Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume7-Issue1, January 2024
Paper: 56

Design and Implementation of high-speed Binary Counters using Sorting Networks and One-Hot Code Generation Technique

CH. Ganesh,
Assistant Professor
Electronics and Communication Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Hyderabad, India
ganesh_ch@vnrvjiet.in

S. Pallavi,
Undergraduate Student
Electronics and Communication Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Hyderabad, India
pallavireddysalla2622@gmail.com

T. Sravan Kumar,
Undergraduate Student
Electronics and Communication Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Hyderabad, India
sravanthotakuri76@gmail.com

G. Sai Preetham Reddy,
Undergraduate Student
Electronics and Communication Engineering
VNR Vignana Jyothi Institute of Engineering and Technology
Hyderabad, India
Preetham5015105@gmail.com

Abstract - This technical paper presents the design and optimization of fast binary counters using the Bitonic sorting network. The proposed (7,3) and (15,4) counters utilize the sorting network to construct a reordered input sequence. The (7,3) counter is optimized to outperform previous designs in terms of delay, area, and power-delay products, achieving improvements of 23.7%, 15%, and 40.7%, respectively. The (15,4) counter is designed to have a shorter delay and consume less area and power compared to previous designs. The implementation of these counters was carried out using Xilinx Vivado and Xilinx ISE Design Suite. The results demonstrate the effectiveness of using the Bitonic sorting network in designing high-performance binary counters.

Tool used -Xilinx vivado and Xilinx ISE design suite.

Keywords **- Sorting Network, Full Adder, FPGA, Bitonic Sorting, One hot code sequence**

I. INTRODUCTION

To add up all the incomplete products, a Wallace Tree structure is utilized, which is the efficiency bottleneck of the fundamental multiplier and the Wallace multiplier also uses half adders in the reduction phase [2]. The summation is commonly used by many DSP devices of multiple operands, and it is a vital route step. In some high-radix NTT versions, the central processor unit is made up of the sum of many operands. The Wallace tree structure, along with its modified technique, is the most well-known method of multiple operands. In these techniques, the summing is accelerated by using complete adders as (3,2) counters, leading to logarithmic time consumption [14]. A post-quantum cryptosystem called completely homomorphic encryption offers robust security for cloud computing that urgently requires number theoretic transform to speed up huge number multiplication and polynomial multiplication [16][17]. Due to the significant routing complexity, this straightforward solution is not practically practical for huge data volumes.

The key idea here is to create a counter that has a greater compression ratio than the (3,2) number by taking into account extra bits of the same weight. A very quick (6,3) counter with a symmetric stacking construction was suggested by Fritz and Fam [6], and using this counter as a base, they constructed a (7,3) saturated counter [8]. The delay performance of this design, although being the fastest in comparison to previous (7,3) counter designs, is inferior since no optimization was done before a MUX was just put to the critical route. We suggest this approach of immediately building a (7,3) counter to address the issue. The asymmetric layering structure, however we

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume7-Issue1, January 2024
Paper: 56

start with two asymmetric sorting networks that have three and four layers of sorting [1].

The sorting network is a powerful concurrent hardware network used for data processing. According to the well-known 0, 1 principle, If a sorting network can sort a collection of data whose members [18][19] are all 1-bit numbers, then it can sort any kind of number.

This paper is organized as follows. Proposed design of counter using conventional techniques using full adders and comparison with previous operation in Section II. With the performance analysis of proposed (7,3) and (15,4) counter, improvement of delay and power-delay product simulations are observed in Section III.

II. DESIGN OF COUNTER USING CONVENTIONAL TECHNIQUES

In this section counters design is explored using different techniques.

A. Design of counter (7,3) using a combination of full adders1

A half adder is a digital circuit that can provide the outcome of adding two 1-bit values. It has two input terminals where 1-bit numbers can be entered to be processed [3]. The half-adder then calculates the total of the numbers and, if applicable, the carry. Three inputs are added, and two outputs are generated by a full adder. A and B are the first two entries, and C-IN is the third. The output carry is indicated by the character C-OUT, whereas the standard output is indicated by the word S, which stands for SUM. The result of (7,3) counter by using full adder is shown in Fig. 8.

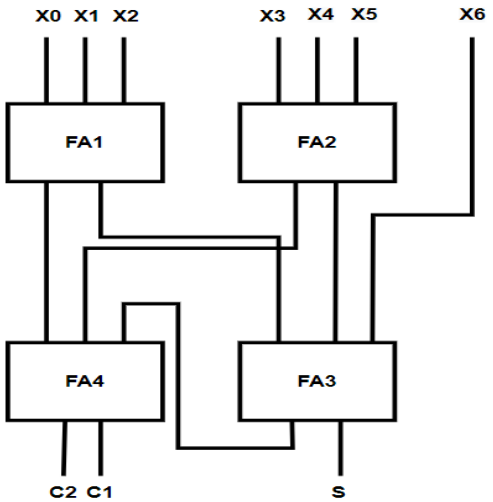


Fig. 1. Design of counter (7,3) using full adders

Half adders are used to create full adders, and full adders are used to build counters as shown in Fig. 1[1].

The construction of counter using full adders as shown in Fig. 2.is implemented by the equations (1) -(2) which comprises of S_2 , C_2 , S_1 , C_1 , S_3 , C_3 , S_4 , C_4.

$$S_2 = X_3 \oplus X_4 \oplus X_5 \quad \text{-(3)}$$

$$C_2 = X_3X_4 + X_4X_5 + X_5X_3 \quad \text{-(4)}$$

$$S_1 = X_0 \oplus X_1 \oplus X_2 \quad \text{-(5)}$$

$$C_1 = X_0X_1 + X_1X_2 + X_2X_0 \quad \text{-(6)}$$

$$S_3 = S_2 \oplus S_1 \oplus X_6$$

$$= X_3 \oplus X_4 \oplus X_4 \oplus X_5 \oplus X_5 \oplus X_3 \quad \text{-(7)}$$

$$C_3 = S_2S_1 + S_1X_6 + X_6S_2 \quad \text{-(8)}$$

$$S_4 = C_1 \oplus C_2 \oplus C_3 \quad \text{-(9)}$$

$$C_4 = C_1C_2 + C_2C_3 + C_3C_1 \quad \text{-(10)}$$

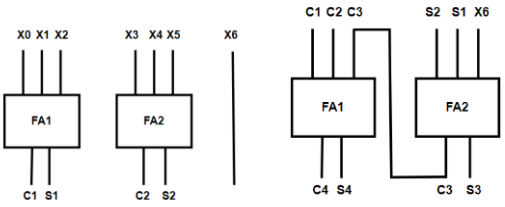


Fig. 2. Block diagram of counter using a combination full adder.

B. Design of counters using a Sorting Network (SN) and one-hot code

A potent parallel network is employed for data sifting called the sorting network. There are only 1-bit numbers, two data inputs, and and two outputs of data on each straight line, which stands for a sorter. The bigger input is always placed higher than the smaller input by the sorter [1]. A three-way sorting network's input is represented by the series [0, 1, 1], whereas a four-way sorting network's input is represented by [0, 1, 1]. After three layers of the sorter, the input sequences for both 4 SN and 3 SN are rearranged with lesser number at lower levels and a larger number at the summit.

By padding the reordered sequence with a fixed bit "1" at the top and a fixed bit "0" at the bottom, as

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume7-Issue1, January 2024
Paper: 56

illustrated in Fig. 3. we may regulate the sequence to guarantee that the 0,1-junction always exits. Second, the initial sequence's total number of "1"s and "0"s was preserved in the rearranged sequence [4]. The padded "1," which is set but would have an impact on the overall number of "1s" in the padded sequence, is not counted. For each layer that makes up a sorter, a simple two-input logical gate layer is used. Two inputs are reordered by the sorter based on numerical magnitudes. The logical circuit shown in Fig. 4 can sort two 1-bit data with ease. A sorter therefore employs a single stack of simple logic circuits with two inputs whereas three- and four-way sorting networks each need three layers of these gates. There are 3 actions. Then, split the 7 bits in half [7]. The two components each have 4 bits and 3 bits, respectively. After that, arrange the two pieces into two sized-appropriate sorting networks. The fixed "0" and the fixed "1" are added to the sorting networks' outputs. Second, using the Boolean expression, one-hot code sequences P0-P4 and Q0-Q3 are produced. One hot code sequence and boolean expressions between reordered sequences are formed [12], which has the same structure as (9)- (11). generation and simplification of the output expressions comes last.
The results of 2-bit sorting network are shown in Fig. 9. where 2 input bits interchange when MSB is 0 and LSB is 1 and the bits will be same when MSB is 1 and LSB is 0 based on the functionality of 2bit sorting. similarly, the 3-bit and 4-bit sorting network uses 2bit SN as shown in Fig. 10. And Fig. 11 With the help of 3bit and 4bit SN the (7,3) counter is built displays in Fig. 12 the output of counter. Fig. 13 displays the RTL schematic of 7,3 counter.

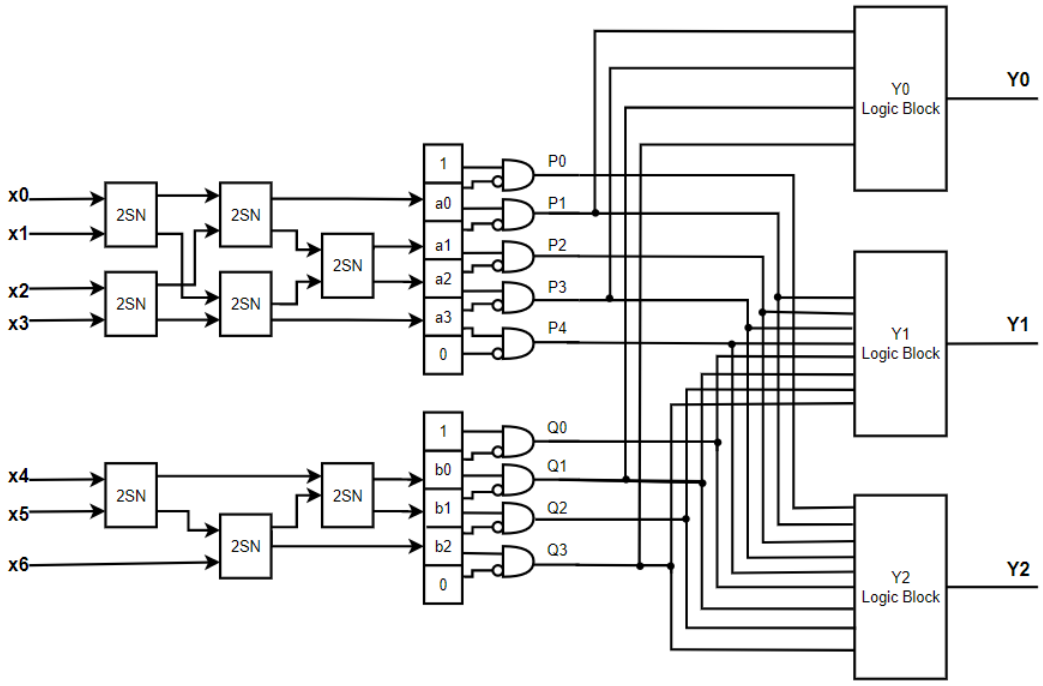


Fig. 3. Design of counters using a Sorting Network (SN) and one-hot code

$$Y_0 = (P_1 + P_3) \oplus (Q_1 + Q_3) \quad \text{-(11)}$$

$$Y_1 = Q_0(P_2 + P_3) + Q_1(P_1 + P_2) + Q_2\big(\sim(P_2 + P_3)\big) + Q_3(\sim(P_1 + P_2)) \quad \text{-(12)}$$

$$Y_2 = P_4 + \big(P_3.(\sim Q_0)\big) + P_2(Q_2 + Q_3) + (P_1.Q_3) \quad \text{-(13)}$$

Both comparators and wires are components of a sorting network. The wires are imagined to be oriented from left to right, carrying values (one per wire) that move simultaneously across the network [5]. Two inputs are reordered by the sorter based on their

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume7-Issue1, January 2024
Paper: 56

numerical magnitudes. This suggests that a sorter use a single layer of elementary logic gates with two inputs for two 1-bit data.

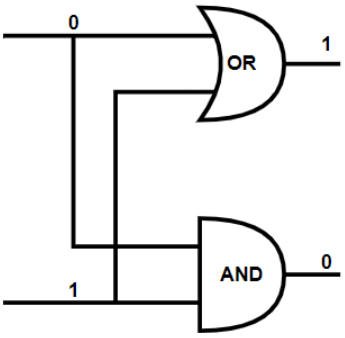


Fig. 4. Design of 2-bit sorting network

III. DESIGN OF COUNTER USNG PROPOSED ALGORITHM

A special kind of sequence known as a bitonic sequence is used in the bitonic sort, a comparison-based technique of arranging bits in an array or sequence[9]. The element sequence in a bitonic array is one in which the elements increase up to a certain element and then decrease in value until the end. Many parallel sorting algorithms are built on bitonic sorting or odd-even sort because of their inherent parallelism and fast time complexity[10]. Bitonic Sort can only be done if the number of elements to sort is 2^n. The bitonic sequence approach fails if the number of components does not precisely fit inside the range [15]. As we did previously, the sorting is carried out in three steps. The first step involves creating the bitonic sequence from the supplied random sequence. It is regarded as the start of the procedure. After finishing this stage, The components of the initial portion of the sequence will be arranged in increasing order, while the second half's elements will be arranged in descending order as shown in Fig. 5. The results of 15,4 counter using bitonic sequence is shown in Fig. 16. With the help of 7-bit and 8-bit sorting network the 15,4 counter is built the 7,8-bit network is shown in Fig. 14. And Fig. 15.

In order to compare the first element of the initial first half with the first element from the second half, the second element from the first half, we must now do a comparison.

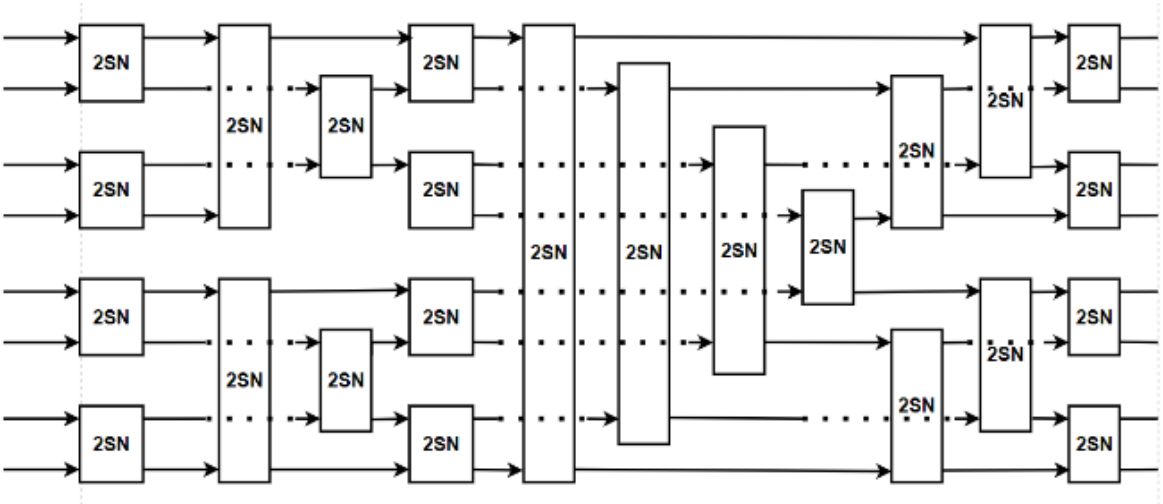


Fig. 5. 8-Bit Sorting Network

Beginning with the second component from the second half, and so on. We must swap the components if it comes out that any element in the second half is smaller. The RTL schematic of bitonic sorting is shown in Fig. 6**.** Because of the preceding stage, every element in the first half was smaller than every element in the second half [13]. The two n/2-length sequences are created because of the compare and swap operation. We iteratively apply the operations of the second phase to every sequence up until we have a singular sorted sequence of length n.

The input bits are applied immediately to the bitonic sorting network, which then completes the process of creating one hot code, as opposed to being divided and sorted from upper level to lower level.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume7-Issue1, January 2024
Paper: 56

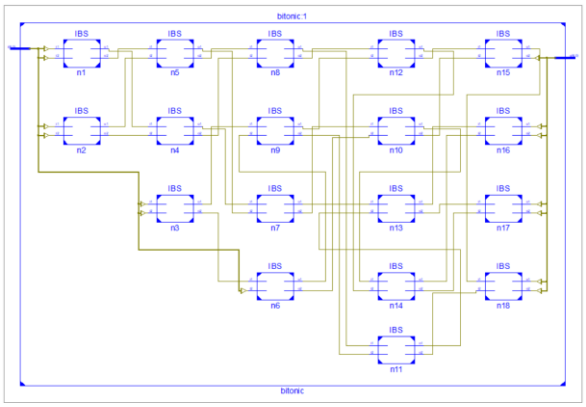


Fig. 6. RTL Schematic of bitonic sorting

By appending the low logic(0) bit to the least significant bit(LSB) and high logic(1) to the most significant bit(MSB) and finding the zero-one junction (0,1) [11] gives base value of the digit which results the output of binary counter when it is converted to the binary form either it may be (7,3) or (15,4) counter.

A. Design of counters implementation on FPGA

Counters are frequently regarded as crucial building blocks for many circuit operations, including configurable frequency divisions, shifters, code generators, memory pick management, and various math operations. Since many applications consist of these basic processes, a lot of study is devoted to the development of effective counter architectures. Design methods for counter architecture examine trade-offsbetween working frequency, power usage, space needs, and target application specialty. The 7,3 counter isimplemented on FPGA Xilinx artix7 is shown in Fig. 7 by assigning the input and output ports with respect to the FPGA I/O pins. Counters are initially built as standard hardware-based components. The counters are created as chunks of built-in memory for FPGAs that share a single processing unit.

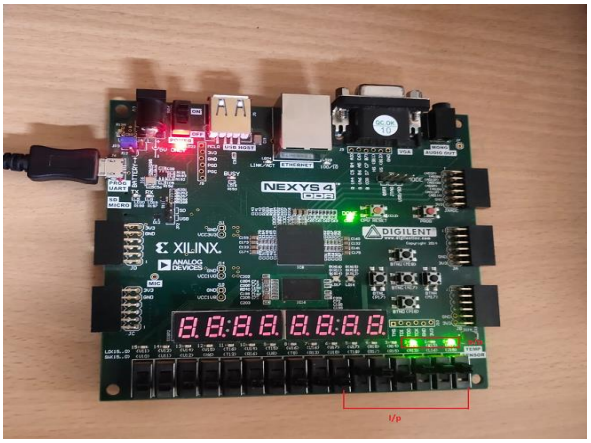


Fig. 7. (7,3) counter on FPGA

Input pins: U18, T18, R17, R15, M13, L16 and J15
output pins: J13, K15 and M17

IV. RESULTS AND COMPARISONS

A. Simulation of counter using a combination of a full adder

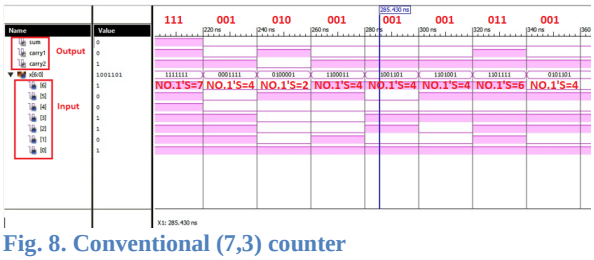


Fig. 8. Conventional (7,3) counter

The conventional (7,3) counter is built by using full adder logic where the 7 inputs is converted into 3 outputs as s, c1, c2

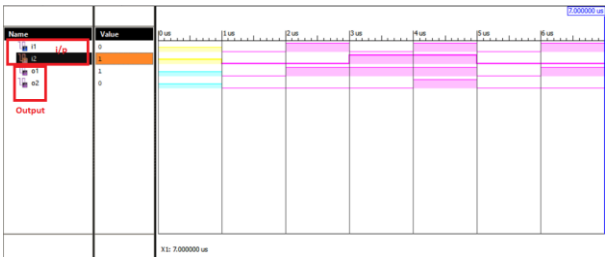


Fig. 9. 2-Bit Sorting Network

In 2-Bit SN where 2 input bits interchange when MSB is 0 and LSB is 1 and the bits will be same when MSB is 1 and LSB is 0 based on the functionality of 2bit sorting

B. Simulation of counter using Sorting Network (SN) and one-hot code

2-Bit Sorting Network

3-Bit Sorting Network

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume7-Issue1, January 2024
Paper: 56

Fig. 10. 3-Bit Sorting Network

The 3-Bit SN consists of 3 inputs and 3 outputs where the bits are sorted based on the functionality of MSB and LSB of 1 and 0

4-Bit Sorting Network

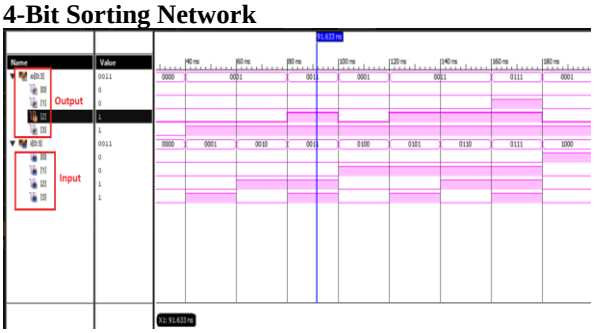


Fig. 11. 4-Bit Sorting Network

The counter is designed by using sorting network logic and one hot code generation of 3 bit and 4 bit SN. Here the 7 inputs are divided into 3 bit SN and 4 bit SN.

RTL Schematic

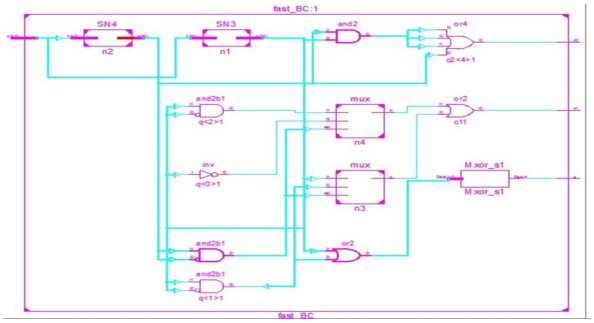
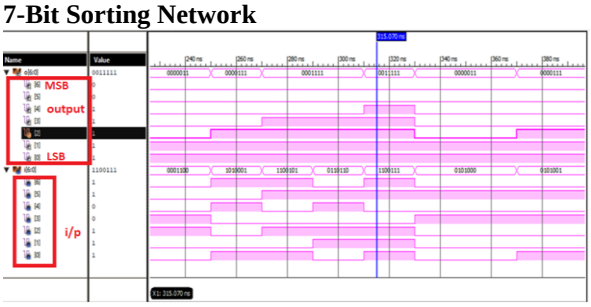


Fig. 13. Schematic RTL view of 7, 3 counter

This is the RTL Schematic view of a (7,3) counter designed with the help of one hot code generation and sorting network

7-Bit Sorting Network



The 4-Bit SN consists of 4 inputs and 4 outputs where the bits are sorted based on the functionality of MSB and LSB of 1 and 0

(7,3) Counter

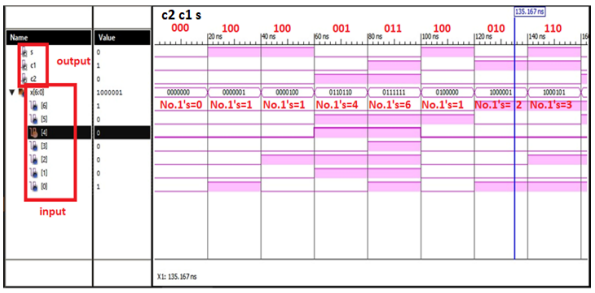


Fig. 12. (7, 3) counter

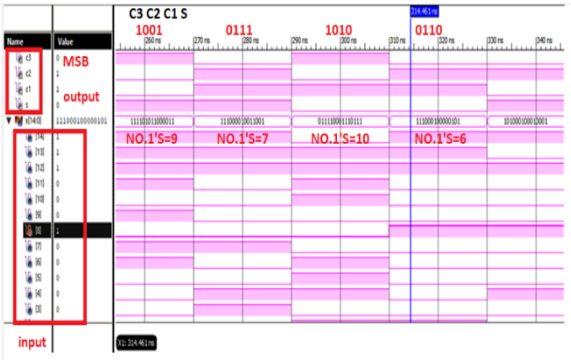


Fig. 14. 7-Bit Bitonic Sorting Network

The 7-bit SN is done by the proposed algorithm of sorting network having of 7 inputs and outputs

8-Bit Sorting Network

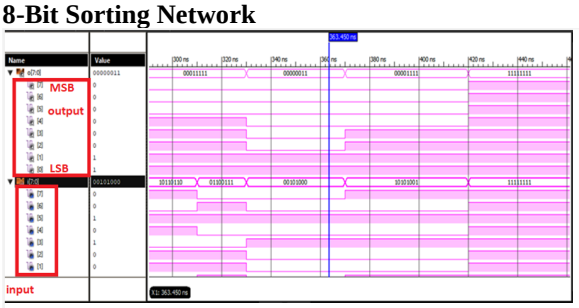


Fig. 15. 8-Bit Bitonic Sorting Network

In 8 bit bitonic sequence the elements are first arranged in increasing order, and then after some particular index, they start decreasing.

(15,4) Counter

Fig. 16. (15,4) Counter

The 15,4 counter is designed by the 7 bit and 8 bit bitonic sorting network where the 15 input bits is converted into 4 output bits of s, c1, c2, c3.

Journal of Current Research in Engineering and Science
Bi-Annual Online Journal (ISSN : 2581 - 611X)

Volume7-Issue1, January 2024
Paper: 56

Comparision Tables:

Here are the comparision results of the proposed model to the reference models [1] where they used 2 way sorting network and symmetric stacking method [6].

(7,3) counter

Method	Delay (ns)	LUT'S	Power (uW)	PDP (ns*uW)
Proposed	0.22	5	28.0	6.16
[1]	0.38	6	27.6	10.5
[6]	0.40	6	34.1	13.6

(15,4) counter

Method	Delay (ns)	LUT'S	Power (uW)	PDP (ns*uW)
Proposed	0.57	32	48.2	27.47
[1]	0.75	44	46.7	35.3
[6]	0.63	38	120.3	76.0

V. CONCLUSION

In this technical paper, we have presented a novel approach for counter design using bitonic sorting and one hot code generation techniques. Our proposed design exhibits 23.7% and 41.3% less delay than the existing models proposed in [1] and [6], respectively. Furthermore, the area (in terms of LUT's) and power consumption are reduced by 15% and 28.2%, and 40.7% and 32.6% compared to the models proposed in [1] and [6], respectively. By reducing both delay and power consumption, our proposed counter design achieves a lower power delay product, making it suitable for various applications that require lower delay, area, and power consumption. Overall, our results demonstrate the effectiveness and efficiency of our proposed approach for counter design, which can offer significant benefits over existing techniques.

REFERENCES

[1] W. Guo and S. Li, "Fast Binary Counters and Compressors Generated by Sorting Network," IEEE Trans. Very Large Scale Integr. Syst., vol. 29, no. 6, pp. 1220–1230, 2021, doi: 10.1109/TVLSI.2021.3067010.

[2] P. E. Stepansky, "Brief Contributions," Freud, V.1, vol. 59, no. 8, pp. 217–280, 2020, doi: 10.4324/9781315791937-9.

[3] J. Ding, S. Li, and Z. Gu, "High-Speed ECC Processor over NIST prime fields applied with toom-cook multiplication," IEEE Trans. Circuits Syst. I Regul. Pap., vol. 66, no. 3, pp. 1003–1016, 2019, doi: 10.1109/TCSI.2018.2878598.

[4] O. Akbari, M. Kamal, A. Afzali-Kusha, and M. Pedram, "Dual-Quality 4:2 Compressors for Utilizing in Dynamic Accuracy Configurable Multipliers," IEEE Trans. Very Large Scale Integr. Syst., vol. 25, no. 4, pp. 1352–1361, 2017, doi: 10.1109/TVLSI.2016.2643003.

[5] R. Chen and V. K. Prasanna, "Computer Generation of High Throughput and Memory Efficient Sorting Designs on FPGA," IEEE Trans. Parallel Distrib. Syst., vol. 28, no. 11, pp. 3100–3113, 2017, doi: 10.1109/TPDS.2017.2705128.

[6] C. Fritz and A. T. Fam, "Fast Binary Counters Based on Symmetric Stacking," IEEE Trans. Very Large Scale Integr. Syst., vol. 25, no. 10, pp. 2971–2975, 2017, doi: 10.1109/TVLSI.2017.2723475.

[7] A. G. M. Strollo, E. Napoli, D. De Caro, N. Petra, and G. DI Meo, "Comparison and Extension of Approximate 4-2 Compressors for Low-Power Approximate Multipliers," IEEE Trans. Circuits Syst. I Regul. Pap., vol. 67, no. 9, pp. 3021–3034, 2020, doi: 0.1109/TCSI.2020.2988353.

[8] M. Mehta, V. Parmar and E. Swartzlander, "High-speed multiplier design using multi-input counter and compressor circuits," [1991] Proceedings 10th IEEE Symposium on Computer Arithmetic, Grenoble, France, 1991, pp. 43-50,doi: 10.1109/ARITH.1991.145532.

[9] Q. Jiang and S. Li, "A design of manually optimized (15, 4) parallel counter," 2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC), Hsinchu, Taiwan, 2017, pp. 1-2, doi: 10.1109/EDSSC.2017.8126527.

[10] Jae-Dong Lee and K. E. Batcher, "Minimizing communication in the bitonic sort," in IEEE Transactions on

Parallel and Distributed Systems, vol. 11, no. 5, pp. 459-474, May 2000, doi: 10.1109/71.852399.

[11] T. Nakatani, S. . -T. Huang, B. W. Arden and S. K. Tripathi, "K-way bitonic sort," in IEEE Transactions on Computers, vol. 38, no. 2, pp. 283-288, Feb. 1989, doi: 10.1109/12.16506.

[12] Tsern-Huei Lee and Jin-Jye Chou, "Some topological properties of bitonic sorters," in IEEE Transactions on Computers, vol. 47, no. 9, pp. 983-997, Sept. 1998, doi: 10.1109/12.713317.

[13] Bonuccelli, Lodi and Pagli, "External Sorting in VLSI," in IEEE Transactions on Computers, vol. C-33, no. 10, pp. 931-934, Oct. 1984, doi: 10.1109/TC.1984.1676356.

[14] J. Xu, S. Kim, B. Nikolic and Y. S. Shao, "Memory-Efficient Hardware Performance Counters with Approximate-Counting Algorithms," 2021 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Stony Brook, NY, USA, 2021, pp. 226-228, doi: 10.1109/ISPASS51385.2021.00041.

[15] G. Bilardi and F. P. Preparata, "An Architecture for Bitonic Sorting with Optimal VLSI Performnance," in IEEE Transactions on Computers, vol. C-33, no. 7, pp. 646-651, July 1984, doi: 10.1109/TC.1984.5009338.

[16] M. B. Swapnika and M. ch. Ganesh, “Design and Implementation of Dual Mode Compressor Based32 Bitdadda Multiplier using Modified Carry Select Adder,” Int. J. Innov. Technol. Explor. Eng., vol. 2, no. 9, pp. 1763–1767, 2019, doi: 10.35940/ijitee.b7893.129219.

[17] S. V. V. Kumar, “ISSN No : 2348-4845 Noval Architecture of UTMI Module in USB,” vol. 3, no. April, pp. 19–24, 2016.

[18] Y. B. Prasad, G. Chokkakula, P. S. Reddy, and N. R. Samhitha, “Design of low power and high speed modified carry select adder for 16 bit Vedic Multiplier,” 2014 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2014, no. 978, 2015, doi: 10.1109/ICICES.2014.7034180.

[19] N. S. Reddy, G. Chokkakula, B. Devendra and K. Sivasankaran, "ASIC implementation of high speed pipelined DDR SDRAM controller," International Conference on Information Communication and Embedded Systems (ICICES2014), Chennai, India, 2014, pp. 1-5, doi: 10.1109/ICICES.2014.7033980.