ISSN 2278-3091
Volume 9, No.5, September - October 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse79952020.pdf
https://doi.org/10.30534/ijatcse/2020/79952020

Improved Scheduling Algorithm for the IIR Filter design for Low power VLSI Applications

Ngangbam Phalguni Singh[1], Shruti Suman[2]
[1,2] Department of Electronics and Communication Engineering, KoneruLakshmaiah Education Foundation (KL University), Vaddeswaram, Andhra Pradesh, India
phalsingh@gmail.com[1], shrutisuman23@gmail.com[2]

ABSTRACT

The behavioural synthesis of infinite impulse response (IIR) filter is a challenge when it comes to high speed VLSI design applications. Implementation in FPGA is quite impossible due to huge hardware requirements and it is a major obstacle to widespread acceptance. This paper presents the algorithm that makes design decisions to reduce power consumption. The operations are performed with proper scheduling techniques like as soon as possible (ASAP) and as late as possible (ALAP). The IIR filter is considered for this implementation so that it makes easier to identify the operations with successive clock cycles and resource utilization. The power consumption has been reduced by 15.27% with sampling rate from 10 MHz to 79 KHz.

Keywords: IIR Filter, FPGA, ASAP, ALAP

1. INTRODUCTION

With increasing applications of VLSI design in digital signal processing, the hardware implementation of filters has become the essential demand. Behavioural synthesis is one of the effective tools in designing the IIR filters in terms of register operations and transform into gates and flip flops. Previously, the conventional programming language C was used to design. However, high-level algorithmic description can also be used for the same design. It can generate a structural representation of the circuit for the filter. Application Specific Integrated Circuits (ASICs) or Digital signal processors were used for hardware implementation of IIR filters. The hardware implementation could have atleast 12 16-bit multipliers and 11 16-bit adders. This may use almost 14 FPGAs with 442544 adders in total with same number of clock cycles. This limitation does not allow to use behavioural synthesis for higher order of IIR filter. This paper proposes an algorithmic approach that minimizes the number of resource utilization with proper scheduling techniques.

The IIR filter implementation with 256 registers and 256 multipliers is shown in figure 1 [1]. A conventional DSP processor takes 256 clock cycles whereas FPGA takes only a single clock cycle for the same filter implementation.

This flexible architecture can have 256 MAC operations per sample. The distributed DSP resources are LUT, registers, multipliers and memory units [2] – [5]. The parallel processing system maximizes the throughput and optimize the performance/cost tradeoff [6] – [8].

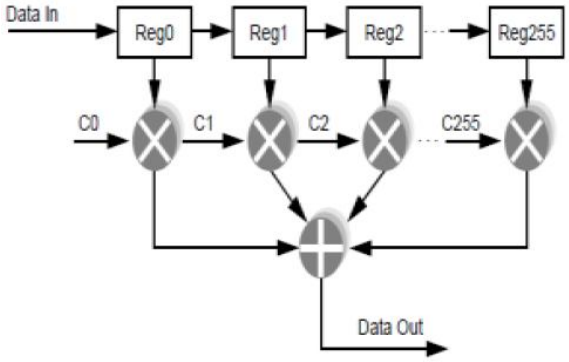

Figure 1: FPGA implementation of IIR filter [1]

This paper contains the IIR filter design algorithms that are modified for better VHDL descriptions in section 2. The scheduling of various resources is described in section 3 with proper graphs. Section 4 shows the hardware implementation ofthe IIR filter. Section 5 represents the result analysis of the design. The conclusion is given in section 6.

2. IIR FILTER ALGORITHM

There are two approaches of IIR filter implementation:

a) Direct Form I
b) Direct Form II

Both consists of forward path and feedback path which contribute to obtain the output. The filter is expressed with Z transfer functions and simulated in Matlab. The fixed point implementation of IIR filter has certain limitations:

a) Co-efficient Quantization
b) Internal Quantization
c) Overflow
d) Stability

The proposed algorithm is used to demonstrate the operations that can be executed concurrently with successive clock cycles [9]. It will identify the dependency of data on each other to expand the loops in behavioural description using FPGA [10]. Algorithm 1 shows the behavioural descriptions of the IIR filter. The generic ports are defined as:

```
generic (coeffa: integer_array (0 to order);
coeffb: integer_array (0 to order - 1));
port(input: in int; strobe: in BIT; output: out int);
```

Based on these generic ports, algorithm 1 is developed for generating the structure of IIR filter. Delays are introduced for clocking the modules, the filter behaviors are well described by representing the algorithms in the graphical formats.

Algorithm 1: Behavioural description

```
behavioural (descriptions_IIR)
{

begin
input_sum := input;
for j in 0 to order - 1 loop
input_sum := input_sum + (delay(j)*coeffb(j))/1024;
end loop;
output_sum := (input_sum*coeffa(order))/1024;
for k in 0 to order loop
output_sum := output_sum + (delay(k)*coeffa(k))/1024;
end loop;
for l in 0 to order - 1 loop
delay(l) := delay (l + 1);
end loop;
delay(order) := input_sum;
output<= output_sum;
wait on strobe;
end process;

}
```

Successive values can be differentiated by separating them with *single assignment form.* In algorithm 2, output_sum indicates the successive values.

Algorithm 2: Single assignment form

```
schedule (assignment form)
{

input_sum := input + delay(0)*coeffb(0);
output_sum0 := input_sum*coeffa(1);
output_sum1 := output_sum0 + delay(0)*coeffa(0);
output := output_sum1 + delay(1)*coeffa(1);
}
```

The data dependency graph is constructed in figure 2 based on algorithm 1. However, it is also assumed that each multiplication and addition are performed at one clock cycle. An as an *as soon as possible* (ASAP) schedule is introduced to this graph showing five clock cycles; latency of five.

This scheduling will perform the operations as early as possible in a single clock cycle. It can be inserted using algorithm 2 in the hardware descriptions of the IIR filter design. But some of the operations are performed lately as required by the design structure. For that, as late as possible (ALAP) schedule technique is used.

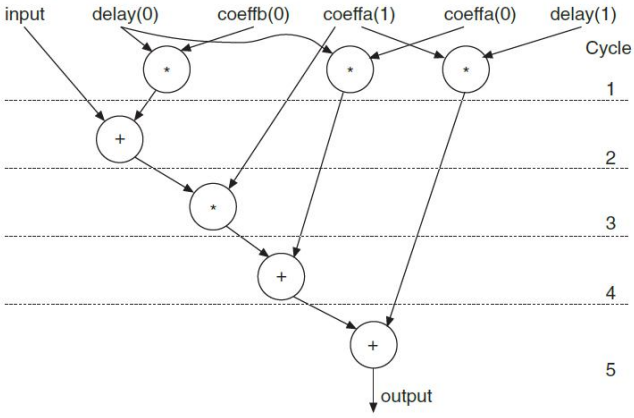

Figure 2: ASAP schedule in data dependency graph

The sequence of operations may not be same as given by the VHDL description. Equivalently, as late as possible (ALAP) schedule can also be applied with five clock cycles as shown in figure 3.

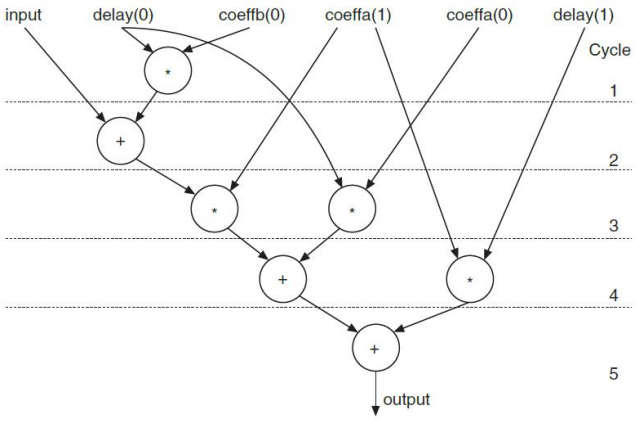

Figure 3: ALAP schedule in data dependency graph

These scheduling techniques are employed using VHDL descriptions so that the IIR filter can be implemented in hardware with less resource utilization [11] – [16]. The design is further divided into various resources with different stages of clock cycles. For example, two multipliers and one adder are being considered just for simplifying the design. Proper operations allocation or mapping onto resources is required.

3. RESOURCE SCHEDULING

The number of cycles may increase if the resources can be constrained to a single arithmetic unit during scheduling. Figure 5 shows the dependency graph with constrained resources. The number of possible scheduling increases with relatively larger problems [17] – [20]. The total design area increases while the speed and power consumption improve. This approaches can trade speed and power consumption against area by changing the schedule.

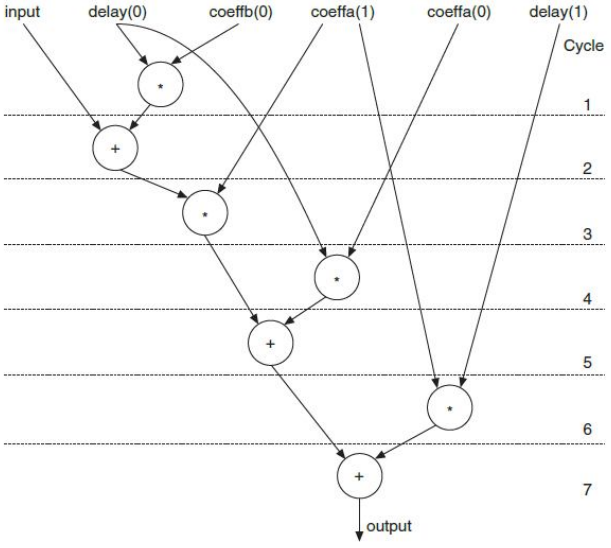

Figure 4: Resource constrained schedule.

Mapping of operations onto particular resources is done in the figure 5. The shaded portions represent resources (two multipliers and one adder) in respective clock cycles.

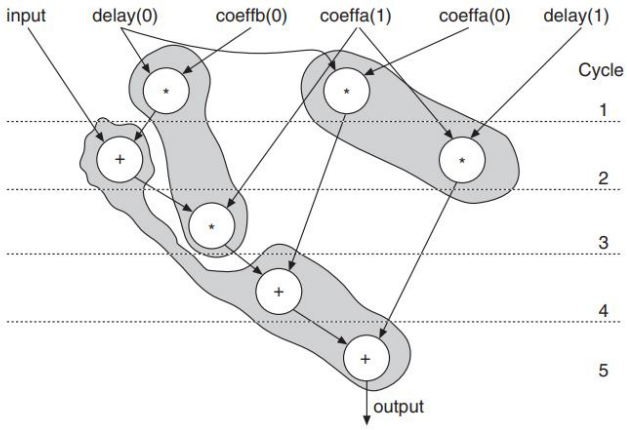

Figure 5: Mapping of operations onto resources

4. HARDWARE IMPLEMENTATION OF IIR FILTER

FPGA can be selected for the hardware implementation pf the proposed algorithms [10], [21] – [22]. Two multipliers and an adder are represented as resources by the three shaded groups in different clock cycles. In the subsequent clock cycle, the result of each operation will be stored [23] – [26]. Thus whenever a data arc crosses a clock boundary, a register must be inserted, refer figure 6. The resources in register can be shared just like arithmetic resources. In this regards, multipliers can be employed for sharing the resources. The possible hardware implementation is shown in figure 7.

The high pass 3rd order IIR filter is implemented with the down samples of sampling rate from 10 MHz to 79 KHz. The filter coefficients are A1 = -1, 93178; A2 = 0, 93403; B0 = 0, 96645; B1 = -1, 93291; B2 = 0,96645. This is implemented for four guard bits.

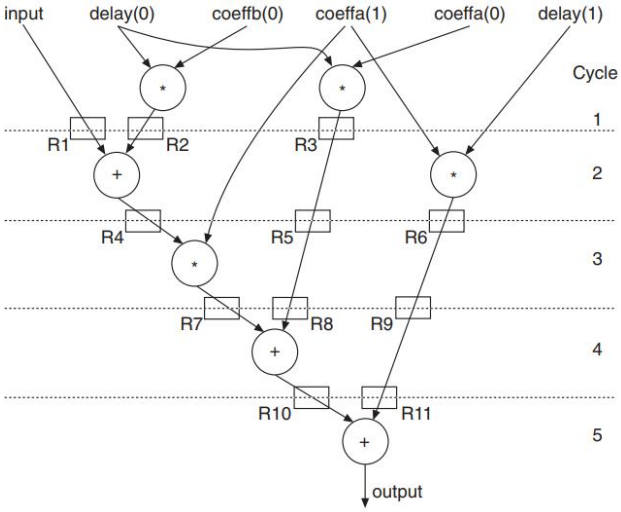

Figure 6: Registers scheduling

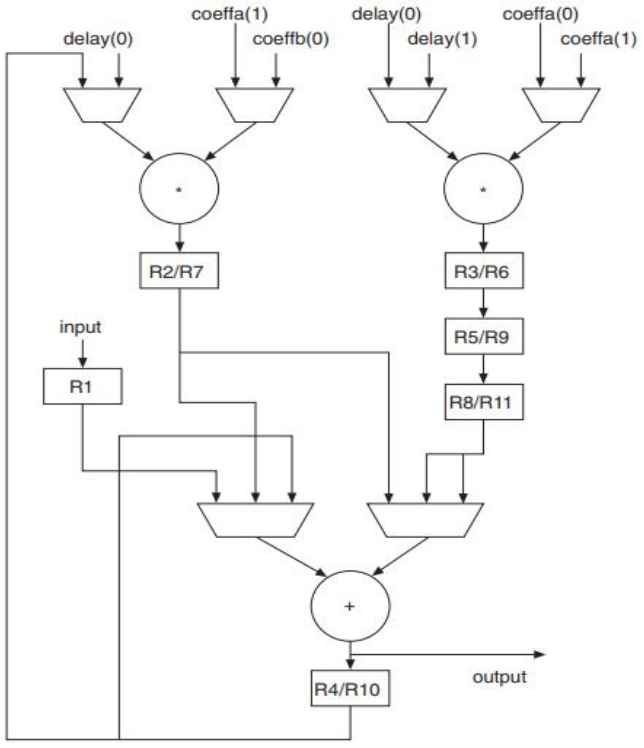

Figure 7: Hardware Implementation of 3rd order IIR filter

This FPGA implementation removes several level operations with proper resource mapping. This is one of the main factors for reducing the power consumption. The mapping of resources are converted to registers using multiplexers. The operations with high power consumptions are being replaced with the low power register mapping with every clock cycle.

5. RESULT ANALYSIS

The proposed algorithms were implemented in VHDL behavioural descriptions and run using Xilinx Vivado, FPGA technology Virtex 7. We run the scheduling algorithm for the resource constraints @ voltage of 3.3 V, 2 multipliers and 1 adder. The proposed algorithm also reduces the number of levels for registers, thus minimizing the power consumption. Figure 8 shows the simulation result of the high pass 3rd order IIR filter. However, the frequencies below 1 KHz still appear in the signal.

Table 1 shows the comparative analysis of the power consumption for 3rd order IIR filter. This result validates that the proposed algorithms are very effective in implementing the behavioural descriptions of IIR filter in FPGA. $E_s$is the power supply at 5V.

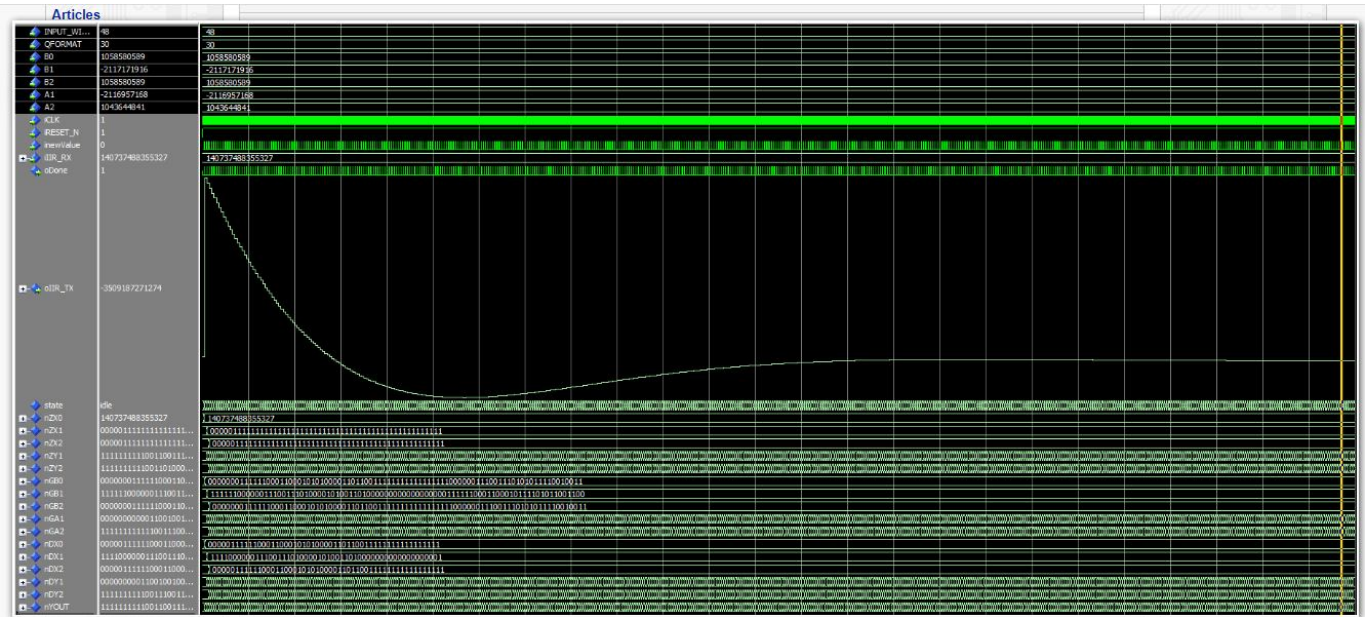

Figure 8: Simulation result of FPGA implementation of high pass IIR filter

Table 1: Power consumption analysis for 3rd order IIR filter

Scheduling algorithm	Power consumption (pJ)	% Improvement
[11]	10092	43.76
[12]	8092	15.27
Proposed	7020	-

6. CONCLUSION

These proposed algorithms minimizes the power consumption of the FPGA implementation of the IIR filter by employing the scheduling techniques of operation mapping onto resources. The experimental results show that there is an improvement of 15.27% in the power consumption at resource constraints of 3.3 V for behavioural synthesis of IIR filter. In future the design can be improved by eliminating the frequencies below 1 KHz after sampling.

ACKNOWLEDGEMENT

We thank the Department of Electronics and Communication Engineering – DST-FIST sponsored, KoneruLakshmaiah Education Foundation (KL University), Vaddeswaram, Indiafor allowing us to utilize the laboratories and complete this research work in time.

REFERENCES

1. Dixit. H.V & Gupta D.V. **IIR filters using Xilinx System Generator for FPGA implementation**, *International Journal of Engineering Research and Applications*, Vol. *2*(5), pp. 303-307, September 2012.
2. Mahabub A. **Design and implementation of cost-effective IIR filter for EEG signal on FPGA**. *Australian Journal of Electrical and Electronics Engineering*, pp. 1-9, June 2020.
3. Seshadri R., & Ramakrishnan, S. **FPGA implementation of fast digital FIR and IIR**

filters. *Concurrency and Computation: Practice and Experience*, e5246, 2019

4. YarragudiMadhuSudhana Reddy, RamaswamiSachidanandan Ernest Ravindran, Kakarla Hari Kishore**. Spatial Mutual Relationship Based Retinal Image Contrast Enhancement for Efficient Diagnosis of Diabetic Retinopathy**. International Journal of Intelligent Engineering and Systems, E-ISSN No: 2185-3118, Vol No: 11, Issue No: 5, pp 48-58, August 2018
5. Madanayake A, Bruton L., &Comis C. (2004, May). **FPGA architectures for real-time 2D/3D FIR/IIR plane wave filters**. In *2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512)*, Canada, 2004, pp. III-613.
6. K Hari Kishore, FazalNoorbasha, Katta Sandeep, D.N.V. Bhupesh, SK. Khadar Imran, K. Sowmya. **Linear convolution using UT Vedic multiplier**. International Journal of Engineering and Technology (UAE), Vol. 7(2.8), pp. 409-418, March 2018.
7. Islam, S.M.R., Sarker R., Saha S., & Uddin, A. N. **Design of a programmable digital IIR filter based on FPGA**. In 2012 International Conference on Informatics, Electronics & Vision (ICIEV),Dhaka, 2012, pp. 716-721.
8. K Hari Kishore, B. K. V. Prasad, Y. Manoj Sai Teja, D. Akhila, K. Nikhil Sai, P. Sravan Kumar. **Design and comparative analysis of inexact speculative adder and multiplier.**International Journal of Engineering and Technology (UAE), Vol. 7(2.8), pp. 413-426, March 2018.
9. G SiriVennela, K Hari Kishore, E Raghuveera. **High Accurate and Power Efficient ECG-Based Processor for Predicting Ventricular Arrhythmia**. Journal of Advanced Research in Dynamical and Control Systems, Vol. 10(2), pp. 1180-1121, May 2018.
10. Poučki, V. M., Žemva, A., Lutovac, M. D., &Karčnik, T. **Elliptic IIR filter sharpening implemented on FPGA**. Digital Signal Processing, Vol. *20*(1), pp.13-22, January 2010
11. W.T. Shiue and C. Chakrabarti**. Low power scheduling with resources operating at multiple voltages**. IEEE Trans. Circuits and Systems II: Analog and Digital Signal Processing, Vol.47, pp.536-543, June 2000.
12. Yang H.C, & Dung L. R. **Algorithmic transformations and peak power constraint applied to multiple-voltage low-power VLSI signal processing**. WSEAS Transactions on Signal Processing, Vol. *3*(12), pp. 479-486, December 2007.
13. Ngangbam Phalguni Singh, Hamisi, N. Y., &Iddi, H. U. **DDFF based Low Power Dissipation in Ring Counter using Octree**. International Journal of Applied Engineering Research, Vol. 12(19), pp. 8346-8353, October 2017.
14. ChellaSanthosh, K. Hari Kishore, G. Pavani Lakshmi, G.Kushwanth, P. Rama Krishna Dharma Teja, R. S. Ernest Ravindran, SreeVardhanCheerala, M. Ravi Kumar. **Detection of Heavy Metal Ions using Star-Shaped Microfluidic Channel**. International Journal of Emerging Trends in Engineering Research, Vol. 7 (12), pp. 768-771, December 2019.
15. Ngangbam Phalguni Singh, Hamisi, N. Y., &Iddi, H. U. **AC Analysis of Octree based Buffer Using Various Data Conditioning Flip Flops.** *International Journal of Applied Engineering Research, 12*(19), pp. 8366-8370, October 2017
16. Sk.Md.Reyaan, B.ManojVenkat, G.Y.V.Siva Shankar, K Hari Kishore, FazalNoorbasha, Y Archana, Shaik Razia. **Power Analysis of FIR Filter Design Using APC-OMS Algorithm**.Journal of Advanced Research in Dynamical and Control Systems, Vol. 12(2), pp. 1085-1092, March 2020.
17. Bhogadi Anil Kumar, ChillapalliHaritha, Gumpena Veda Sri Leela, E Raghuveera, K Hari Kishore. **A Parametric DFT Scheme for RAMs**.Journal of Advanced Research in Dynamical and Control Systems, Vol.12 (2), pp. 2298-2305, May 2020.
18. KDivyaMadhuri, K Hari Kishore. **Implementation of 4-bit Ripple Carry Adder by Adopting Sub threshold Adiabatic Logic for Ultralow-Power Application**.Journal of Advanced Research in Dynamical and Control Systems, Vol. 12(6), pp. 11-17, May 2020.
19. A Murali, K Hari Kishore. **An Efficient Debugging Architecture for DTG Based FIR Filter Using I2C Protocol in DSP Processor**. International Journal of Emerging Trends in Engineering Research, Vol. 8(7), pp. 3468-3477, August 2020.
20. AvinashYadlapati, Hari Kishore Kakarla. **Design and Verification of Asynchronous FIFO with Novel Architecture Using Verilog HDL**.Journal of Engineering and Applied Sciences, Vol. 14(1), pp. 159-163, January 2019.
21. AvinashYadlapati, K Hari Kishore**. System Level Verification of Advanced Extensible Interface Protocol Using Verilog HDL.**Journal of Advanced Research in Dynamical and Control Systems, Vol. 10(7), pp. 1359-1365, June 2018.
22. B. Srikanth, M. Siva Kumar, J.V.R. Ravindra, K. Hari Kishore.**Double Precession Floating Point Multiplier using Schonhage-Strassen Algorithm used for FPGA Accelerator**.International Journal of Emerging Trends in Engineering Research, Vol. 7(11), pp. 677-684, December 2019.
23. B. Murali Krishna, K. Ch. Sri Kavya, P.V. S. Sai Kumar, K. Karthik, Y. Siva Nagababu. **FPGA Implementation of Image Cryptology using Reversible Logic Gates**. International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9(3), pp. 2522-2526, June 2020.
24. AntipasT. Teologo Jr., Reggie C. Gustilo. **Performance Analysis of MUSIC Algorithm for Microphone Array Using FPGA board.**International Journal of Advanced Trends in Computer Science and Engineering, Vol. 9(3), pp. 3478 –3484, June 2020.
25. Jagdeep Kaur Sahani, ShrutiSuman, P. K. Ghosh, Double **Gate based OTAs for Analog Filters: Design of**

Analog Filter. *LAP LAMBERT Academic Publishing*, ISBN-10: 3659565407 ISBN-13: 978-3659565, 2014.

26. Jagdeep Kaur Sahani, ShrutiSuman, P. K. Ghosh, **Design of Second Order Low Pass and High Pass Filter using Double Gate MOSFET based OTA,** *Innovative Systems Design and Engineering*, Vol 5, Issue 4, pp.111-122, 2014.