European Journal of Molecular & Clinical Medicine
ISSN 2515-8260 Volume 7, Issue 11, 2020

High Speed Parallel VLSI Architecture for 9 elements sorting in Image De-noising applications

Rajini Karanam[1] , Vasanth. K[1]

[1]Vidya Jyothi Institute of Technology, Aziz Nagar, Chilukur Road,
Hyderabad – 500075, India

Abstract. A Novel High Speed Parallel structure to Shapiro technique is proposed to XCV 600e-8fg676 target device. The main component in the parallel structure is binary comparator which offers less time delay. The Parallel structure of Shapiro technique is differentiated with 1D sorting and elevated in faster operation. The Proposed architecture was also compared with 9 element sorter that required 25 comparators of same class. It was found that the proposed architecture is faster to its counterparts. Hence High Speed VLSI architecture is proposed for arranging 9 elements in increasing order.

1. Introduction

Digital images are corrupted with salt and pepper noise during the digital dish TV transmission and reception. Median filters are traditionally used to remove the outliers. Computation of Median involves arranging elements in an increasing order. Every pixel is processed using Neighborhood operation involving a window of size 3x3, 5x5 or adaptive depending upon a particular logic. A need for high speed architecture is always required in modern time application that offers a less delay with enhanced capabilities. There are two types in which rank ordering are done to arrange data in ascending order. Bitwise approach and Byte wise approach are generally used for the above purpose. The basic processing element performs compare and swap operation in the two element comparator. A parallel architecture consists of horizontal lines which act as inputs lines and vertical lines are the comparators which perform the comparison operation. The processing element of a Parallel architecture is a compare and swap operation. Over the years many architecture had been formulated. The following gives the overview of work done in recent times. In this parallel architecture using 25 comparators for Codish sorting was proposed. By using generate and prune approach sorting was optimized. Codish sorting operates with 66.46 ns delay, with 7mw power by using 4200gate count and 402 slices. Sorting of 9 cell is done by direction comparison. The advantage of this architecture high speed less power and reduced area [1]. A decision based modified selection sort filter was proposed with sequential architecture is designed and simulated to remove salt and pepper noise. Several comparisons for different algorithm on done based on number of slices, look up tables, power, period, operating frequency, floor plan etc. The advantage of the sequential architecture is that it consumed low power, operates at high speed and required less area [2]. Vasanth et al introduced a sequential architecture to remove impulse noise, gaussian and mixed noise and remove noise in an image. A sequential VLSI architecture was proposed for the above algorithm. Snake-like shear sorting was implemented as parallel architecture is used to arrange the elements in ascending order. Two cases are used to detect noise in a pixel. Salt and pepper noise are corrected by using increasing the forward counter by 1 and decreasing the reverse count by 1. Unsymmetrical trimmed median filter is evaluated by using Finite state machine with data (FSMD). FPGA implementation showed the performance of algorithm in terms of area, speed and power [3]. New adaptive based median filter is implemented to remove impulse noise. In this algorithm, noise pixel is detected and modified. PSNR is calculated for different densities

of noise [4]. In this work a new adaptive partition cluster based median filter is introduced in which random value impulse noise is removed. The corrupted pixel is identified and filtered. It has an advantage of less complexity in computation [5]. Hoong & Ibrahim proposed different median filtering methods to remove impulse noise [6]. Zheng and Zang introduced a New MM-sorting is used with fixed size. It iteratively sorts the input data. Delete min, delete max are used to delete minimum and maximum number in an array. It has an advantage of reduced number of iterations and disadvantage is that it could not analyze the average running time of MM-sort [7]. Several sorting techniques were compared based on time complexity, space complexity. Different sorting algorithms are compared, their advantages, disadvantages, applications were explained [8]. Alnihoud & Mansi introduced a new enhanced selection sort (ESS), enhanced bubble sort (EBS). They compared different sorting with the proposed ESS and EBS sorting. The proposed sorting was faster with reduced number of swap operation. It had an advantage of reduced time complexity and operates faster compared with previous sorting [9]. Zhang and Karim introduced a new impulse detector technique for switching median filter using 4convolutions.It preserved line information in the image It requires 4N^2logN+1 compare/swap operations. The main advantage of the proposed method is that it used less compare/swap operation. The main disadvantage of the work is that it was difficult to find optimal threshold [10]. Codish et al proposed 25 comparators for sorting 9 inputs. In previous work 29 comparators were used to sort 10 inputs. Generate and prune approach and SAT encoding approach is used for optimization. The main advantage of the work needs reduced number of comparators [11]. Chakrabarti introduced Recursive and non-recursive filtering with reduced compute and swap operations. Comparators are reduced by fixing rank order that reduced latency [12]. Vasanth and Karthik introduced a new modified decomposition filter (MDF) which overcomes the drawbacks in threshold decomposition with simple decomposition approach. In MDF decomposition and regrouping is done to detect the corrupted pixel arrange the data to remove noise. Simple comparators, buffers and adders are used. It has an advantage of reduced area with a smaller number of slices and look up tables [13]. Vasanth et al proposed decision based unsymmetrical trimmed midpoint algorithm (DBUTMF) for noise detection and correction. The elements are compared and replaced by using forward counter F and reverse counter L. It is good for high noise densities effectively remove salt and pepper noise; edges are preserved. It has an advantage that the proposed algorithm is used for color image noise detection and correction [14]. Yester years have produced high speed architectures but still the speed to rank order the data is still demanding. High speed architecture is required for arranging elements in increasing order. Section II explains about the methodology of Shapiro Sorting. Section III explains about the proposed parallel architecture. Section IV gives the information regarding the simulation results. Section V concludes the work which has been.

2. Shapiro sorting

Over the years Codish, Shapiro, Floyd and Vasanth had proposed 25 comparator sorting using compare and swap approach. The Shapiro sorting is implemented as Mesh sorting with minimum of 25 comparators. The methodology of the sorting is given in figure 1. The main reason to rank order only 9 elements because these elements are used in image processing as neighborhood operation demanding 3x3 values of a confined neighborhood. The Shapiro sorting is a Mesh sorting that arranges 9 elements in increasing order with 11 different steps. The method in which sorting is done is given below. Consider an Image de-noising application using 3x3 Neighborhood. Extract the 3x3 pixels in a confined neighborhood and perform Shapiro sorting as shown in figure 1.

An example is taken and Shapiro sorting is explained below.

Step1: Arrange the first two elements in all rows in ascending order.

Step 2: Arrange the last two elements in all row in ascending order.
Step 3: Arrange all the row of first two elements in ascending order.
Step 4: Arrange first two elements in column 1 and column 3 in ascending order.
Step 5: Arrange last two elements in column 3 in ascending order.
Step 6: Arrange last two elements in column1 and first two elements in column 2, column 3 in ascending order.
Step 7: Arrange first two elements in column 1 and last two elements in column 2 in ascending order.
Step 8: Arrange the first two elements in column 2 and the sixth &eight elements in increasing order.
Step 9: Arrange the second &fourth and third &seventh elements in ascending order.
Step 10: Arrange the third & fifth elements and sixth & seventh elements in increasing order.
Step 11: Arrange the third & fourth elements and fifth & sixth elements in increasing order.
Lastly sorted list of 3*3 window is obtained which are arranged in ascending order.

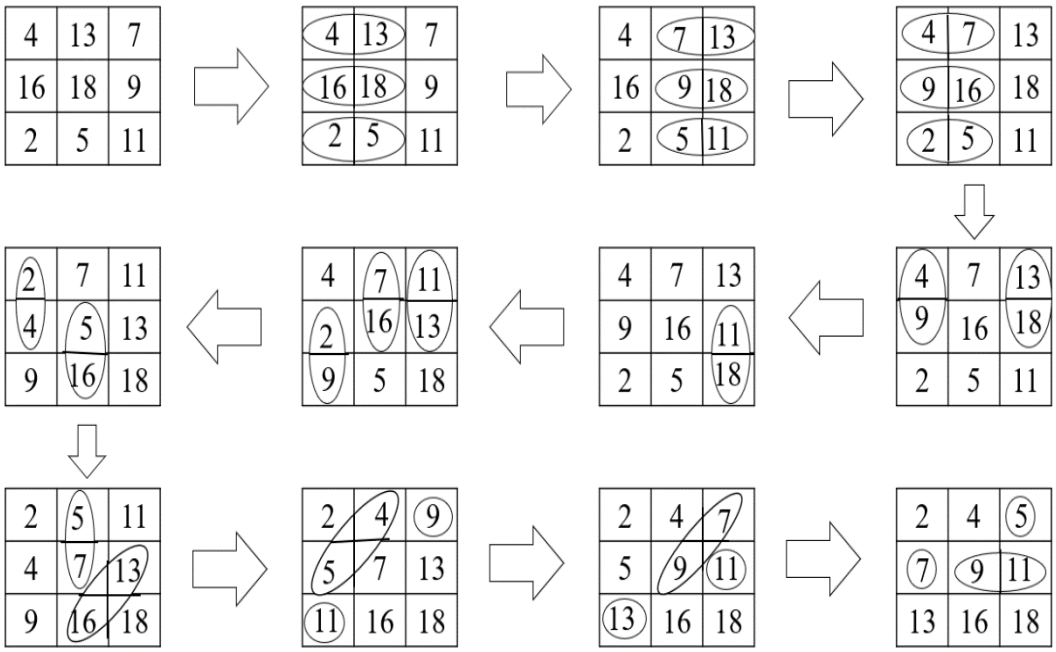

Figure.1 Methodology of proposed sorting using 3*3 window

3. Proposed parallel architecture

To enhance the speed of the Shapiro sorting a parallel architecture is built. The basic building block in the parallel design is a two cell comparator. The basic two cell comparator will compare two pixel elements and gives out low and high values. The Parallel architecture also classifies the data in rising form. It provides sorted data list at the output. Figure 2 gives the flow diagram of the Shapiro sorting. Parallel architecture has horizontal line which represents the input line and the vertical line represents comparator used for comparison. Comparator compares the given elements; it provides the minimum value and maximum value as shown in figure 3. The flow diagram shown in figure 3 has a two cell sorter which compares the value; it gives minimum and maximum value. The Parallel architecture consists of series of two cell architecture that compares two pixel elements in parallel and propagates the output to next stage. Every subsequent stage has two cell comparators that compare the propagated output. The flow diagram pumps the data into the parallel architecture and arranges data in increasing order finally. The level of the parallel architecture and depth of the architecture is 9 and 8 respectively. The sorting requires 25 comparators in order to classify 9 input data in

rising form. Figure 4 gives the block diagram representation of the proposed parallel architecture. Figure 5 is the two cell component that acts as main module in the parallel designing structure. The methodology of proposed parallel architecture is given below in different steps.

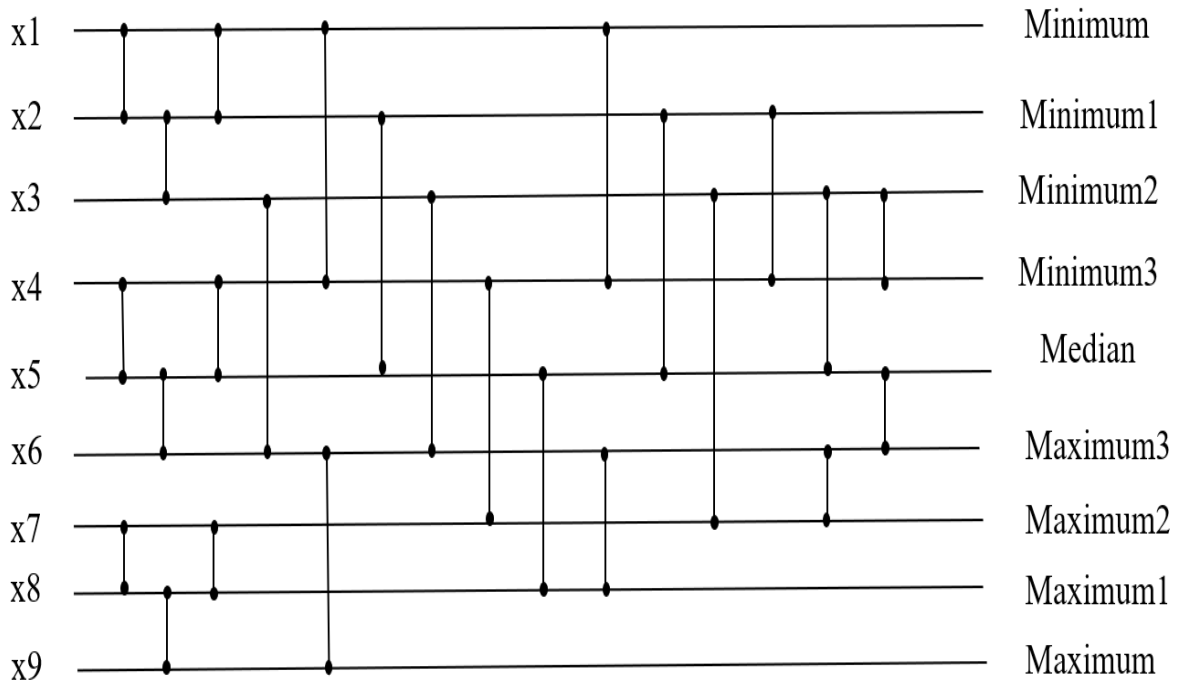

Figure 2 Flow diagram of proposed Parallel architecture

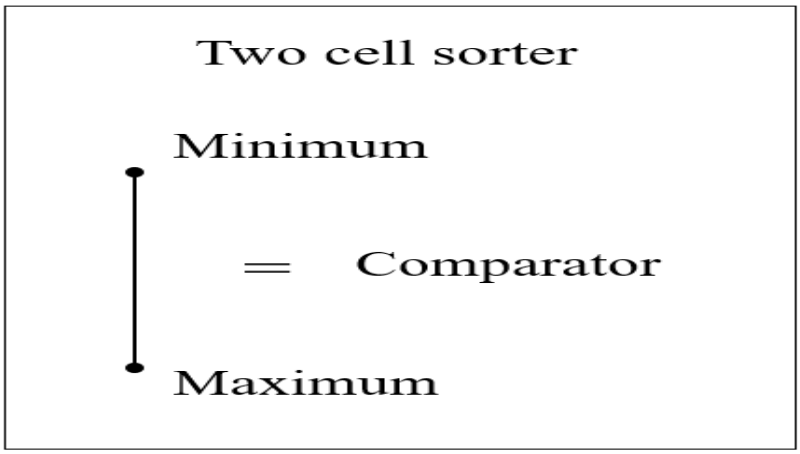

Figure 3 Flow diagram of two cell Comparator

The methodology of proposed parallel architecture is given below in different steps.

Step 1: First two cell comparator is given with inputs x1 and x2, it provides the output min1 and max1.

Step 2: Second two cell comparator is given with inputs x4 and x5, it provides the output min2 and max2.

Step 3: Third two cell comparator is given with inputs x7 and x8, it provides the output min3 and max3.

Step 4: Fourth two cell comparator is given with inputs max1 and x3, it provides the output min4 and max4. Step 5: Fifth two cell comparator is given with inputs max2 and x6, it provides the output min5 and max5.

Step 6: Sixth two cell comparator is given with inputs max3 and x9, it provides the output min6 and max6.
Step 7: Seventh two cell comparator is given with inputs min1and min4, it provides the output min7 and max7.
Step 8: Eight two cell comparator is given with inputs min2 and min5, it provides the output min8 and max8
Step 9: Ninth two cell comparator is given with inputs min3 and min6, it provides the output min9 and max9.
Step 10: Tenth two cell comparator is given with inputs max4 and max5, it provides the output min10 and max10.
Step 11: Eleven two cell comparator is given with inputs min7 and min8, it provides the output min11 and max11.
Step 12: Twelve two cell comparator is given with inputs max10 and max6, it provides the output min12 and maximum.
Step 13: Thirteen two cell comparator is given with inputs max7and max8, it provides the output min13 and max13.
Step 14: Fourteen two cell comparator is given with inputs min10 and min12, it provides the output min14 and max14.
Step 15: Fifteen two cell comparator is given with inputs max11 and min9, it provides the output min15 and max15.
Step 16: Sixteen two cell comparator is given with inputs max13and max9, it provides the output min16 and max16.
Step 17: Seventeen two cell comparator is given with inputs min 11 and min15, it provides the output minimum and max17.
Step 18: Eighteen two cell comparator is given with inputs max14 and max16, it provides the output min18 and maximum1.
Step 19: Nineteen two cell comparator is given with inputs min13 and min16, it provides the output min19 and max19.
Step 20: Twenty-two cell comparator is given with inputs min14 and max15, it provides the output min20 and max20.
Step 21: Twenty-one two cell comparator is given with inputs min19 and max17, it provides the output minimum1 and max21.
Step 22: Twenty-two two cell comparator is given with inputs min20 and max19, it provides the output min22 and max22.
Step 23: Twenty-three two cell comparator is given with inputs min18 and max20, it provides the output min23 and maximum2.
Step 24: Twenty-four two cell comparator is given with inputs min22 and max21, it provides the output minnimum2 and minimum3.
Step 25: Twenty-five two cell comparator is given with inputs max22 and min23, it provides the output median and maximum3.

3.1 Proposed Two cell Architecture

Two cell Comparator compares two 8 bit numbers and generates carry based on which minimum and maximum values are evaluated. The proposed comparator generates a carry by comparing two 8 bit data which is sent to each 1 bit comparator which is staged in the form of multiplexer. Based on the output carry respective values are selected from A or B from the multiplexer. The generated carry will propagate through two sets of eight 1 bit comparators that select minimum and maximum values respectively. The internal architecture diagram of the Data Comparator is shown in figure 6[15].

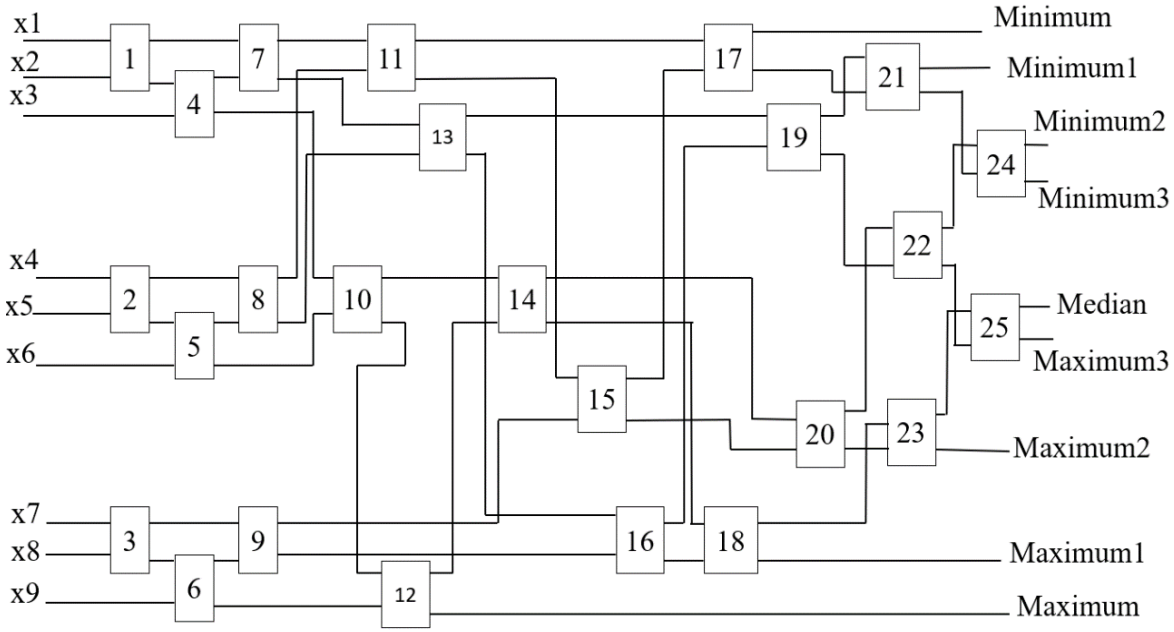

Figure 4 Block diagram of the proposed architecture for Shapiro sorting

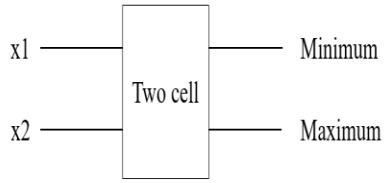

Figure 5 Architecture for Two cell sorter

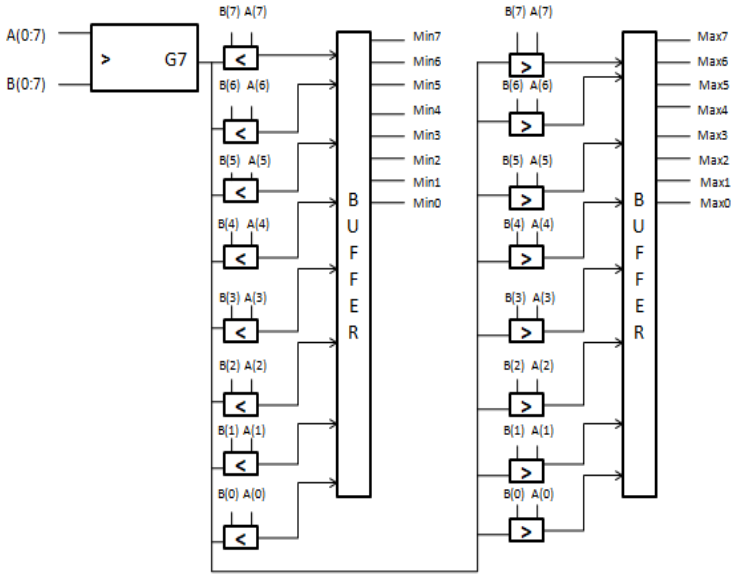

Figure 6 Block Diagram of Bitwise Data Comparator[15]

4. Simulation results

The simulations and synthesis were done in Modelsim 6.6 and Xilinx 7.1 targeted for the device XCV600e-8fg676. Table 1 gives the different two cell comparators targeted for XCV600e-8fg676. Table 2 gives the different sorting algorithms targeted for XCV1000-5bg560. Table 3 gives the comparison of different sorting techniques that requires 25 comparators to arrange 9 elements in increasing order. Initially few two cell comparators were developed and compared It for Virtex E hardware. The nomenclature of different two cell sorters are Borrow look ahead logic (BLAC), Binary to excess one circuit based comparator (2BEC), Decoder based data comparator (DDC), Multiplexer based Data comparators (MDC), Carry select logic data comparators (CSLA) and Binary Comparator (BC). It was found from table 1 that binary comparator operates at high speed with lesser delay of 10.45ns as we look for a high speed device as the aim of the work. Different types of comparators were realized using look ahead logic, carry select logic, multiplexer based and decoder based data comparators.

The proposed binary comparator used carry signal that acts as selection lines for different multiplexer based one bit comparator. This bitwise approach makes it faster in operation. The next level of simulation and synthesis is done by using the binary comparator as a basic processing element on Shapiro sorting. The algorithm was synthesized and compared with renowned algorithms that are targeted for the device XCV1000-5bg560. Several Renowned sorting algorithms developed by Vasanth et al along with Shapiro sorting. By the observation of the table 2 the propound parallel design of Shapiro method consumed less area of 406, operates very faster with less combinational delay of 63.78ns with gate count of 4200. The proposed parallel architecture also required only 7mw of power consumption.

Hence, propound parallel structure of Shapiro method operates faster when compared to its 1D and 2D counterparts. The Architecture required only 25 comparisons in organizing 9 input data in rising format. Next Parallel architecture of Shapiro was compared with other algorithms that require 25 comparisons for classifying 9 data elements in ascending form. The comparison is illustrated in table 3. The parallel designing of Shapiro method operates very faster when compared to Floyd, Codish and Vasanth sorting respectively. The Shapiro sorting offered 44.39ns of combinational delay. The level and depth of the propound method was less from the other algorithm. Figure 7 and 8 gives the floor plan and routed FPGA of the targeted device XCV 600e-8fg676. This makes the proposed algorithm performs better with a high speed architecture.

Table 1 Comparison of various two cell comparator for the target device XCV 600e-8fg676

PARAMETERS		BC (PA)	BLAC	2BEC	CSLA	DDC	MDC
SYNTHESIS REPORT	NO OF SLICES	13	14	17	14	52	14
	NO OF 4 I/P LUT	24	25	29	24	92	24
	BONDED IOB	32	32	33	32	32	32
MAP REPORT	NO OF 4 I/P LUT	24	25	29	24	92	24
	BONDED IOB	32	32	32	32	32	32
	GATE COUNT	168	150	302	144	641	144
	AVG FANOUT OF LUT	1	1.44	1.69	1.46	1.92	1.63
	MAX FAN OUT OF LUT	1	12	14	12	16	14

	AVG FAN IN FOR LUT	2.66	3.12	3.31	3.12	3.57	3.12
PLACE AND ROUTE REPORT	**EXTERNAL IOB**	32	32	32	32	32	32
	SLICES	12	13	15	13	15	13
	MAX COMBINATIONAL DELAY(ns)	10.45	15.397	18.643	15.397	18.303	15.87
	POWER CONSUMPTION(mw)	295	295	295	295	295	295

Table 2 Comparison of different sorting algorithms for the target device **XCV1000-5bg560**

PARAMETERS		**BITONIC**	**BUBBLE**	**HEAP**	**INSERTION**	**ODD-EVEN TRANSPOSITION**	**SELECTION**	**SNAKE1**	**26-CELL SNAKE**	**PROPOSED SHAPIRO**
SYNTHESIS REPORT	**8-BIT COMPARATORS**	80	-	-	-	-	-	42	33	25
	NO OF SLICES	1346	6713	6770	6423	946	7208	675	528	406
	SLICES FLIPFLOPS	-	9	9	9	-	9	-	-	-
	NO OF 4 I/P LUT	1920	9711	9620	9765	1344	10481	1008	792	600
	BONDED IOB	256	328	328	328	257	328	144	144	144
	MAX COMBINATIONAL DELAY in ns	81.362	337.765	613.266	483.533	80.231	742.394	113.469	104.894	63.78
MAP REPORT	**NO OF 4 I/P LUT**	1920	9730	9641	9819	1344	10504	1008	792	600
	NO OF BONDED IOB	256	328	328	328	256	328	144	144	144
	GATE COUNT	13440	72156	71526	70002	9408	77568	7056	5544	4200
	AVG FANOUT OF LUT	2.73	3.06	3.00	2.97	2.62	2.97	2.69	2.60	2.49
	MAX FANOUT OF LUT	6	93	96	98	6	95	6	6	6
	AVG FANIN FOR LUT	3.26	3.29	3.38	3.33	3.23	3.34	3.28	3.27	3.25
PLAC	**EXTERNAL**	256	328	328	328	256	328	144	144	144

E AND ROUTE REPORT	**IOB**									
	SLICES	1227	6150	6191	5836	857	6584	617	481	373
	POWER CONSUMPTION in mW	7	32	32	32	32	32	32	32	7

Table 3 Comparison of various sorting architectures for the target device XCV 600e-8fg676

PARAMETERS		**FLOYD ARCHITECTURE**	**CODISH ARCHITECTURE**	**VASANTH ARCHITECTURE**	**PROPOSED ARCHITECTURE**
SYNTHESIS REPORT	**8-BIT COMPARATORS**	25	25	25	25
	NO OF SLICES	406	411	406	406
	NO OF 4 I/P LUT	600	600	600	600
	BONDED IOB	144	144	144	144
	MAX COMBINATIONAL DELAY in ns	50.063	46.533	54.629	44.939
MAP REPORT	**NO OF 4 I/P LUT**	600	600	600	600
	NO OF BONDED IOB	144	144	144	144
	GATE COUNT	4200	4200	4200	4200
	AVG FANOUT OF LUT	2.49	2.52	2.49	2.49
	MAX FANOUT OF LUT	6	6	6	6
	AVG FANIN FOR LUT	3.25	3.22	3.25	3.2533
PLACE AND ROUTE REPORT	**EXTERNAL IOB**	144	144	144	144
	SLICES	369	373	370	368
	POWER CONSUMPTION in mw	295	295	295	295

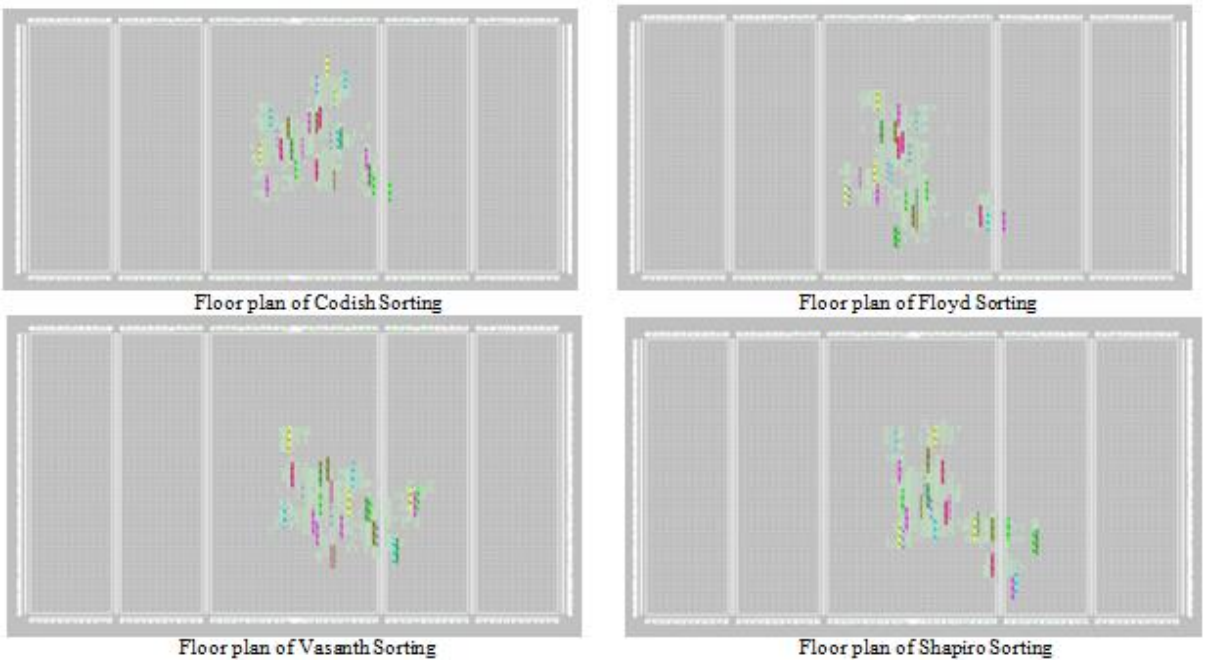

Figure 7 Floor plan of Different 9 cell sorters for the targeted FPGA XCV 600e-8fg676

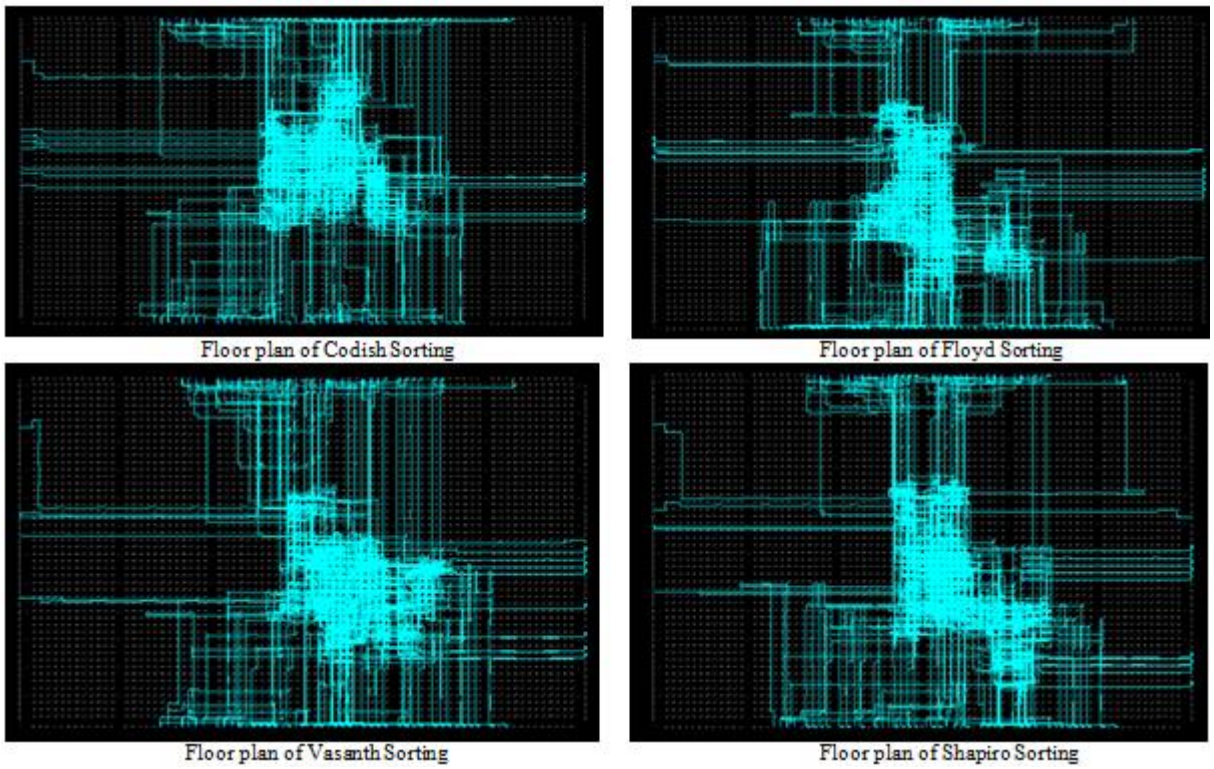

Figure 8 Routed FPGA of Different 9 cell sorters for the Targeted FPGA XCV 600e-8fg676

5. Conclusion

In this paper new parallel architecture is proposed for Shapiro sorting that consumes lesser delay is proposed. A binary comparator was formulated which acts as basic developing component in the propound parallel structure of Shapiro technique which is of high speed compared with the previous architectures of its class. The proposed parallel architecture requires only 44.39ns when compared with other 9 element sorters. The level and depth are less when compared to other sorting techniques and bit wise comparison make it suitable for

development of high speed architecture. By using the proposed architecture elements are arrange in increasing order with less combinational delay.

References

[1] Sindhu.E, K. Vasanth, "VLSI Architecture For 9 Element Optimized Sorting Network Using 25 comparator for Image De-Noising", International journal of innovative Technology and exploring engineering, 8, 6 1177-1183, (2019).

[2] Vasanth.K, "VLSI Architecture of Decision Based modified Selection Sort Filter For Salt and Pepper Noise Removal",International Journal on Intelligent electronics system, 7, 2,41-51, (2013).

[3] Vasanth.K, S. Karthik, "Performance Analysis of Unsymmetrical Trimmed Median As Detector On Image Noises& Its FPGA Implementation", International journal of information sciences and techniques, 2, 3,19-40, (2012).

[4] Suman Shrestha, "Image De-Noising Using New Adaptive Based Median Filter", Signal and image processing: An international journal, 5, 4,1-13, (2014).

[5] Ke Pang, Zaifeng Shi, Jiangtao Xu and Suying Yao, "Adaptive Partition-Cluster-Based Median Filter For Random-Valued Impulse Noise Removal", Journal of circuits, Systems and computers, 27, 7,1850110, 1-26, (2018).

[6] Sin Hoong Tech, Haidi Ibrahim, "Median Filtering Frameworks for Reducing Impulse Noise from Grayscale Digital Images", International Journal of Future computer and communication, 1, 4,323-326,(2012).

[7] Zheng S.Q, Yanjun Zhang, "An Efficient Parallel VLSI Sorting Architecture", VLSI Design, 11,2, 137-147, (2000).

[8] Purvi Prajapati, Nikita Bhatt, Nirav Bhatt, "Performance Comparison of Different Sorting Algorithms", International Journal of Latest Technology in engineering, Management and Applied Science, 6, 6, 39-41,(2017).

[9] Jehad Alnihoud, Rami Mansi, "An Enhancement of Major Sorting Algorithms", International Arab Journal of Information Technology, 7, 1, 55-62, (2010).

[10] Shuqun Zhang, Mohammed A. Karim, "A New Impulse Detector for Switching Median Filters", IEEE Signal processing letters, 9, 11, 360-363, (2002).

[11] Michael Codish, Micheal Frank, Lufs Cruz -Filipe, Peter Schneider- Kamp, "Twenty-Five Comparators Is Optimal When Sorting Nine Inputs (And Twenty-Nine for Ten)", IN: Proceedings in IEEE 26TH International conference on tools with artificial intelligents, 186-193, (2014).

[12] Chaitali Chakrabarti, "Sorting Network Based Architectures for Median Filters", IEEE Transactions on circuits and systems-II:Analog and digital signal processing, 40, 11, 723-727, (1993).

[13] Vasanth.K, S.Karthik, "FPGA Implementation of Modified Decomposition Filter", In: Proceedings in International conference on signal and image processing, 526-530, (2010).

[14] Vasanth K, V. Jawahar Senthil Kumar, "A Decision Based Unsymmetrical Trimmed Midpoint Algorithm for the Removal of High- Density Salt and Pepper Noise", Journal of theoretical and applied information technology, 2, 2, 245-251, (2012).

[15] Sindhu.E, K. Vasanth , "VLSI Architectures for 8 Bit Data Comparators for Rank Ordering Image Applications", 2019 International Conference on Communication and Signal Processing (ICCSP), 1-7, (2019).