Area Efficient Sorting Unit Using Scalable Digital CMOS Comparator

B.Sargunam[#1], S.Srinitha*[2]

[#1]*Assistant Professor, Department of Electronics and Communication Engineering,*
Avinashilingam University, Coimbatore, India.
[*2]*PG scholar, Avinashilingam University, Coimbatore, India.*

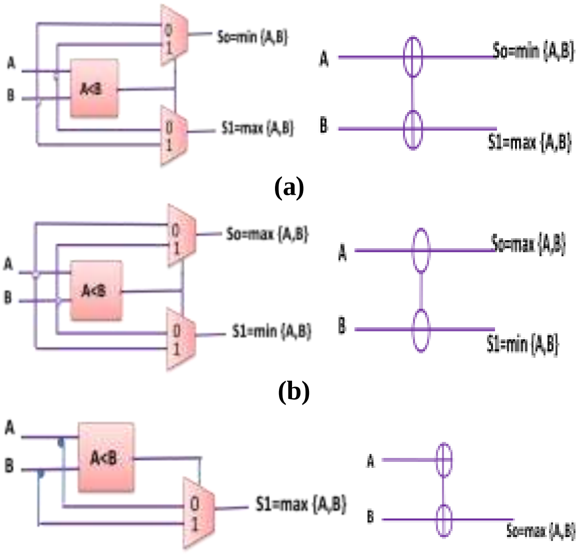
***Abstract*— Sorting is the process of arranging the data into a meaningful order so that we can analyze it more effectively. Sorting is a key requirement in many applications like digital signal processing, scientific computing, network processing etc. This paper presents an area efficient technique for designing high throughput and low latency sorting units. Two popular parallel sorting algorithms are used in this paper, they are Bitonic sorting network and odd-even merge sorting network. These sorting units utilize parallel sorting method which uses Compare-and-Exchange (CAE) blocks. When number of inputs increases, the number of CAE blocks also increases and hence the area increases. To obtain an area efficient sorting network, CAE blocks used in parallel sorting units are replaced with scalable CMOS comparators. Sorting units are coded in VHDL, simulated using Modelsim SE 10.0b and implemented in FPGA using Xilinx ISE for analysis.**

***Keywords*— Scalable CMOS comparator, Bitonic sorting, odd-even merge sorting, max-set-selection, partial sorting.**

I. INTRODUCTION

Sorting is done to arrange an unordered collection of elements into a monotonically increasing or decreasing order.Sorting is an important operation in a wide range of applications including data mining, databases[1],[2], digital signal processing[3], network processing, communication switching systems and high energy physics (HEP) [4].In general,sorting unit return all of its inputs in sorted (increasing or decreasing) order. But in many applicationslike signal processing, data mining and high energy physics only the M largest (or smallest) output values need to be selected from N input values, where M<N.In many HEP applications only the M most energetic particles are selected for analysis. In signal processing applications, only the M strongest signals are needed for analysis. Depending on the application, the M largest (smallest) output values may not need to be in order. The max (min)-set selection units return only the M largest (smallest) outputs. But the partial sorting units return the M largest (smallest) outputs in sorted order.

Sorting networks are comparison networks that sort their inputs [5]. A sorting network is acollection of interconnected CAE blocks that guides a parallel set of inputs to a parallel set of outputs in a sorted order.The high level implementation and schematic symbols of three building blocks used in sorting networks are shown in Fig 1. Each CAE block has two inputs and two outputs. If the inputs are already in order, then they are directed to the corresponding outputs, otherwise the values are swapped.



(a)

(b)

(c)

Fig 1 Schematic symbols of three building blocks used in sorting networks.

There are two types of CAE blocks, they are increasing and decreasing CAE blocks. Fig 1(a) shows an increasingCAE block, which outputs its two inputs in an ascending order, Fig 1(b) shows a decreasing CAE block and it outputs its inputs in descending order. Each CAE block consists of a comparator and two multiplexers. Fig 1(c) shows a max unit, it takes two inputs and returns the larger input.

Two popular parallel sorting networks are Bitonic and odd-even merge sorting network. In Bitonic sorting network, aBitonic sequence is formed by using two sub sequences, one is monotonically increasing and other is monotonically decreasing. Bitonic sorting network recursively merges an ascending and a descending sequence of length N/2 to make a sorted sequence of length N. A K-input Bitonic merging unit (BM-K) contains $\log_2$ (K) stages of parallel CAE blocks and each stage corresponds to a CAE stage with K/2 CAE blocks. BM-K requires $\log_2$ (K)x K/2 CAE blocks.An 8-input Bitonic sorting unit consists of four parallel BM-2 units, two parallel BM-4 unit and one BM-8 unit as shown in Fig 2.

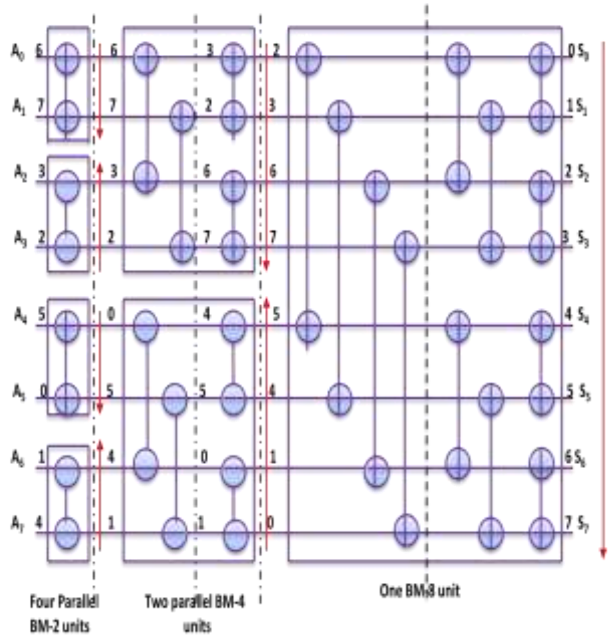


Fig 2 The CAE network for an 8-input Bitonicsorting network

An 8-input odd-even merge sorting unit is shown in Fig 3.

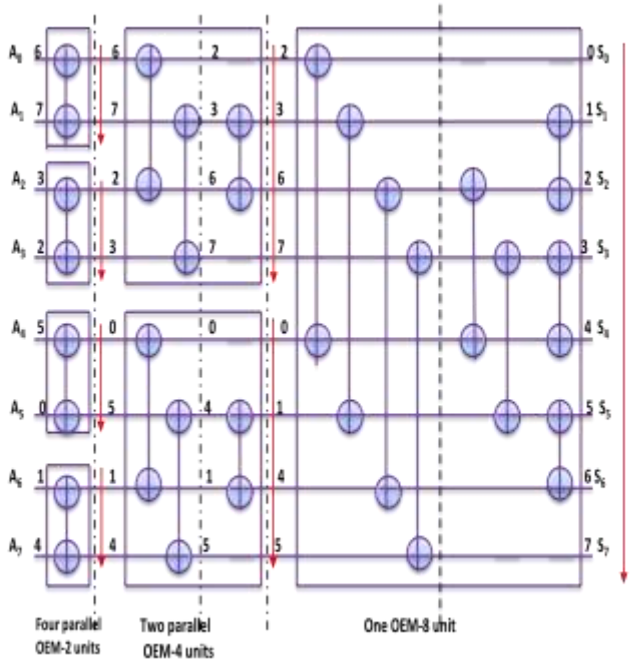


Fig 3 The CAE network for an 8-input odd-even merge sorting network

In odd-even merge sorting network, two ascending sequences of length K/2 are recursively merged to make a sorted sequence of length K. Each odd-even merge sorting network is composed of a number of odd-even merging units. A K-input odd-even merging unit (OEM-K) merges two ascending input sequences into a single ascending output sequence. It contains $\log_2 (K)$ CAE stages, where each stage has between K/4 and K/2 CAE blocks. OEM units are built only from increasing CAE blocks and it is composed of four OEM-2 units, two OEM-4 units and one OEM-8 unit.

The remainder of this paper is organized as follows: Section II deals with the related work. Section III says about the sorting unit using CAE block and Max unit. Section IV implements sorting unit using scalable digital CMOS comparator. Section V describes the simulation and FPGA implementation results. Section VI deals with the conclusion.

II. RELATED WORK

There are several methods for sorting, in this section some of them will be reviewed and their performance will be discussed.

Batcher [5] proposed a Bitonic and odd-even merge sorting networks that are widely used in VLSI and FPGA implementations due to their regularity, simplicity and parallelism.

Lee and Batcher [6] presented a novel recirculating Bitonic sorting network made up of a level of CAE blocks. The purpose of the recirculating network is to reduce the area complexity of original Bitonic sorting networks.

A new parallel sorting algorithm based on odd-even merge sort for shared memory multi processor systems is developed in [7].

A modified odd-even merge sorting network for an arbitrary number of inputs is described in [8] and this approach can be used to implement custom sorting units in hardware.

Farmahini et al.[9] introduced a FPGA-based sorting units used in the large hadron collider. Bitonic sorting network is used in this paper.

Modular design of high-throughput and low latency sorting units is presented and implemented in [10]. In this paper, Bitonic sorting network is compared with the proposed odd-even merge sorting network.

III. SORTING UNITS USING CAE BLOCKS AND MAX-UNIT

Max-set-selection units and Partial sorters [10] are the key components in many applications. Max-set-selection units provide M largest numbers in an arbitrary order and Partial sorting units provides M largest numbers in sorted order. For example, in many HEP applications only the M most energetic particles are needed and in multimedia applications the partial sorters are used to speed up the data sorting algorithms [11].To design 8-to-4 max-set-selection units, only the four largest inputs are needed so that resource requirements and number of CAE stages is reduced. Fig 4 represents an 8-to-4 bitonic max set selection unit.

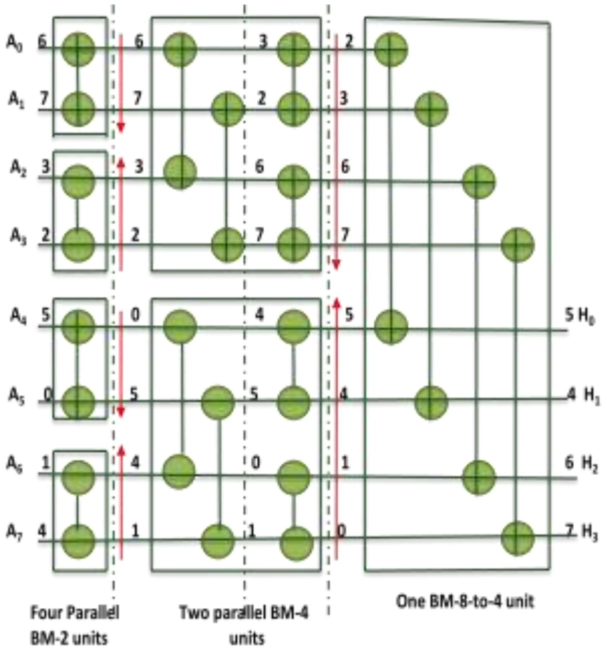


Fig 4An 8-to-4 bitonic max set selection unit

Fig 5 represents an 8-to-4 odd-even merge max set selection unit. The required number of CAE stages decreases from six in 8-input sorting unit to four in 8-to-4 max-set-selection unit.

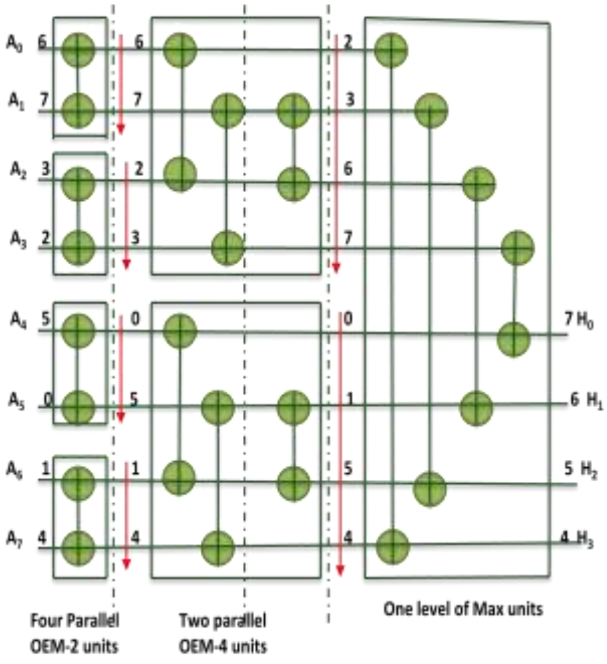


Fig 5An 8-to-4 odd-even merge max set selection unit

Fig 6 represents an 8-to-4 bitonic partial sorting unit. Here the bitonic max-set-selection unit output values are sorted in an increasing order. Fig 7 represents an 8-to-4 odd-even merge partial sorting unit where odd-even merge max-set-selection output values are sorted in an increasing order. If the number of inputs increases, then the number of CAE blocks and the latency is also increases as the latency is proportional to its depth (the number of consecutive CAE blocks).

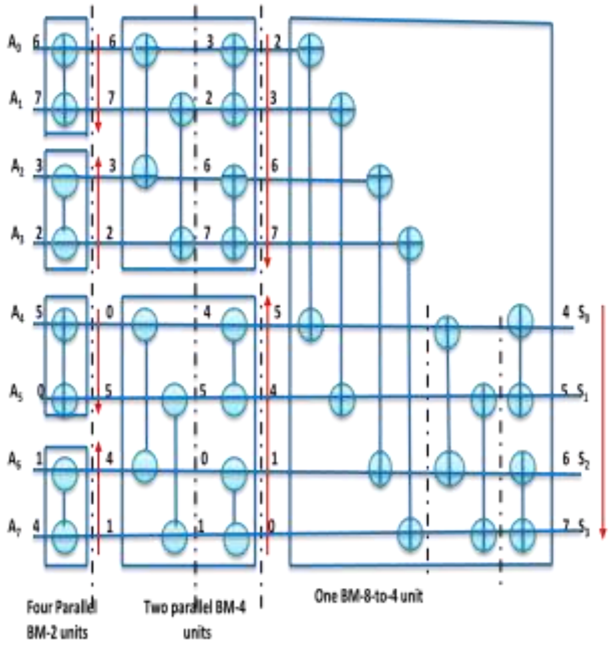


Fig 6An 8-to-4 bitonic partial sorting unit

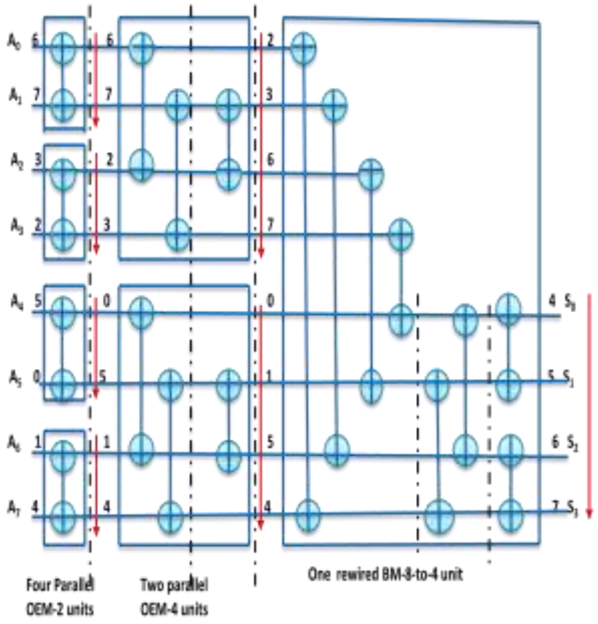


Fig 7An 8-to-4 odd-even merge partial sorting unit

IV. SORTING UNITS USING SCALABLE CMOS COMPARATOR

Comparators are the key design elements for wide range of applications. Even though the comparator logic design is straight forward, the extensive use of comparators in high-performance systems plays a major role on performance and area efficient optimizations. To obtain an area-efficient sorting network, the comparator present in the existing methodology [10] is replaced with the efficient scalable digital CMOS comparator using a novel parallel prefix tree [12]. This comparator consists of comparator resolution module connected to a decision module.These modules are structured as parallel prefix trees with repeated cells in the form of simple stages. Fig 8 shows a scalable digital CMOS comparator.

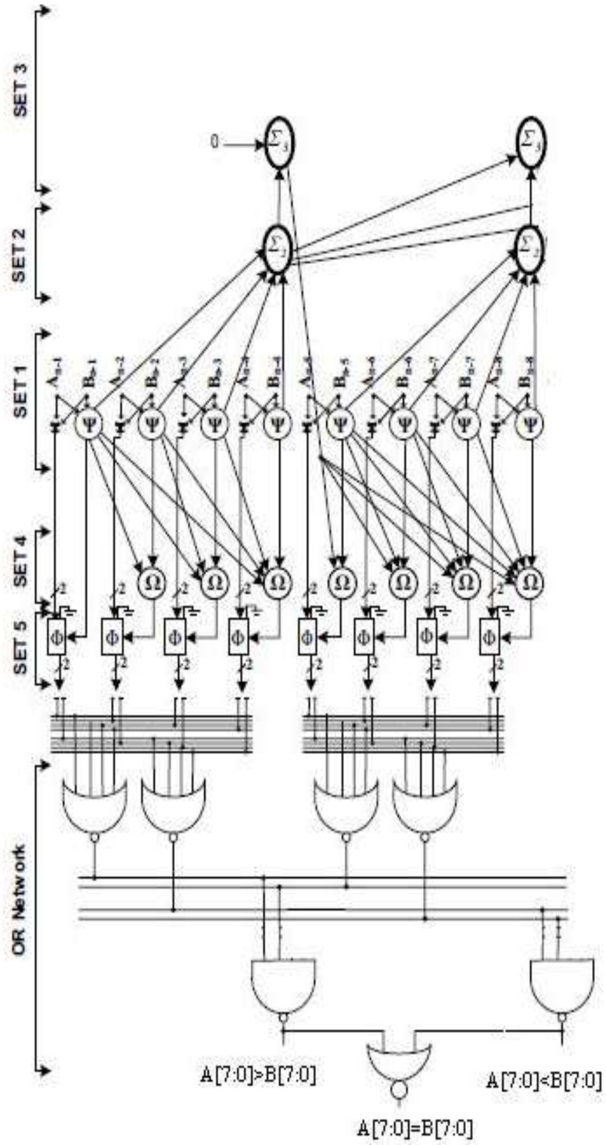
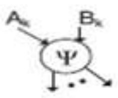
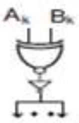
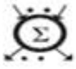
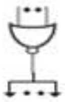
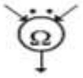
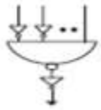
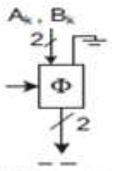
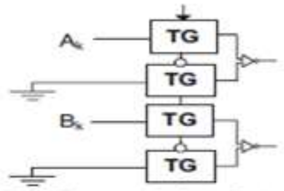


Fig 8Scalable digital CMOS comparator

The comparator resolution module is a novel prefix tree structure that performs bit-wise comparisons of two N-bit operands A and B. The decision module uses two OR-networks to output the final comparison decision.The comparator performs the comparison operation from the most significant bit to the least significant bit by using parallel operation. The comparator structure is partitioned into five hierarchical prefixing sets. Each set performs the specific function and each set's output serves as input to the next set in the hierarchy. All components within each set operate in parallel. The symbols used in the comparator represent certain functionality. TABLE I denotes the logic gate representation for symbols used in the comparator structure.

TABLE I

LOGIC GATE REPRESENTATION FOR SYMBOLS USED IN FIG 8

Symbols (Cells)	Logic Gate
	
	
	
 MUX-Logic	 TG: Transmission Gate

The inputs A0-A7 and B0-B7 are given in the set 1. This set compares 2 inputs bit-by-bit by using a single level of $N\Psi$-type cells. Ψ-type cells perform the Ex-or operation of two input bits.

Set 2 consists of Σ_2 type cells. These cells will combine the termination flags for each of the four Ψ-type cells from set-1 using NOR-logic.

Set 3 consists of Σ_3 type cells. This is similar to Σ_2 type cells, but can have more logic levels; different inputs and it carry different triggering points.

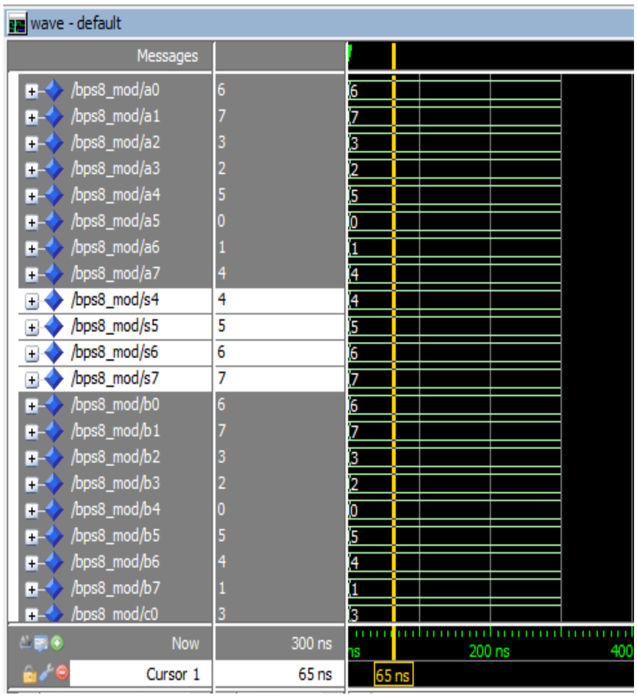
Set 4 consists of Ω type cells. These cells perform the NAND operation and outputs from Ω type cells control the select input of Φ type cells (two input multiplexers) in set 5.

Set 5 consists of Φ type cells (2-input, 2 bit wide multiplexers). One input to the cell is (A_k, B_k) and the other input is hardwired to "00". The select input comes from Ω type cell output in set 4. Using this modified scalable comparator Bitonic and odd-even merge sorting unit are constructed and their performances are analyzed.

V. SIMULATION AND FPGA IMPLEMENTATION RESULTS

Sorting units are coded in VHDL, simulated using Modelsim SE10.0b and area analysis is made by calculating the total equivalent gate count using Xilinx ISE.

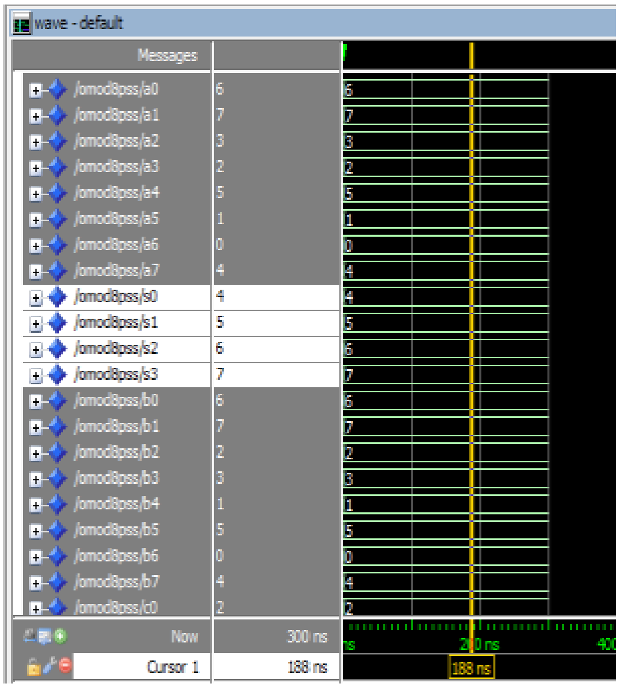
Fig 11 represents the simulation waveform of an 8-input Bitonic partial sorting unit using scalable CMOS comparator. It returns the 4 largest output values in sorted order.



Messages	
/bps8_mod/a0	6
/bps8_mod/a1	7
/bps8_mod/a2	3
/bps8_mod/a3	2
/bps8_mod/a4	5
/bps8_mod/a5	0
/bps8_mod/a6	1
/bps8_mod/a7	4
/bps8_mod/s4	4
/bps8_mod/s5	5
/bps8_mod/s6	6
/bps8_mod/s7	7
/bps8_mod/b0	6
/bps8_mod/b1	7
/bps8_mod/b2	3
/bps8_mod/b3	2
/bps8_mod/b4	0
/bps8_mod/b5	5
/bps8_mod/b6	4
/bps8_mod/b7	1
/bps8_mod/c0	3
Now	300 ns
Cursor 1	65 ns

Fig 11 Simulation result of an8-input Bitonic partial sorting unit using scalable CMOS comparator

Fig 12 denotes the simulation waveform of an 8-input odd-even merge partial sorting unit using scalable CMOS comparator.



Messages	
/omod8pss/a0	6
/omod8pss/a1	7
/omod8pss/a2	3
/omod8pss/a3	2
/omod8pss/a4	5
/omod8pss/a5	1
/omod8pss/a6	0
/omod8pss/a7	4
/omod8pss/s0	4
/omod8pss/s1	5
/omod8pss/s2	6
/omod8pss/s3	7
/omod8pss/b0	6
/omod8pss/b1	7
/omod8pss/b2	2
/omod8pss/b3	3
/omod8pss/b4	1
/omod8pss/b5	5
/omod8pss/b6	0
/omod8pss/b7	4
/omod8pss/c0	2
Now	300 ns
Cursor 1	188 ns

Fig 12 Simulation result of an 8-input odd-even merge partial sorting unit using scalable CMOS comparator

. The sorting unit using scalable CMOS comparators has less area when compared to the sorting unit using CAE block. TABLE II represents an area comparison of Bitonic and odd-even merge sorting architectures using CAE blocks and by using scalable CMOS comparator.

TABLE II
AREA COMPARISON OF DIFFERENT SORTING UNITS INTERMS OF GATE COUNT

Sorting Unit	**Using CAE Block**	**Using Scalable CMOS Comparator**
Bitonic sorting unit	4,032	3,696
Odd-even merge sorting unit	3,192	2,928
Bitonic max set selection unit	2,484	2,280
Odd-even merge max-set selection unit	2,148	1,968
Bitonic partial sorting unit	3,162	2,664
Odd-even merge partial sorting unit	2,826	2,580

Bitonic and odd-even merge sorting networks using scalable digital CMOS comparator consume less area than the sorting networks using CAE blocks. Fig 9 represents an area comparison chart of bitonic sorting units using CAE blocks and scalable digital CMOS comparator.

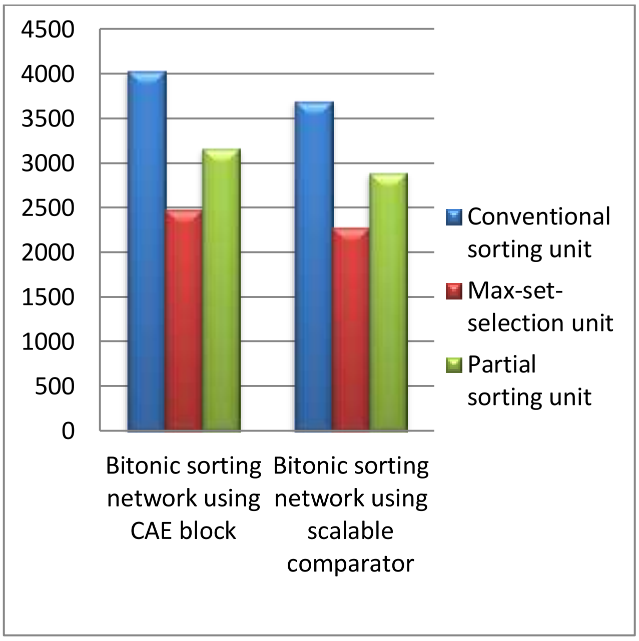


Fig 9 Area comparison of Bitonic sorting network using CAE blocks and by using scalable comparators (in terms of gate count)

Area comparison chart for odd-even merge sorting units using CAE blocks and scalable digital CMOS comparator is shown in Fig 10.

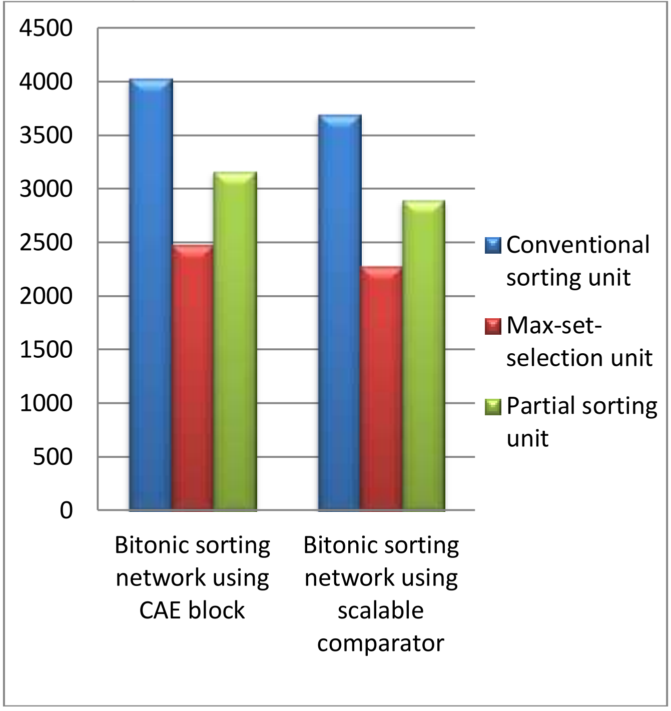


Fig 10 Area comparison of odd-even merge sorting network using CAE blocks and by using scalable comparators (in terms of gate count)

VI. CONCLUSION

This paper implements a flexible, high-throughput, area efficient max-set selection units and partial sorting units. The proposed sorting network using scalable digital CMOS comparator is found to be an area efficient sorting network compared to the conventional Bitonic and odd-even merge sorting networks using CAE blocks. The sorting units are coded in VHDL, simulated using Modelsim SE 10.0b and implemented in FPGA using Xilinx ISE for area analysis.

Future work will include the additional circuit optimizations to reduce the power dissipation and latency by replacing a decision module in the scalable comparator by the high-speed zero-detector circuit.

REFERENCES

[1] S. Azuma, T. Sakuma, T. Takeo, T. Ando, and K. Shirai, “Diaprism Hardware Sorter - Sort a Million Records within a Second”Available:http://sortbenchmark.org/Y2000_Datamation_DiaprismSorter.pdf, 2000.

[2] N. Govindaraju, J. Gray, R. Kumar, and D. Manocha, “GPUTeraSort: High Performance Graphics Co-Processor Sorting for Large Database Management” in *Proc. Conf. Management of Data*, 2000. pp. 325- 336.

[3] D. Pok, C.-I. Chen, J. Schamus, C. Montgomery, and J. Tsui, “Chip Design for Monobit Receiver,” *IEEE Trans. Microwave Theory and Techniques*, vol. 45, no. 12, pp. 2283-2295, Dec. 1997.

[4] Gregerson, M. Schulte, and K. Compton, *High-Energy Physics Handbook of Signal Processing Systems*, pp. 179-211, Springer, 2010.

[5] K.E. Batcher, “Sorting Networks and Their Applications,” *Proc.AFIPS Proc. Spring Joint Computer Conf.* pp. 307-314, 1968.

[6] J.-D. Lee and K.E. Batcher, “Minimizing Communication in theBitonic Sort,” *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 5, pp. 459-474, May 2000.

[7] E. Herruzo, G. Ruiz, J.I. Benavides, and O. Plata, “A New Parallel Sorting Algorithm Based on Odd-Even Mergesort,” *Proc. Int’l Conf. Parallel,distributed and Network-Based Processing*, pp. 18-22, 2007.

[8] C.J. Kuo and Z.W. Huang, “Modified Odd-Even Merge-Sort Network for Arbitrary Number of Inputs,” *Proc. IEEE Int’l Conf. Multimedia and Expo*, pp. 929-932, Aug. 2001.

[9] A. Farmahini-Farahani, A. Gregerson,M. Schulte, and K. Compton, “Modular High-Throughput and Low-Latency Sorting Units for FPGAs in the Large Hadron Collider,” *Proc. IEEE Int’l Symp. Application Specific Processors*, pp. 38-45, June 2011.

[10] A. Farmahini-Farahani, HenryJ.Duwe III, Michael J.Schuttle and Katherine Compton “Modular Design of High-throughput and Low-Latency sorting units,” *IEEE Trans on computers*, vol.62, no.7, July 2013.

[11] S.-N. Dong, X.-T. Wang and X.-B. Wang, “A Novel High-Speed Parallel Scheme for Data Sorting Algorithm Based on FPGA,” *Proc. Int’l Cong. Image and Signal Processing*, pp. 1-4, Oct. 2009.

[12] Abdel-Hafeez.S, Gordon-Ross.A,Parhami.B(2013) “Scalable Digital CMOS Comparator Using a Parallel Prefix Tree” *IEEE Transcations on Very Large Scale Integration (VLSI) Systems*, vol.21, no. 11, pp 1989-1998.