*Grenze International Journal of Engineering and Technology, Jan Issue*

# Efficient VLSI Architecture of FIR Filter for Processing Seismic Signal

Nirup[1], Dr. V. Anandi[2] and Dr. M. Ramesh[3]
[1]Department of ECE, Ramaiah Institute of Technology, Bangalore
Email: akashnirup@gmail.com
[2]Associate Professor, Department of ECE, Ramaiah Institute of Technology, Bangalore
Email: anandi.v@msrit.edu
[3]Associate Professor, School of Management Studies, Presidency University, Bangalore
Email: ramandsur@gmail.com

***Abstract*—The main requirements for filters used in real-time seismic warning systems are low complexity, high speed, and reconfigurability. To achieve these objectives, a combination of techniques is employed. This includes minimizing logic operators and logic depths within the FIR filter and using a Carry Look Ahead Adder (CLA) to reduce critical path delay (CPD). Additionally, the common sub-expression elimination (CSE) technique, based on canonical signed digits (CSD), is commonly utilized to decrease hardware complexity. The proposed design of an efficient FIR filter incorporates the techniques. This results in several advantages, including superior performance in terms of area, power, and delay. The implementation of this architecture is done using Verilog, and the results obtained are analyzed through simulation utilizing Xilinx Vivado 2019.1 and Cadence 45nm technology. The architecture proposed yields a 3%,70%, and 83% reduction in CPD, Number of LUTs, and SDP over state of art CSE-based architecture.**

***Index Terms*— Finite impulse response (FIR), a canonical signed digit (CSD), common sub-expression elimination (CSE), carry look-ahead adder (CLA), Vertical Common Sub Expression (VCSE), Horizontal Common Sub Expression (HCSE), Partial Product (PP).**

## I. INTRODUCTION

Seismic refers to an earthquake or something related to one. Sometimes noise may be added to this signal, and it will be difficult for the alert system of Seismic to recognize whether the signal is due to an earthquake, volcano, etc; The key problem in processing the seismic signal is to remove the noise, which can occasionally be added to the signal. For calculation, band selection, signal analysis, etc., digital filters are utilized. Finite Impulse Response Filter is the name given to the impulse response over a finite time period. Many of its uses are in the areas of image processing, voice processing, and signal processing for signals in signals. The FIR filter is important in creating effective stable filters with linear phase and unconditional stability because of its outstanding properties. The main goal of this filter is to maintain important signals by clipping off undesired noise and distortion. This filter is beneficial for significant signal processing applications because of the critical components such as pre-processing of signal, anti-aliasing, band selection, interpolation, and low-pass filtering. The FIR filter will emerge as the best option because it does not require bit truncation or bit rounding to produce a noise-free filter. Low complexity, high speed, and reconfigurability are the three main demands of these FIR

*Grenze ID: 01.GIJET.10.1.512_1*

filters which are used in the processing of the obtained seismic signal in a real-time alert system of Seismic. Realizing these filters with the appropriate accuracy level in real-time is a difficult undertaking. Many techniques and methods have been realized in existing "Reference [10]-[15]" in designing efficient FIR filters. The optimized FIR filter is done by using CSE (Common Sub Expression Elimination Method) and has gained more importance in "Reference [1]-[7]", and the use of adders like CLA "Reference [8]-[9]". Therefore, the main aim is to design an efficient VLSI architecture for the FIR filter.
The major works which are done are presented in this brief are as follows.
i) The pre-processing of seismic signals has been implemented using a high-speed and low-power FIR filter. The suggested architecture employs the CSE algorithm with CSD-coded filter coefficients as input data to reduce hardware costs. Carry Look Ahead Adder (CLA) is also used to attain high speed (low CPD). The proposed structure outperforms the FIR architectures already in use, according to a comparison of FIR filter architecture for seismic applications "Reference [1]".
ii) The crucial requirement for real-time applications is the dynamically changeable filter coefficients. Consequently, the state-of-the-art "Reference [1]" in terms of the complexity of hardware and CPD is outperformed by the CSE-based architecture that has been presented. The outline of the brief is as follows.
The current algorithms are discussed in Section II. Section III of the approach has been made available. Section IV discusses the FIR filter's potential VLSI architectures. Results obtained from the implementation are provided in Section V. Section VI brings the brief to a close.

## II. Preliminaries

In a structure where a group of constant multipliers is used for multiplying a common variable, such as when the common input x(n) variable is multiplied with all the constant coefficients H(n), using fundamental operations like addition, and hardware shifting to get the output, the CSE (Common Sub Expression Elimination) algorithm technique is used to minimize the logical operators and logical depths to reduce the hardware complexity.

### *A. Encoding Coefficients of Filter in CSD*

To reduce the hardware expense of the FIR design, constant coefficients might be represented as canonical signed digits (CSD). CSD can represent a signed number with bit values -1,0,1. For example, Consider the 4-bit representation in binary of decimal value ho = 7, i.e., 0111, then the CSD representation of 7 can be written as $(2^3\text{-}2^0)$ and in binary representation (100-1). As there are three nonzero bits, constant multiplication requires the use of two adders. However, the 4-bit CSD representation will be 100-1, contain only two non-zero bits, and multiply (ho X x(n)) with a single adder/subtractor operation.
Generating Magnitude and Sign values of each coefficient for their respective CSD values that is the above CSD coefficient value will have a Sign value of h0=0001 and a Magnitude value of h0=1001.

### *B. CSD-Based VCSE (Vertical CSE) Algorithm*

There is a presence of vertical common sub-expressions (VCSs) i.e,[1 1] and [1 $\bar{1}$] and their versions of negative vertically "Reference [1][2][3]" in CSD-based coefficients of a filter as shown in Fig. 2.

### *C. CSD-Based HCSE (Vertical CSE) Algorithm*

We can also observe there is a repetitive occurrence of HCSs [101], [10$\bar{1}$], [1001], [100$\bar{1}$] and their versions of negative in CSD-based filter coefficients "Reference [1][2][3]" as shown in Fig. 2.

### *D. CLA (Carry Look-Ahead Adder)*

The adder produces carry propagation delay while performing other arithmetic operations like multiplication and divisions as it uses several addition or subtraction steps. This is a major problem for the adder and hence improving the speed of addition will improve the speed of all other arithmetic operations. Hence reducing the carry propagation delay of adders is of great importance. There are different logic design approaches that have been employed to overcome the carry propagation problem. One widely used approach is to employ a carry look-ahead which solves the delay problem by calculating the carry in advance, based on the inputs "Reference [8][9]".

## III. Methodology Proposed

The steps involved in the methodology of the Filter proposed are described below.
i)Generating the coefficients of desired frequency from Matlab based on the order numbers.

ii) Getting the CSD representation of filter coefficients and storing their sign and magnitude values in LUTs as in Fig. 2.
iii)Sign and magnitude bit is partitioned into 5 groups of 3bits P1=(s[14:12],m[14:12]), P2=(s[11:9],m[11:9]), P3=(s[8:6],m[8:6]), P4=(s[5:3],m[5:3]), P5=(s[2:0],m[2:0]).
iv) Partial product unit to generate the products partially by add and shift technique.
v) 3-bit CSDs used are "001", "010", "100", "101", and "10$\bar{1}$" and their versions of negative, which produce partial products(x1-x10).
vi)Control Signal Generator identifies similarity among partitioned groups by comparing sign and magnitude parts of every group with others and reducing the logical operators is done.
vii) Based on 3-bit VHCSE, partial products are chosen using muxes.
viii) Area, power, and latency are reduced by CLA "Reference [8][9]".
ix)Synthesis is done in both Xilinx Vivado 2019.1 and Cadence 45nm technology to obtain area, power, and delay.

## IV. VLSI Architecture of FIR Filter Proposed

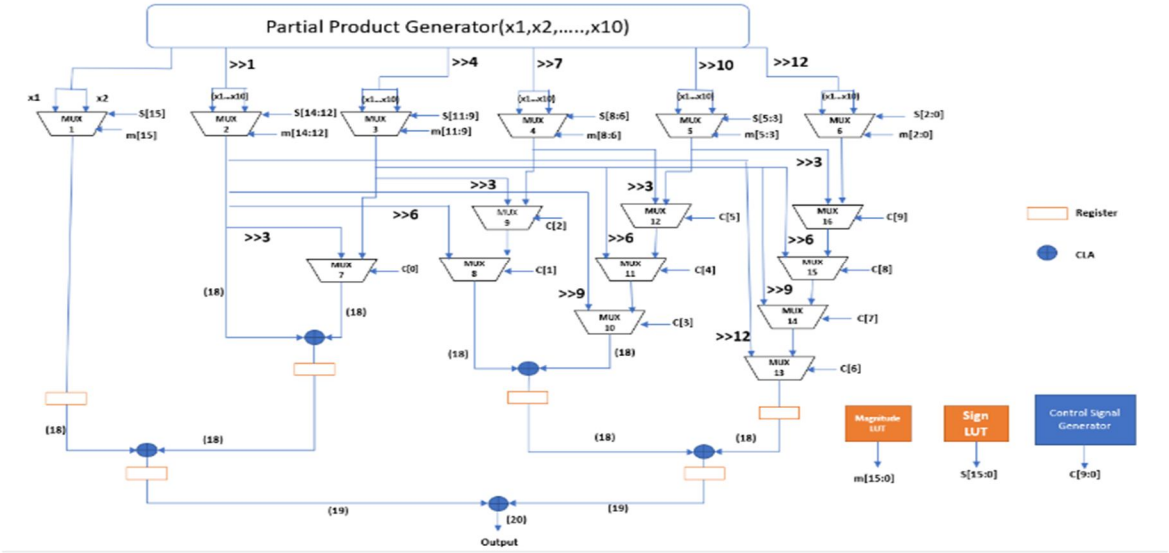


Figure 1. FIR filter Architecture Proposed

The four units that make up the proposed FIR filter's architecture are shown in Fig. 1: (i) A unit for generating partial products (PPs) using the shift-and-add method. The CSG compares the magnitude and sign components of every group with those of the other groups to determine if groups P1 to P5 are comparable to one another. The inside structure of PP generating (PPG) is seen in Fig. 3. Fig. 5 depicts the internal organization of the control signal unit; (iii) Muxes (MUX2- MUX6) layer selects the PP according to 3-bit VHCSE as illustrated in Fig. 4. (iv) an added layer under control according to the CSE, Carry Look Ahead adders are utilized to carry the controlled additions of multiplexers (MUX7-MUX16) PPs. CPD is decreased with CLA adder "Reference [8][9]".
The main blocks of FIR Filter architecture include

### *A. CSD, Sign, and Magnitude values format*

Fig. 2 shows an example of the CSD representation of 3rd-order 16-bit coefficients for certain frequencies and their sign and magnitude representation values and how they will be stored in LUTs "Reference [2]".

### *B. Partial Product Generator*

The partial product generator unit Fig. 3 is obtained from the Fig. 4 which doesn't contains a consequent number of one's according to the CSE algorithm. Considering the 3-bit representation of our CSD values, the possible repetitive patterns x1 to x10 are generated. Where x1, x3, x5, x7, x9 and their negative versions x2, x4, x6, x8, x10 are obtained by 2's complement.

CSD

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H0 | 0 | 0 | -1 | 0 | 0 | -1 | 0 | 1 | 0 | 0 | 0 | -1 | 0 | 1 | 0 | 0 |
| H1 | 0 | 1 | 0 | -1 | 0 | 0 | -1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| H2 | 1 | 0 | 1 | 0 | 1 | 0 | -1 | 0 | 1 | 0 | 0 | 1 | 0 | -1 | 0 | 0 |
| H3 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | -1 | 0 | 1 | 0 | -1 | 0 | 0 | 0 | -1 |

Magnitude

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| H1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 |
| H2 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| H3 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

Sign

| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| H0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| H1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| H2 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| H3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |

Figure 2. CSD VHCSE algorithm applied on coefficient set of 3-tap filter

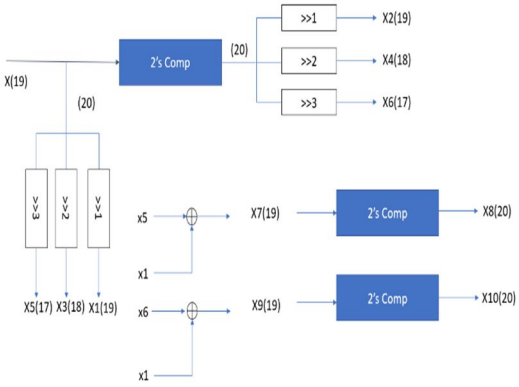


Figure 3. Partial Product Generator

| H[2] | H[1] | H[0] | M[2] | M[1] | M[0] | S[2] | S[1] | S[0] | NO |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | x1 |
| 0 | 0 | -1 | 0 | 0 | 1 | 0 | 0 | 1 | x2 |
| 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | x3 |
| 0 | -1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | x4 |
| 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | x5 |
| -1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | x6 |
| 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | x7 |
| 1 | 0 | -1 | 1 | 0 | 1 | 0 | 1 | 0 | x9 |

x7 = x1 + x5 = 101
X8=2'scomp(x7)

X9 = x1 + x6=10 $\bar{1}$
X10= 2's comp(x9)

Figure 4. Partial Product Generator for 3-bit CSD-Based Common Sub Expressions

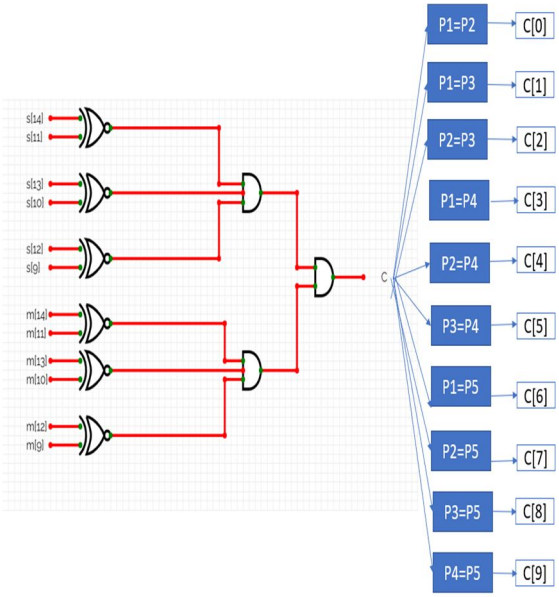


Figure 5. Control Signal Generator Unit

### *C. Control Signal Generator*

Control Signal Generator Unit takes the 16-bit coefficients generated from Matlab and partitions the Sign and magnitude bits of CSD-based coefficients into 5 groups of 3bits P1=(s[14:12],m[14:12]), P2=(s[11:9],m[11:9]), P3=(s[8:6],m[8:6]), P4=(s[5:3],m[5:3]), P5=(s[2:0],m[2:0]). The Control Signal Generator block will produce 10 control signals depending on the equality check for 10 different cases. The control signal generator unit is seen in

Fig. 5. It compares the sign and magnitude components of each partitioned group with those of the other groups to determine how similar they are.

*D. Mux2-Mux6*

The mux layer selects partial product corresponding to 3bit CSE as in Fig. 1.

*E. Mux7-Mux16*

These muxes select partitioned groups, where P1 is compared with P2, if P1 is the same as that of P2 then it is right shifted by 3 bits, again p1 is compared with P3, if it is the same then it is right shifted by 6 bits and so on.

Similarly, P2 is compared with P3, P4, and P5 and they are shifted based on the comparison.

And this continues with the other groups P3, P4, and P5 as shown in Fig. 1.

## V. FPGA AND ASIC IMPLEMENTATION RESULTS AND COMPARISON

The behavioral model of the filter is coded in Verilog. By Xilinx's Vivado design suite 2019.1, we were able to synthesize the filter while optimizing the design. As the target device, we used the Xilinx ZYNQ-XC7Z020-1CLG484 FPGA device, and we got results from the synthesis and implementation. Cadence's 45nm technology library has been used to develop ASICs.

TABLE I. RESULT OF THE PROPOSED FILTER

| Filter Order Number | Number Of Slices | Slice Look Up Tables | Slice Registers | Power(W) | Critical Path Delay(ns) | Slice Delay Product(us) |
|---|---|---|---|---|---|---|
| 16 | 60 | 190 | 100 | 0.131 | 2.811 | 0.168 |

TABLE II. VLSI FPGA IMPLEMENTATION COMPARISON

| Filter Order | Number of SLUT's | Critical Path Delay (ns) | Slice Delay Product (us) |
|---|---|---|---|
| 16" Reference [1]" | 746 | 2.88 | 1.298 |
| 16 proposed | 190 | 2.811 | 0.168 |

Table I shows the results of area, power, and CPD of the 16$^{th}$ of the Filter architecture proposed in FPGA Implementation. The Slice number in the 16$^{th}$ Order is 60, the Slice number LUTs are 190 in the 16$^{th}$ Order, and the Slice numbers Registers are 100 in the 16$^{th}$ Order. Power Dessiptaed in the 16$^{th}$ Order is 0.131W. The Critical path delay(CPD) of the 16$^{th}$ Order is 2.811ns and the SDP is 0.168(us) for the 16$^{th}$ Order. The proposed architecture and the current architecture "Reference [1]" have been compared, as shown in Table II.

Table III shows the ASIC implementation results of the proposed FIR filter of the 16$^{th}$ order in Cadence 45nm technology which has 665 in the 16$^{th}$ Order, Power dissipated is 137.381uW of the 16$^{th}$ Order and the Delay is 2238ps of the 16$^{th}$ Order.

TABLE III. VLSI ASIC IMPLEMENTATION RESULTS OF PROPOSED FIR FILTER ARCHITECTURE USING CADENCE 45NM TECHNOLOGY

| Filter Order Number | Area(cells) | Power(uW) | Delay(ps) |
|---|---|---|---|
| 16 | 665 | 137.381 | 2238 |

## VI. CONCLUSION

In this work, a FIR filter architecture is proposed. Matlab tool is used to generate coefficients of different orders of different frequencies based on input signal frequency. The FIR filter proposed is coded in Verilog and it was simulated and synthesized using Xilinx vivado 2019.1 tools. The proposed filter provides less delay compared with the existing filter by 3%. Table I shows the complete area, power, and delay summary of our proposed FIR filter for the 16th order. Table II represents the comparison report of FIR filter proposed and the existing reconfigurable FIR filter, where the proposed one is better than the existing one" Reference [1]". Red hat enterprise 6 workstations are used to implement the same FIR filter proposed in cadence 45nm technology to generate area, power, and delay reports.

Table III shows the area, power, and delay of the 16th order of the proposed filter obtained from cadence 45nm technology.

## REFERENCES


[1] Sudipta Bose, Arijit De, Member and Indrajit Chakrabarti, "Area-Delay-Power Efficient VLSI Architecture of FIR Filter for Processing Seismic Signal," IEEE Transactions on Circuits and Systems, 2021.

[2] I. Hatai, I. Chakrabarti, and S. Banerjee, "A computationally efficient reconfigurable constant multiplication architecture based on CSD decoded vertical–horizontal common sub-expression elimination algorithm," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 65, no. 1, pp. 130–140, Jan. 2018.

[3] I. Hatai, I. Chakrabarti, and S. Banerjee, "An efficient constant multiplier architecture based on vertical horizontal binary common sub-expression elimination algorithm for reconfigurable FIR filter synthesis," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 62, no. 4, pp. 1071–1080, Apr. 2015.

[4] S. Roy and A. Chandra, "A triangular common subexpression elimination algorithm with reduced logic operators in FIR filter," IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 67, no. 12, pp. 3527–3531, Dec. 2020.

[5] Mitsuru Yamada, Akinori Nishihara, "A high-speed FIR digital filter with CSD coefficients implemented on FPGA," Proceedings of the 2001 Asia and South Pacific Design Automation Conference, 2001.

[6] Chia-Yu Yao, Hsin-Horng Chen, Tsuan-Fan Lin, Chiang-Ju Chein, Chun-Te Hsu, "A novel common-subexpression-elimination method for synthesizing fixed-point FIR filters," IEEE Transactions on Circuits and Systems I,2004.

[7] A. P. Vinod, E. Lai, D. L. Maskell, and P. K. Meher, "An improved common subexpression elimination method for reducing logic operators in fir filter implementations without increasing logic depth," Integration, vol. 43, no. 1, pp. 124–135, 2010.

[8] Jasbir Kaur, Lalit Sood," Comparison Between Various Types of Adder Topologies," International Journal of Computer Science And Technology, 2015.

[9] Basavoju Harish, Dr.K.Sivani, Dr. M.S.S. Rukmini," Design and Performance Comparison among Various Types of Adder Topologies," IEEE Third International Conference on Computing Methodologies and Communication,2019.

[10] S. Akash, M. Ajeeth, Radha. N," An Efficient Implementation of FIR Filter Using High-Speed Adders for Signal Processing Applications," IEEE International Conference on Inventive Research in Computing Applications, 2020.

[11] Neha Goel, Ashutosh Nandi, "Design of FIR filter using FCSD Representation," IEEE International Conference on Computational Intelligence & Communication Technology,2015.

[12] N Sameeksha Rai, Pannaga Shree B S, Meghana Y P, Arunkumar P Chavan, Dr.H.V. Ravish Aradhya," Design and implementation of 16 tap FIR filter for DSP Applications," IEEE International Conference on Advances in Electronics, Computer and Communications, 2018.

[13] D. Xu and J. Chiu, "Design of a high-order FIR digital filtering and variable gain ranging seismic data acquisition system," in Proc. IEEE Southeast, Charlotte, NC, USA, Apr. 1993.

[14] Usha Maddipati, Shaik Ahemedali, Maddipati Sri Sai Ramya, M D Praneeth Reddy, K N J Priya," Comparative analysis of 16-tap FIR filter design using different adders," IEEE FIR Filters, 2020.

[15] S. Akash, M. Ajeeth, Radha. N," An Efficient Implementation of FIR Filter Using High-Speed Adders for Signal Processing Applications," IEEE International Conference on Inventive Research in Computing Applications, 2020.