Indian Journal of Natural Sciences *www.tnsroindia.org.in ©IJONS*

Vol.14 / Issue 78 / June / 2023 *International Bimonthly (Print) – Open Access* *ISSN: 0976 – 0997*

RESEARCH ARTICLE

VLSI Implementation of Sobel Edge Detector using Mode Selection-Based Hybrid Reconfigurable Adders

Telugu Maddileti[1], P.Manmohan Shashank Varma[2]* and P.Nandha Kumar

[1]Associate Professor, Department of ECE, Malla Reddy Engineering College, Secunderabad, Hyderabad, Telangana 500100, India.
[2]PG Scholar, Department of ECE, Malla Reddy Engineering College, Secunderabad, Hyderabad, Telangana 500100, India.
[3]Assistant Professor, Department of ECE, Malla Reddy Engineering College, Secunderabad, Hyderabad, Telangana 500100, India.

Received: 20 Feb 2023 Revised: 20 Apr 2023 Accepted: 30 May 2023

***Address for Correspondence**
P.Manmohan Shashank Varma
PG Scholar, Department of ECE,
Malla Reddy Engineering College,
Secunderabad, Hyderabad,
Telangana 500100, India.
Email:shashankvarma@gmail.com

ABSTRACT

The Sobel Edge Detector is essential to the operation of a great deal of software for digital image processing, such as programmes for pattern identification and area detection, amongst others. The hardware implementation for sobel edge detection is the essential to meet the specifications of these applications. However, the conventional methods are suffering with improper edge estimations with higher hardware resource utilizations. Therefore, this work is focused on implementation of hybrid sobel edge detector (HSED) using half sum-carry generation square root carry select adders (HSCG-SQRT-CSLA). In practice, the HSED process is carried out by conducting a matrix multiplication between the pixel values of the picture and the HSED. Further, this matrix multiplication operation is performed through successive HSCG-SQRT-CSLA additions. In addition, subtraction in matrix multiplication process is achieved by twos complement addition through HSCG-SQRT-CSLA. The simulations demonstrated that the suggested HSED resulted in improved performance when compared to the more traditional edge detection approaches.

Keywords: Hybrid sobel edge detector, half sum-carry generation, square root carry select adders, twos complement addition.

INTRODUCTION

The design of edge detector-based VLSI circuits needs to be improved for the development of VLSI technology [1]. In order to improve the controllability and observability methods of edge detector, partitioning and random test pattern are necessary. The edge detector methods should cover 90% of perfect edge coverage [2]. The edge detector-based VLSI circuits with more than 20000 gates can be implemented in simulation. The cost of perfect edge simulation is also to be reduced. Certain guidelines need to be followed in designing of VLSI circuits to make the edge detector-based VLSI simple and perfect edge free [3]. The factors such as test strategy, initialization, synchronous system, test mode logic, wired logic, floating signals, one shot, clock control, power on reset and analog modules are to be considered for the best design of VLSI circuits [4]. The test strategy is to be identified at the time of designing itself. Proper test strategy is not determined at the beginning it will be too difficult to identify the perfect edge at later stage. In order to test the VLSI circuit, it is to be driven from a known state [5]. When a reset signal is received, all of the digital circuits, including flip-flops, counters, latches, registers, and so on, must have their starting values fed into them in order to function properly as edge detectors [6]. These initial values are to be loaded by a separate driven circuit. It reset signal is given the driven circuit will load the initial values. This initialization is to be done before test vectors is passed to the circuit under test. All the components in the VLSI circuit are to be synchronized to avoid race condition [7]. The clock and reset signals are the two types of signals specifically designed for proper synchronization of VLSI circuits. Certain hardware is specifically designed in the chip for edge detector purpose [8]. This hardware is separated from the normal functions of the chip. In the test mode logic various components of chip is detected. The VLSI input output pins are multiplexed for normal operation and edge detector mode operation. The number of pins used for this purpose is maintained as minimum as possible. The wired AND, wired OR logic sometimes creates problems during edge detector. This can be avoided by using AND/OR logic or OR/AND logic [9]. The floating signals need to be monitored carefully, otherwise it can store charge in the temporary capacitor which act as a dynamic RAM. This memory will be refreshed and make some changes in data stored in the static memory. In order to get good reliability, the floating signals are to be avoided. One stable multivibrator is unstable during edge detector [10]. The use of one stable multivibrator is to be avoided, instead counters, logic gates and time generating circuits. The one-shot circuits will not be compatible with synchronous design circuits. As a result, the most significant contributions that this study has made are as follows:

- Execution of HSED using matrix multiplication between image pixel values to the HSED through HSCG-SQRT-CSLA.
- Construction of novel HSEDs and matrix multiplication operation is performed through successive HSCG-SQRT-CSLA additions.
- Subtraction in matrix multiplication process is achieved by twos complement addition through HSCG-SQRT-CSLA.

The remaining parts of the article are structured as follows: The essay is concluded with some potential future directions in section 5, after which sections 2 and 3 deals with the literature review, section 3 with the suggested implementation of HSED, section 4 with the analysis of data with performance comparison, and section 2 with the conclusion.

Literature survey

In [11] authors used the two-point crossover for the edge generation. The crossover points are created randomly. There is a 50% possibility of each chromosome in the chromosome pair to appear in the new edge. The new chromosome pair inherits the properties of parent chromosomes. The same procedure is repeated till the whole population of chromosomes pair have been created. In [12] authors have introduced mutation operator to create diversity in the edge. Adaptive mutation is followed to create new papulation. The mutation probability rate considered for this test pattern generation is 1% and this may be increased to higher values to generate a greater number of populations if there is no perfect edge coverage in three successive generation of genetic algorithm with respect to Global Record Table [13]. The delay test system uses mutation property. External clock control can be used by the VLSI circuit for edge detector purpose by disabling the input clock generator. The use of external clock signal

avoids race condition. The race condition normally takes place in master slave flip flop [14]. If the internal clock generator is not manageable during edge detector, then the need of external clock generator arises. The power on reset is an option available in VLSI circuits for function during the start of the particular applications [15]. But during edge detector the power on reset option is not necessary and it should be bypassed. The edge detector for power on reset is also necessary.

Additional circuits for bypassing the power on reset hardware during edge detector should be included in the design stage itself [16]. Analog circuits such as amplifiers, analog to digital converter, digital to analog converter comparators etc. may be presented in the VLSI chip. A Separate control for analog circuit is necessary for edge detector; otherwise, it will create short circuit or race condition in the circuit. The isolation of analog and digital circuit is necessary for the proper implementation, design and edge detector-based VLSI circuits [17].The fitness function and the perfect edge simulation function [18] in relation with the global record table allow the scoring function to assign fitness value to a chromosome pair. The perfect edge coverage of the chromosome is satisfactory then the chromosome pair will enter into the delay test. Then the chromosome pair is then updated in the Global Record Table. In [19] authors designed a new infrastructure design analog and digital system. In this new structure the System on Chip (SOC) is combined with the analog cores for efficient edge detection. The remaining parts of the article are structured like this: The article is concluded with a discussion of possible future directions in section 5, after which sections 2 and 3 deal with the literature review, section 3 with the proposed implementation of HSED, section 4 with the analysis of results and performance comparison, and section 2 with the conclusion. The audio CODEC circuits are combined with SOC in cellular phone applications. This infrastructure circuits can be detected in this design. The test wrapper infrastructure circuit consists of digital test control circuit, digital RTCED clock, serial to parallel conversion registers.

The high-speed image edge detection (HSIED) [21] circuit can be combined with other RTCED techniques. The requirement of the analog signal for the RTCED can be obtained from the HSIED. This RTCED approach reduces the edge detector time and overall cost of the system [22]. This HSIED optimization easily handles the analog cores and digital circuits easily. The area of the system is also reduced. The experimental result of edge detector the SOC with the analog cores is presented. In [23] authors proposed digital fuzzy system edge detection (DFSED) algorithm for open defects in the VLSI circuits. The open circuit may take place in two layers of the digital circuits such as inter layer opens and intra layer opens. A branch and bound algorithm were designed, which reduces the functions of ATPG. The flow is designed for detecting the open circuit perfect edge and analyzing the parameters of the digital circuits [24]. By tracking the fan-out structure the interconnect open can be detected. Edge open defect model is designed for the inter layer and intra layer open defects. Energy efficient sobel edge detection (EESED)[25] is used for this model. This model functions by extracting the layout capacitances values obtained from the circuit layout. The open circuit defect model is combined with automatic test pattern generation to improve the edge detector.

Proposed Method

The HSED takes less time to calculate than the traditional Roberts, canny, and prewit edge detection techniques, but because of its wider convolution kernel, which more thoroughly smooths the input picture, the operator is less sensitive to noise. In comparison to Roberts's edges, the operator often yields output values that are noticeably greater for comparable edges. For picture formats that only permit modest integer pixel values, the operator's output values may quickly exceed the maximum permissible pixel value, similar to the Roberts Edge detection (e.g., 8-bit integer images). The remaining portions of the article are structured as follows: The article is concluded with a discussion of possible future directions in section 5, after which sections 2 and 3 deal with the literature review, section 3 discusses the proposed implementation of HSED, section 4 discusses the analysis of results along with a performance comparison, and section 2 deals with the literature survey. Using an image type that allows pixel values with a wider range can help you avoid the issue. Due to the HSED's smoothing effect, lines with natural edges in photos often result in output images that are several pixels wide. To combat this, some thinning could be beneficial. If it doesn't work, hysteresis ridge tracking could be used, similar to the Canny operator. The HSED is able to identify portions of an image that have a high spatial frequency and which correlate to edges by carrying out a

measurement of the two-dimensional spatial gradient on the picture. In most circumstances, it is used to find out the estimated absolute magnitude of the gradient at each place in a grayscale input picture. This is the case because grayscale images include gradients that vary in magnitude. At least in principle, the operator may be partitioned into two convolution kernels, each of which has a total of 33 nodes, as is seen in Figure 1. The other kernel is simply rotated through 90 degrees to create one kernel.

The HSED is able to emphasise parts of an image that have a high spatial frequency and which correlate to edges thanks to a measurement that it does on an image that is called a two-dimensional spatial gradient. The majority of the time, it is used in order to determine the estimated absolute magnitude of the gradient at each place in a grayscale input picture. As can be seen in Figure 1, the operator may, at least in principle, be partitioned into two convolution kernels, each of which has a total of 33 nodes. It is therefore possible to calculate the absolute gradient magnitude at each location as well as the direction of the gradient by combining them. It is possible to determine the magnitude of the gradient by:

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (1)$$

An approximate magnitude is often calculated using:

$$|G| = |G_x| + |G_y| \quad (2)$$

The HSED is able to emphasize areas of an image that have a high spatial frequency and which correlate to edges thanks to a measurement that it does on an image that is called a two-dimensional spatial gradient. In most circumstances, it is used to find out the approximate absolute magnitude of the gradient at each point in a grayscale input picture. This is the case since the magnitude of the gradient might vary significantly depending on the position in the image. As can be seen in Figure 1, the operator may, at least in principle, be decomposed into two convolution kernels, each of which has a total of 33 nodes.

$$\theta = \arctan(G_y / G_x) \quad (3)$$

The HSED is able to identify parts of an image that have a high spatial frequency and which correlate to edges by carrying out a measurement of the 2-D spatial gradient on the picture. In most circumstances, it is used to find out the approximate absolute magnitude of the gradient at each point in a grayscale input picture. This is the case since the magnitude of the gradient might vary greatly depending on the position in the image. As can be seen in Figure 1, the operator may, at least in principle, be segmented into two convolution kernels, each of which has a total of 33 nodes. It stands for pseudo-convolution kernels that are swiftly utilized to calculate the estimated gradient magnitude.

Further, theG_x and G_y are derived by performing the dot-wise matrix multiplication between G_x kernel andG_y kernels to the 3x3 patch of image.

$$G_x = (P_3 + 2P_6 + P_9) - (P_1 + 2P_4 + P_7) \quad (4)$$

$$G_y = (P_1 + 2P_2 + P_3) - (P_7 + 2P_8 + P_9) \quad (5)$$

The estimated magnitude may be calculated using this kernel as follows:

$$|G| = |(P_1 + 2P_2 + P_3) - (P_7 + 2P_8 + P_9) + (P_3 + 2P_6 + P_9) - (P_1 + 2P_4 + P_7)| \quad (6)$$

Here, P_1 to P_9represents the image pixels.Equation6 repress the magnitude of Sobel edge detection process, which consisting of additions, subtractions, and multiplication by factor 2. Figure 3 illustrates the VLSI design of HSED, which is then put into action by the application of Equation 6.

The VLSI design of HSED may be seen shown in Figure 3, and it is put into action with the help of Equation 6. Here, multiplication by factor 2 is implementable by introducing the left shifting operation. Further, this matrix multiplication operation is performed through successive HSCG-SQRT-CSLA additions. In addition, subtraction in matrix multiplication process is achieved by twos complement addition through HSCG-SQRT-CSLA.

Proposed HSCG-SQRT-CSLA

The HSCG-SQRT-CSLA is an improved version of conventional SQRT-CSLA, which is generated by incorporating the HSCG, SCS modules as shown in Figure 4. The conventional SQRT-CSLA does not able to select the carry outputs and sum outputs parallelly, which is overcome in HSCG- SQRT-CSLA by introducing the SCS block. The VLSI design of HSED may be seen shown in Figure 3, and it is put into action by the application of Equation 6. The multiplexer-based selection logic also minimizes the logical delays and propagation delays. Here, multiplexer is used to select the carry of AND outputs and sum of XOR outputs. Here, CIN provides the input to the multiplexer selection purpose. If CIN is zero, the XOR of inputs will be selected; else XNOR of inputs will be selected and resulted in SUM output. Table 1 depicts the truth table of the proposed HSCG-SQRT-CSLA, which consist of A, B, CIN as inputs with SUM, CARRY as outputs, XOR,~XOR,OR,AND are the temporary outcomes.

Further, HSCG-SQRT-CSLA is developed by using HSCG-SCS-Full adder modules. Initially, HSCG-SCS-RCA is developed, which is replaced by conventional RCA modules as shows in Figure 5. The proposed structure of the HSCG-SCG SQRT-based CSLA has been updated from the structure of the prior SQRT-based CSLA in order to accommodate this adding procedure. The earlier modules of traditional CSLA were built using RCA-based structures, and then those structures were updated with RCA-BEC-based structures. At this time, those structures have been further developed to include HSCG and SCS units. SCS will be able to shorten the critical path by using this approach of adding in the processes of HSCG, as well as get rid of the superfluous arithmetic functions and logic operations in sequences. In the long run, arithmetic operations benefit from this adder's ability to cut down on the size of the logic and the propagation latency.

RESULTS AND DISCUSSIONS

For the creation of each and every HSED design, the Xilinx ISE software was used. This piece of software has been designed to provide two distinct kinds of outputs, namely simulation and synthesis. The findings of the simulation allow for a comprehensive investigation of the HSED architecture with regard to the various combinations of input and output byte levels. A simple decoding technique may be approximated by applying a large number of different combinations of inputs and watching a wide variety of outputs while doing a simulation study of accurate encoding. As a consequence of the conclusions of the synthesis, the use of space in proportion to the number of transistors will be carried out. In addition, a time summary will be acquired with reference to the different route delays, and a power summary will be generated making use of the static and dynamic power consumption. Both of these summaries will be done. The results of running the HSED simulation are shown in figure 6. In this situation, P0, P1, P2, P3, P4, P5, P6, P7, and P8 are the input pins, and each of them has 64 bits, while out is the output port, which has 65 bits.

The design (area) overview of the suggested technique may be seen in Figure 7. In this case, the suggested technique makes use of a very small portion of the available slice LUTs, namely 1117 of the total 17600. The timing breakdown of the suggested technique is shown in Figure 8. In this instance, the suggested procedure used a total of 21.217ns of time delay, of which 1.720ns of delay was logical and 19.4974ns of delay was route. The report on the power consumption of the proposed HSED may be seen in Figure 9. In this scenario, the suggested approach has a power consumption of 0.166 watts. The results of the performance assessment of several HSED approaches are compared in Table 2. In this case, the proposed HSED resulted in superior (reduced) performance in terms of LUTs, time-delay, and power consumption in comparison to conventional approaches such as Optimized RTCED [20], HSIED [21], and DFSED [23]. This was the case because the proposed HSED resulted in fewer LUTs, a shorter time-delay, and lower power consumption.

CONCLUSION

The implementation of HSED using HSCG-SQRT-CSLA is the main goal of this work. By applying the matrix multiplication of image pixel values to the HSED, the HSED operation is carried out. Additionally, this matrix multiplication operation is carried out using a series of additions using the HSCG-SQRT-CSLA formula. Additionally, the HSCG-SQRT-CSLA twos complement addition method is used to accomplish subtraction in the matrix multiplication process. Based on the results of the simulations, the newly suggested HSED performed much better than the conventional edge detection methods. Combined edge detection approaches such as sobel-canny and sobel-prewit that use modified adders and subtractors can be used to enhance the scope of this study.

REFERENCES

1. Chakrapani, Y. Sri, N. Venkatewsara Rao, and M. Kamaraju. "A survey of sobel edge detection vlsi architectures." *Journal of Physics: Conference Series*. Vol. 1804. No. 1. IOP Publishing, 2021.
2. Lee, Da-Huei, et al. "High-Efficient Low-Cost VLSI Implementation for Canny Edge Detection." *Journal of Information Science & Engineering* 36.3 (2020).
3. Joshi, Rajeev, Md Adnan Zaman, and Srinivas Katkoori. "Novel bit-sliced near-memory computing based VLSI architecture for fast Sobel edge detection in IoT edge devices." *2020 IEEE International Symposium on Smart Electronic Systems (iSES)(Formerly iNiS)*. IEEE, 2020.
4. Gayathri, A. G., and RemyaAjai AS. "Vlsi implementation of improved sobel edge detection algorithm." *2021 International Conference on Communication, Control and Information Sciences (ICCISc)*. Vol. 1. IEEE, 2021.
5. Kumar, Pardeep, and Kuldeep Singh. "Hardware Model for Efficient Edge Detection in Images." *2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, 2020.
6. Zhong, Xiaopeng, et al. "A 2pJ/pixel/direction MIMO processing based CMOS image sensor for omnidirectional local binary pattern extraction and edge detection." *2018 IEEE Symposium on VLSI Circuits*. IEEE, 2018.
7. Vardhana, M., et al. "Convolutional neural network for bio-medical image segmentation with hardware acceleration." *Cognitive Systems Research* 50 (2018): 10-14.
8. Manikandan, L. C., Selvakumar, R. K., Nair, S. A. H., &Sanal Kumar, K. P. (2021). Hardware implementation of fast bilateral filter and canny edge detector using Raspberry Pi for telemedicine applications. *Journal of Ambient Intelligence and Humanized Computing*, *12*(5), 4689-4695.
9. Ni, C. T., Chang, C. A., Liao, J. W., & Chen, P. Y. (2020, December). A novel vlsi architecture for barrel distortion correction. In *2020 International Computer Symposium (ICS)* (pp. 209-213). IEEE.
10. Katayama, Takafumi, et al. "Boundary correlation-based intracoding for SHVC algorithm and its efficient VLSI architecture." *Journal of Real-Time Image Processing* 15.1 (2018): 107-122.
11. Zhang, Yangyi, et al. "Efficient VLSI Architecture for Edge Sensing Anti-Aliasing Demosaicing." *2021 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*. IEEE, 2021.
12. Yang, Xiaokun, T. Andrew Yang, and Lei Wu. "An Edge Detection IP of Low-Cost System on Chip for Autonomous Vehicles." *Advances in Artificial Intelligence and Applied Cognitive Computing*. Springer, Cham, 2021. 775-786.
13. Ravindran, Ammu, and P. Rajeswari. "FPGA Implementation of Approximate Multiplier with Edge Detection Application." *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2022.
14. Mehena, J. (2019). Medical image edge detection using modified morphological edge detection approach. *International Journal of Computer Sciences and Engineering*, *7*(6), 523-528.
15. Devaraj, S. A., Muthukumaran, N., Jacob, S., Rosita, J. D., Vijay, M. M., &Roobert, A. A. (2022). Performance analysis of reconstructed image using colourdemosaicing algorithm for digital images. *International Journal of Innovative Computing and Applications*, *13*(3), 138-149.
16. Sasongko, Arif, and Bima Sahbani. "VLSI Architecture for Fine Grained Pipelined Feature Extraction using Histogram of Oriented Gradient." *2019 IEEE 7th Conference on Systems, Process and Control (ICSPC)*. IEEE, 2019.

17. Yildirim, Melih, and FiratKacar. "Retina-inspired neuromorphic edge enhancing and edge detection." *AEU-International Journal of Electronics and Communications* 115 (2020): 153038.
18. Kuo, Y. T., Chen, W. T., Chen, P. Y., & Li, C. H. (2019). VLSI implementation for an adaptive haze removal method. *IEEE Access*, *7*, 173977-173988.
19. Behbahani, Fereshteh, et al. "Leveraging negative capacitance CNTFETs for image processing: An ultra-efficient ternary image edge detection hardware." *IEEE Transactions on Circuits and Systems I: Regular Papers* 68.12 (2021): 5108-5119.
20. Soares, Leonardo Bandeira, et al. "An energy-efficient and approximate accelerator design for real-time canny edge detection." *Circuits, Systems, and Signal Processing* 39.12 (2020): 6098-6120.
21. Mannepalli, Y., Korede, V. B., & Rao, M. (2021, June). Novel Approximate Multiplier Designs for Edge Detection Application. In *Proceedings of the 2021 on Great Lakes Symposium on VLSI* (pp. 371-377).
22. Menaka, R., RamadossJanarthanan, and K. Deeba. "FPGA implementation of low power and high speed image edge detection algorithm." *Microprocessors and Microsystems* 75 (2020): 103053.
23. Nausheen, Nazma, et al. "A FPGA based implementation of Sobel edge detection." *Microprocessors and Microsystems* 56 (2018): 84-91.
24. Bozorgmehr, Ali, et al. "A novel digital fuzzy system for image edge detection based on wrap-gate carbon nanotube transistors." *Computers & Electrical Engineering* 87 (2020): 106811.
25. Peng-o, Thaufig, and PanyayotChaikan. "High performance and energy efficient sobel edge detection." *Microprocessors and Microsystems* 87 (2021): 104368.

Table 1. Verification table of HSCG-SCS-Full adder

A	B	CIN	~XOR	XOR	AND	OR	SUM	CARRY
0	0	0	1	0	0	0	0	0
0	0	1	0	1	0	1	1	0
0	1	0	0	1	0	1	1	0
0	1	1	1	0	1	1	0	1
1	0	0	1	1	0	0	1	0
1	0	1	0	0	0	1	0	1
1	1	0	0	0	0	1	0	1
1	1	1	1	1	1	1	1	1

Table 2. Performance Comparison.

Metric	RTCED [20]	HSIED [21]	DFSED[23]	Proposed HSED
LUTs	2913	2448	1572	1117
Time delay (ns)	43.28	32.284	.453	21.217
Power consumption (w)	1.49	1.34	1.049	0.166

-1 0 +1 -2 0 +2 -1 0 +1	+1 +2 +1 0 0 0 -1 -2 -1
Figure 1. Sobel convolution kernels, (a) G_x kernel, (b) G_y kernel	

P_1	P_2	P_3
P_4	P_5	P_6
P_7	P_8	P_9

Figure 2: 3x3 patch of image.

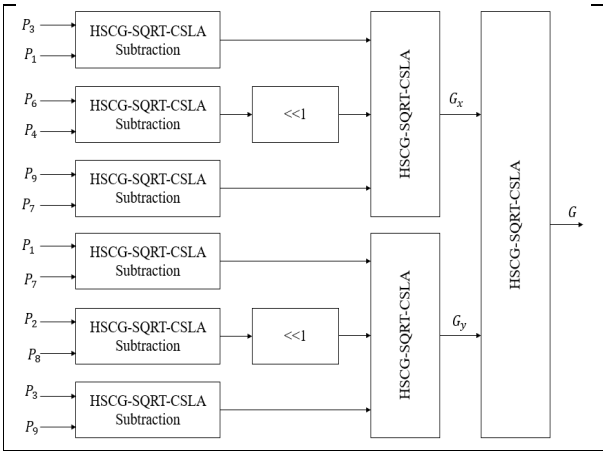

Figure 3. Proposed HSED Architecture.

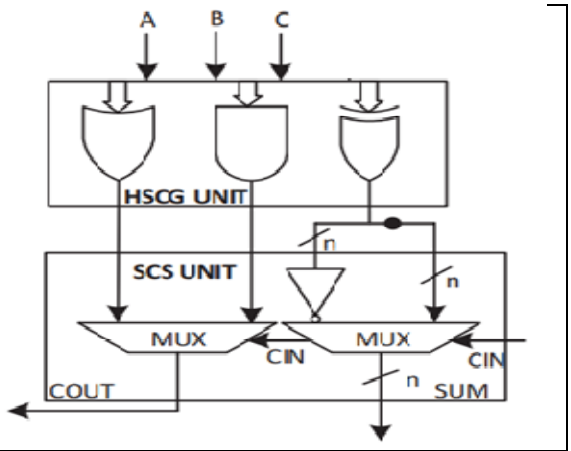

Figure 4. HSCG-SCS-Full adder

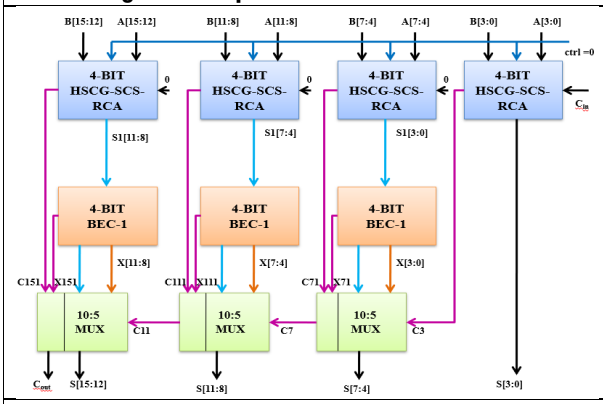

Figure 5. Architecture of HSCG-SQRT-CSLA.

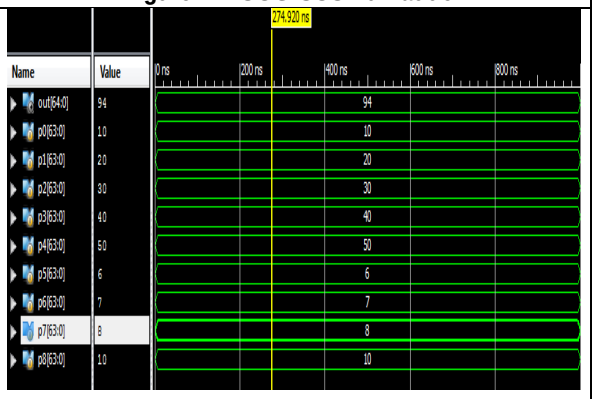

Figure 6. Simulation outcome of HSED

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slice LUTs	1117	17600	6%
Number of fully used LUT-FF pairs	0	1117	0%
Number of bonded IOBs	573	100	573%

Figure 7. Design summary.

```
LUT6:I2->O            1    0.043   0.343  A9/KK1/s6/gc_16/f2/r3_SW0 (N1072)
LUT6:I4->O            8    0.043   0.582  A9/KK1/s6/gc_16/f2/r3 (A9/KK1/g6<31>)
LUT6:I0->O            2    0.043   0.293  A9/KK2/s5/gc_0/f2/r1 (A9/KK2/s5/gc_0/f2/r1)
LUT6:I5->O            8    0.043   0.582  A9/KK2/s5/gc_0/f2/r2 (A9/KK2/g5<7>)
LUT6:I0->O            6    0.043   0.573  A10/KK2/s1/pg_8/f2/g1 (A10/KK2/p1<8>)
LUT6:I0->O            5    0.043   0.561  A10/KK2/s2/bc_8/f3/a_c_AND_2_o1 (A10/KK2/p2<8>)
LUT5:I0->O            3    0.043   0.466  A10/KK2/s3/bc_8/f3/a_c_AND_2_o1 (A10/KK2/p3<8>)
LUT4:I0->O            5    0.043   0.569  A10/KK2/s4/bc_8/f3/a_c_AND_2_o1 (A10/KK2/p4<8>)
LUT6:I0->O            4    0.043   0.442  A10/KK2/s5/bc_0/f2/r (A10/KK2/g5<15>)
LUT6:I3->O            1    0.043   0.343  A10/KK2/s6/gc_16/f2/r3_SW0 (N1066)
LUT6:I4->O            4    0.043   0.556  A10/KK2/s6/gc_16/f2/r3 (A10/KK2/g6<31>)
LUT5:I0->O            5    0.043   0.308  A11/KK1/s2/gc_0/f2/r1 (A11/KK1/g2<0>)
LUT5:I4->O            6    0.043   0.451  A11/KK1/s3/gc_1/f2/r1 (A11/KK1/g3<2>)
LUT6:I3->O            2    0.043   0.347  A11/KK1/s5/gc_7/f2/r1 (A11/KK1/s5/gc_7/f2/r1)
LUT6:I4->O            2    0.043   0.554  A11/KK1/s5/gc_7/f2/r (A11/KK1/g5<14>)
LUT6:I0->O            1    0.043   0.279  A11/KK1/s7/Mxor_o_s<31:1>_14_xo<0>1 (out_15_OBUF)
OBUF:I->O                 0.000          out_15_OBUF (out<15>)
----------------------------------------
Total                      21.217ns (1.720ns logic, 19.497ns route)
                                    (8.1% logic, 91.9% route)
```

Figure 8. Time summary

Device	
Family	Virtex4
Part	xc4vfx12
Package	sf363
Temp Grade	Commercial
Process	Typical
Speed Grade	-12

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250

Characterization	
PRODUCTION	v1.0,02-02-08

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	1	---	---
Logic	0.000	23	10944	0
Signals	0.000	38	---	---
DCMs	0.000	0	4	0
IOs	0.000	18	240	8
Leakage	0.166			
Total	0.166			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	14.7	82.6	52.4

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.200	0.071	0.000	0.071
Vccaux	2.500	0.031	0.000	0.031
Vcco25	2.500	0.001	0.000	0.001

	Total	Dynamic	Quiescent
Supply Power (W)	0.166	0.000	0.166

Figure 9. Power Summary.