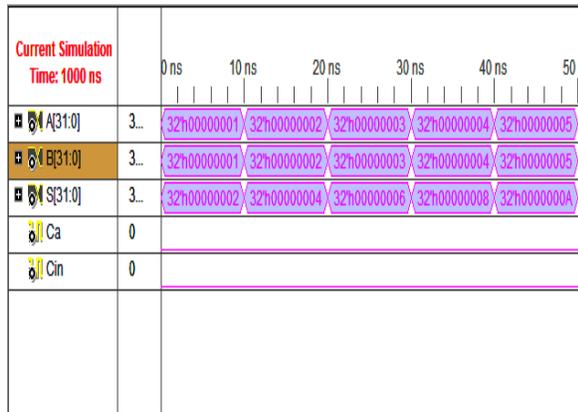


Synthesis Report:

Selected Devices	: 3s1600efg320-5
Number of Slices	: 113 out of 14752
Number of Slice FFs	: 97 out of 29504
Number of 4 input LUTs	: 208 out of 29504
Number of Ios	: 69
Number of bonded IOBs	: 69 out of 250
Number of GCLKs	: 1 out of 24
Max combinational path delay	: No path delay
Total memory usage	: 217004kilobyte

V.SIMULATION AND SYNTHESIS REPORT FOR ADDER

Simulation Results:



Synthesis Report:

Selected Devices	: 3s1600efg320-5
Number of Slices	: 37 out of 14752
Number of 4 input LUTs	: 64 out of 29504
Number of Ios	: 98
Number of bonded IOBs	: 98 out of 250
Max combinational path delay	: 38.041ns
Total memory usage	: 210860kilobyte

VI.CONCLUSION

A 32-bit parallel prefix tree comparator and adder have been designed, verified functionally using VHDL-simulator, synthesized by the synthesis tool, and a final net list has been created. This design is capable of comparing all the equality, greater and lesser comparisons. Because of the structure of the configurable logic and routing resources in FPGAs, parallel-prefix adders will have a different performance than VLSI implementations. The Functional-simulation has been successfully carried out with the results matching with the expected ones.

VII.FUTURESCOPE

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 high-speed zero-detector circuit for the decision module. Given that our comparator is composed of two balance 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 at the expense of increased power and latency.

VIII.REFERENCES

1. F.Frustaci, S.Perri, M.Lanuzza and P.Carsonello,"Energy-efficient single clock cycle binary comparator",Int .J.Circuit Theory Appl.,vol 40,no.3,pp.237-246, March.2012.
2. 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.
3. M. S. Schmookler and K. J. Nowka, "Leading zero anticipation and detection—a comparison of methods," in *Proc. 15th IEEE Symp. Compute. Arithmetic*, Sep. 2001, pp. 7–12.
4. Basic VLSI design, 3rd Edition by Douglas A. Pucknell, and Kamran Eshraghian.

IX ABOUT AUTHOR (S)



Tulluri Jyostna was born in Nandigama, AndhraPradesh. She had done her B. Tech from MIC College of Technology located near Vijayawada in Electronics

Communications Engineering. After completion of B. Tech., she has undergone a training PG diploma in industrial automation in Hyderabad. She presently pursuing M.Tech (VLSI System Design) in Geethanjali college of engineering and Technology,Cheeryal village,Ghatkesar (mandal) Telangana.

2. **Mrs.J.Mrudula, She** is presently pursuing her PhD and she working as a Associate professor in Geethanjali college of Engineering and Technology, Cheeryal village, Ghatkesar (mandal) ,Telangana