Implementation of High Speed FIR Filter using Serial and Parallel Distributed Arithmetic Algorithm

Narendra Singh Pal
Department of Electronics &Communication, Dr.B.R.Ambedkar National Institute of Tecnology, Jalandhar (Punjab)

Harjit Pal Singh
Department of Physics Dr.B.R.Ambedkar National Institute of Tecnology, Jalandhar (Punjab)

R.K.Sarin,
Department of Electronics & Communication Dr.B.R. Ambedkar National Institute of Technology Jalandhar

Sarabjeet Singh
Department of Physics Dr.B.R. Ambedkar National Institute of Technology Jalandhar (Punjab)

ABSTRACT
This paper describes the implementation of highly efficient multiplierless serial and parallel distributed arithmetic algorithm for FIR filters. Distributed Arithmetic (DA) had been used to implement a bit-serial scheme of a general symmetric version of an FIR filter due to its high stability and linearity by taking optimal advantage of the look-up table (LUT) based structure of FPGAs. The performance of the bit-serial and bit-parallel DA technique for FIR filter design is analyzed and the results are compared to the conventional FIR filter design techniques. The proposed algorithm has been synthesized with Xilinx ISE 10.1i and implemented as a target device of Spartan3E FPGA.

KEYWORDS
Distributed Arithmetic (DA), FIR filter, Look up table (LUT), FPGA

1. INTRODUCTION
Due to the intensive use of FIR filters in video and communication systems, high performance in speed, area and power consumption is demanded. Basically, digital filters are used to modify the characteristic of signals in time and frequency domain and have been recognized as primary digital signal processing [1]. In DSP, the design methods were mainly focused in multiplier-based architectures to implement the multiply-and- Accumulate (MAC) blocks that constitute the central piece in FIR filters and several functions [2]. The FIR digital filter is presented as:

$$y[n] = \sum_{k=0}^{N-1} c[k]x[n-k] \quad (1)$$

Where $y[n]$ is the FIR filter output, $x[n-k]$ is input data and $c[k]$ represents the filter coefficients Eq. (1) shows that multiplier-based filter implementations may become highly expensive in terms of area and speed. This issue has been partially solved with the new generation of low-cost FPGAs that have embedded DSP blocks [3]. The advantages of the FPGA approach to digital filter implementation include higher sampling rates than are available from traditional DSP chips, lower costs than an ASIC for moderate volume applications, and more flexibility than the alternate approaches.

In literature, several multiplier-less schemes had been proposed. These methods can be classified in two categories according to how they manipulate the filter coefficients for the multiply operation. The first type of multiplier-less technique is the conversion-based approach, in which the coefficients are transformed to other numeric representations whose hardware implementation or manipulation is more efficient than the traditional binary representation. Example of such techniques are the Canonic Sign Digit (CSD) method, in which coefficients are represented by a combination of powers of two in such a way that multiplication can be simply implemented with adder/subtractors and shifters [4], and the Dempster-Mcleod method, which similarly involves the representation of filter coefficients with powers of two but in this case arranging partial results in cascade to introduce further savings in the usage of adders [5]. The second type of multiplier-less method involves use of memories (RAMs, ROMs) or Look-Up Tables (LUTs) to store pre-computed values of coefficient operations. These memory-based methods involve Constant Coefficient Multiplier method and the very-well known Distributed Arithmetic method [6] as examples.

Distributed Arithmetic (DA) algorithm appeared as a very efficient solution especially suited for LUT-based FPGA architectures. Croisier et al [7] had proposed the multiplier-less architecture of DA algorithm and it is based on an efficient partition of the function in partial terms using 2's complement binary representation of data. The partial terms can be pre-computed and stored in LUTs. Yoo et al. [8] observed that the requirement of memory/LUT capacity increases exponentially with the order of the filter, given that DA implementations need 2^K – words, *K* being the number of taps of the filter.

The work in this paper presents the design and implementation of serial and parallel distributed algorithm for FIR filter target as Spartan 3E FPGA. The results of the implementation experiment are analyzed in terms of parameters such as area and speed. The brief description of the distributed algorithm is presented in Section 2. The implementation of the proposed technique for FIR filters is discussed in Section 3. The Section 4 presents the implementation results. The last section concludes the work and presents the future work.

2. DISTRIBUTED ARITHMETIC (DA)

Distributed arithmetic (DA) is an important FPGA technology. It is extensively used in computing the sum of products,

$$y[n] = < c, x > = \sum_{n=0}^{N-1} c[n]x[n] \quad (2)$$

DA system, assumes that the variable $x[n]$ is represented by-

$$x[n] = \sum_{b=0}^{B-1} x_b[n] \times 2^b, x_b[n] \in [0,1] \quad (3)$$

If $c[n]$ are the known coefficients of the FIR filter, then output of FIR filter in bit level form is:

$$y = \sum_{n=0}^{N-1} c[n] \times \sum_{b=0}^{B-1} x_b[n] \times 2^b \quad (4)$$

In distributed arithmetic form-

$$y = \sum_{b=0}^{B-1} 2^b \times \sum_{n=0}^{N-1} f(c[n], x_b[n]) \quad (5)$$

In Eq. (5) second summation term realizing as one LUT. The use of this LUT or ROM eliminates the multipliers [9].

For signed 2's complement number output of FIR filter can be computed as-

$$y = -2^B \times f(c[n], x_B[n]) + \sum_{b=0}^{B-1} 2^b \times \sum_{n=0}^{N-1} f(c[n], x_b[n]) \quad (6)$$

Where *B* represents the total number of bits used. Fig 1 shows the Distributed architecture for FIR filter and different with the MAC architecture.

When $x[n]$ <0, Binary representation of the input is [10],

$$x[n] = -x_b[0] + \sum_{b=1}^{B-1} x_b[n] 2^{-b} \quad (7)$$

The output in distributed arithmetic form-

$$y = -(\sum_{n=0}^{N-1} c[n] x_b[0]) + \sum_{b=1}^{B-1} (\sum_{n=0}^{N-1} c[n] x_b[n]) 2^{-b} \quad (8)$$

If the number of coefficients *N* is too large to implement the full word with a single LUT (Input LUT bit width = number of coefficients), then partial tables can be and add the results as shown in Fig2. If pipeline registers are also added, then this modification will not reduce the speed, but can dramatically reduce the size of the design [8].

2.1 Parallel Distributed Arithmetic Architecture

A basic DA architecture, for a length N^{th} sum-of-product computation, accepts one bit from each of N words. If two bits per word are accepted, then the computational speed can be essentially improved. The maximum speed can be achieved with the fully pipelined word-parallel architecture as shown in Fig 3. For maximum speed, a separate ROM (with identical content) for each bit vector x_b[n] should be provided [11].

3. IMPLEMENTATION

To evaluate the performance of the Distributed Arithmetic serial and parallel scheme for symmetric FIR filters are implemented and synthesized using Xilinx ISE 10.1 Target as a Spartan 3E (Xc3s100c-5vq100) FPGA device and the results are compared to conventional FIR filter. ISE design software offers a complete design suit based programmable logic devices on Xilinx ISE. The design can be simulated and synthesized in the form of schematic or HDL entry on Xilinx ISE platform. Spartan3E FPGA can be programming directly from Xilinx ISE in configuration logic blocks interconnected with switching matrix. Spartan 3E has a microblaz DSP processor of 325 MHz operating frequency, so that DSP design can be implemented for less resources, high speed and low power. The designed FIR filter is programmed in verilog HDL language [12]. The proposed design is implemented for small memory location LUT and also for large memory location LUT to analyze the performance of the proposed design for speed and area parameters. In the present work, the proposed design is analyzed through 3-tap and 16 –tap DA FIR filters.

3.1 Design Procedure:

Step1: Derive the filter coefficient according to specification of filter.

Step2: Store the inputs value in input register

Step3: Design the LUTs as shown in Fig.4, which represents all the possible sum combination of filter coefficients.

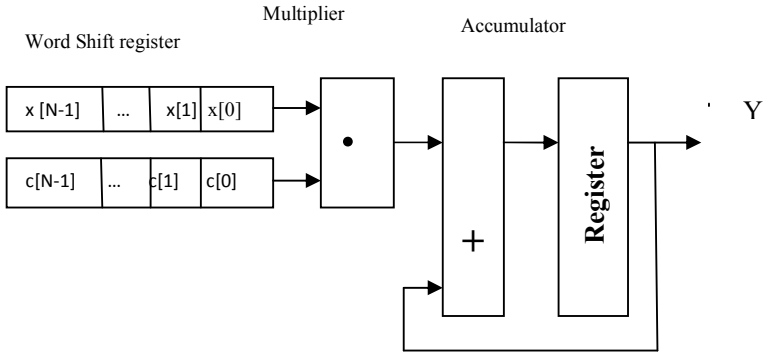
Step4: Accumulate and shift the value according to partial term beginning with LSB of the input and shift it to the right to add it to the next partial result.

Step5: First value must be subtracted, due to negative bit of MSB.

Step6: Analyze the output of filter as per specifications, otherwise go to step 1.

Step7: The same procedure is applied for Parallel DA FIR from step 1 to step 6 except in Step3 where bit address value is being called for 2-bits at single time, so that two times LUT is required in comparison to the serial DA.

(a)



(b)

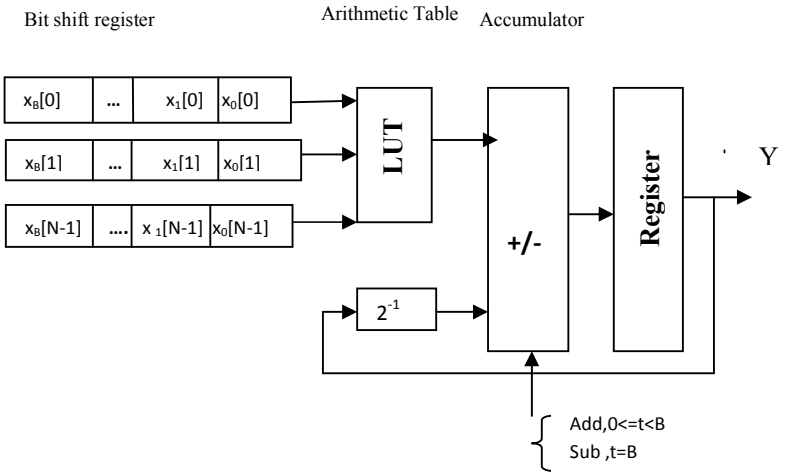


Fig 1: (a) Conventional MAC and (b) DA Architecture

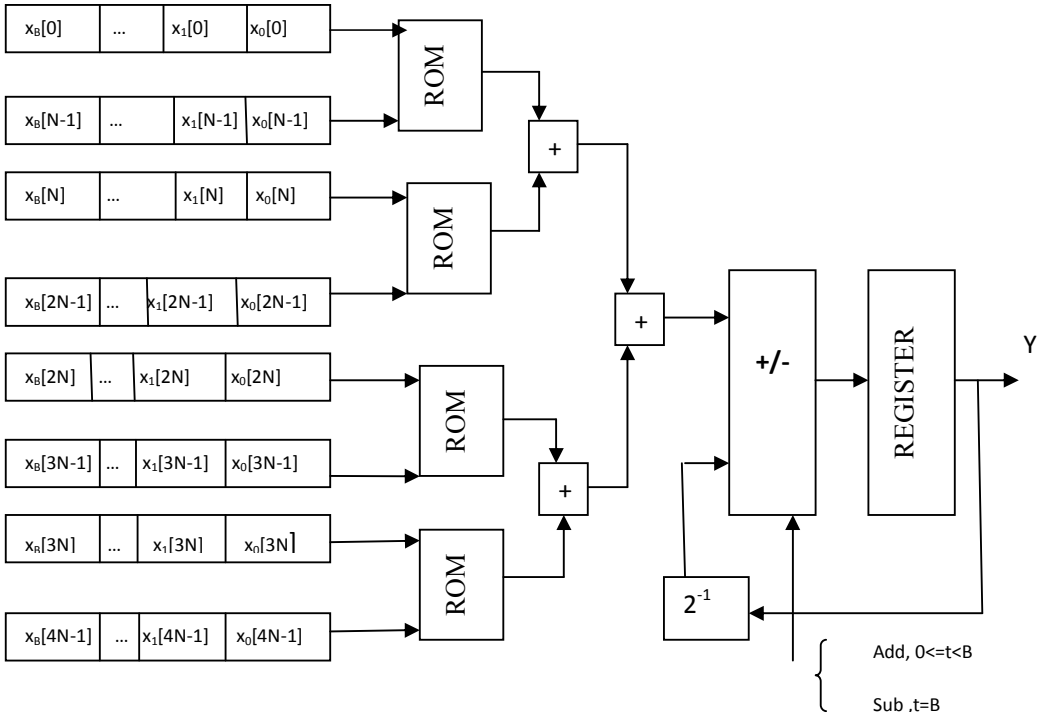


Fig 2: DA with Table partitioning to yield a reduced Size

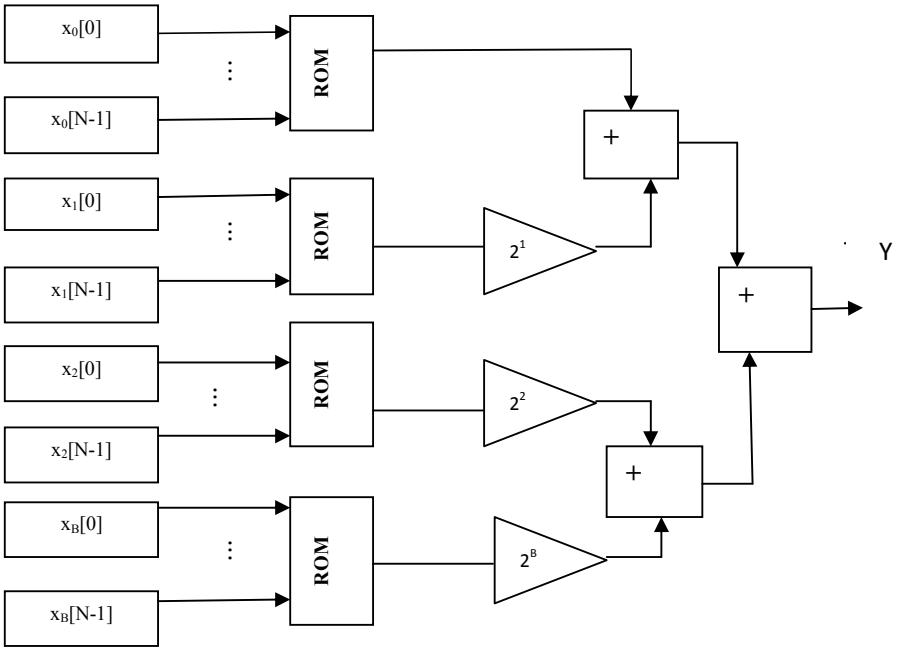


Fig 3: Parallel Distributed Arithmetic Architecture

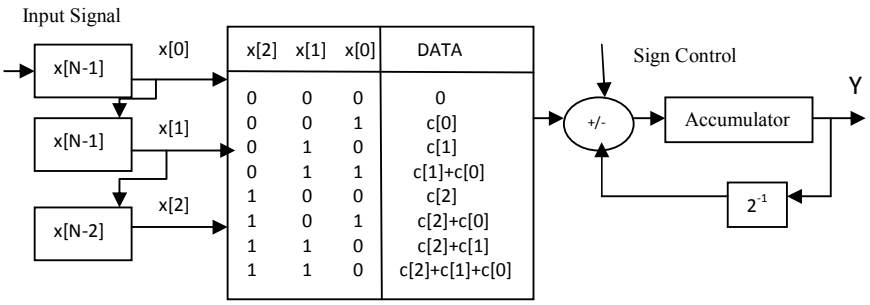


Fig 4: 2^3xB LUT based DA FIR filter

4. IMPLEMENTATION RESULTS

The 3-tap and 16-tap FIR filters are implemented for performance analysis of the proposed DA algorithm. The 3-tap DA FIR filter required very small area, since only single LUT is used for implementation due to its 2^3 memory locations. The control unit clears buffers and then the input collected by the input registers during the previous clock cycles is serially injected to the circuit. These bits address a value in the input LUT structure, and this partial result is accumulated and shifted by the shifter/adder unit, taking into account that the very first value must be subtracted. Number of clock cycle depends on the number of bits in the input. The test bench simulation results for 3 tap DA based FIR filter are shown in Fig 5(a) and Fig 5(b). The RTL view of 3-tap serial and parallel DA FIR filter, which shows the flow from input to output, are presented in Fig 6(a) and Fig.6 (b).

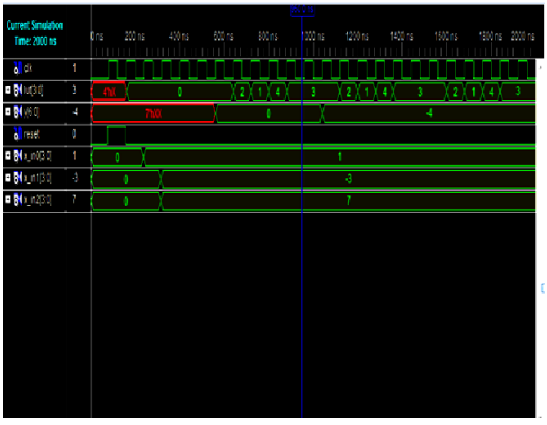


Fig 5 (a): Simulation for 3-tap serial DA FIR filter

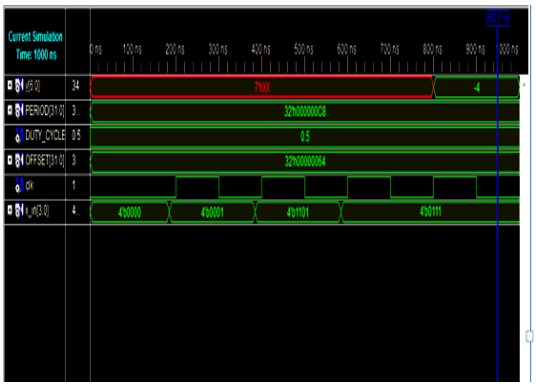


Fig 5 (b): Simulation for 3-tap parallel DA FIR filter

Fig 6 (a): RTL view of 3-tap serial DA FIR filter

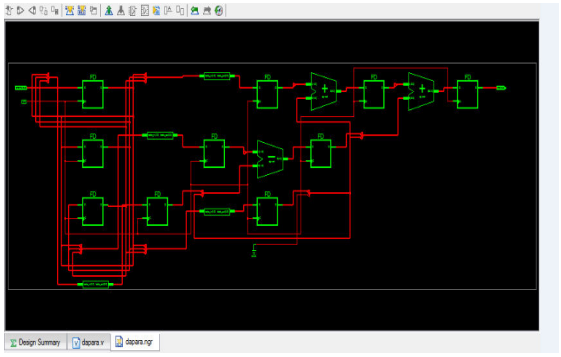


Fig 6 (b): RTL view of 3-tap serial DA FIR filter

For 16-tap DA FIR filter, partitioning the input into 4 bit units, reduces the single look up table 2^{16} memory locations to 4×2^4 LUTs. The Logic flow is similar as shown in Fig.4, excepting that there are 4 partial results each from one of the each basic 4 input LUT cells in this case. The addition of these partial results gives the value of output and this output is sent to accumulate where shifting for next value occurs as shown in Fig 2. The precision of n-bit input, the partial results accumulated and shifted n times with new value addressed in 4 input LUT units. Finally in $(n+1)^{th}$ clock cycle signal from control indicate latch structure to output the final result, which will be shown in every $(n+2)^{th}$ clock cycle. In parallel distributed arithmetic, two bits send for a single word so an extra LUT is required for second bit. For 16 tap FIR filter total number of LUT required is 8. Each contains 2^4 memory address location. Fig.7 (a) and Fig.7 (b) show the simulation result for 16 tap DA based FIR filters for serial and parallel DA algorithm resp. The register transfer level (RTL) views for 16 tap DA FIR filters that shows there is no multiplier use in the architecture of distributed arithmetic FIR filters as presented in Fig 8.

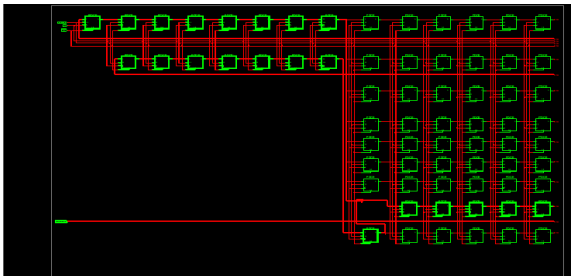


Fig 8 (a): RTL view of 16-tap serial DA FIR filter

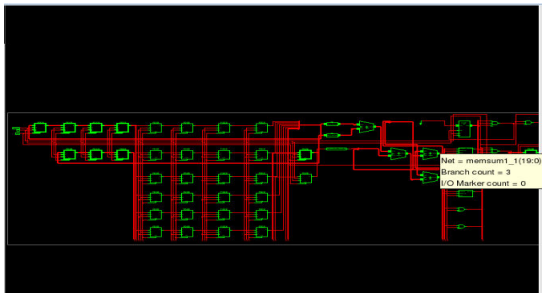


Fig 8 (b): RTL view of 16-tap serial DA FIR filter

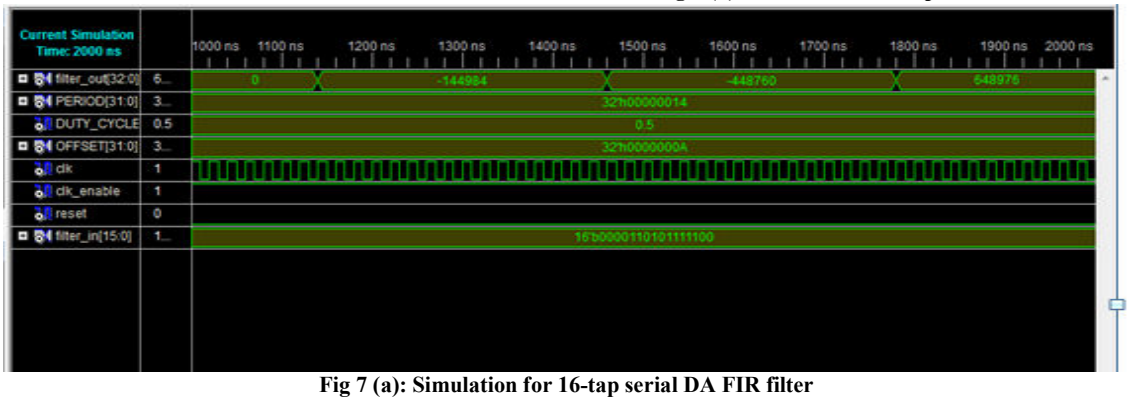


Fig 7 (a): Simulation for 16-tap serial DA FIR filter

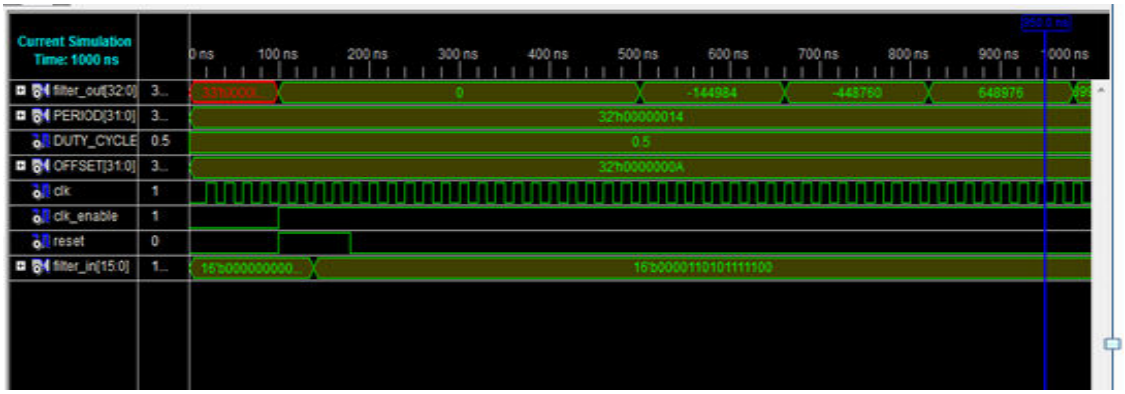


Fig 7 (b): Simulation for 16 tap parallel DA FIR filter

4.1 Discussion

The implementation results of 3-tap and 16-tap FIR filter after applying the distributed arithmetic algorithm as shown in Table 1 and 2. The 3-tap parallel DA FIR filter take high speed and lowest power dissipation in comparison to serial DA FIR filter and conventional FIR filter as shown in Table 1. For small tap filters, the serial DA algorithm saves 50 % of the area and cost in comparison to the conventional design techniques. The speed is approximately 2 times for serial DA and 3 times in parallel DA is achieved and very less power is consumed in comparison to simple FIR filter.

Table 1. Synthesis and simulation result for 3 tap FIR filter

Parameter	Direct FIR Filter	Serial DA FIR Filter	Parallel DA FIR filter
Slices	38	25	26
Slice Flip-flops	32	30	44
4 input LUTs	57	47	30
Delay (ns)	13.116	6.413	4.014
Power (mW)	1.5220	1.3134	1.1965

TABLE 2 shows that after partitioning the input bits and increasing the number of LUTs in 16 tap DA FIR filter, parallel DA FIR filter is area efficient in comparison to conventional FIR filter but at slight compromise of power which is equivalent to Conventional FIR, Parallel DA FIR gives high speed about 3 times faster than conventional FIR filter. Whereas in serial DA FIR filter saved resources area is greater than 50% and speed is 2 times in comparison to conventional FIR filter. Fig 9 and Fig 10 shows the layout of serial DA and Parallel DA FIR Filter in Cadence SOC Encounter 180nm which represent the floor-plan and core area of the digital circuit.

Table 2. Synthesis and simulation result for 16 tap FIR filter

parameter	Conventional FIR	Serial DA FIR	Parallel DA FIR
Slices	412	180	243
Flip-Flop	305	171	192
4 input LUTs	503	263	407
Delay(ns)	46.011	19.2385	15.725
Power(mW)	10.3478	12.1451	11.7805

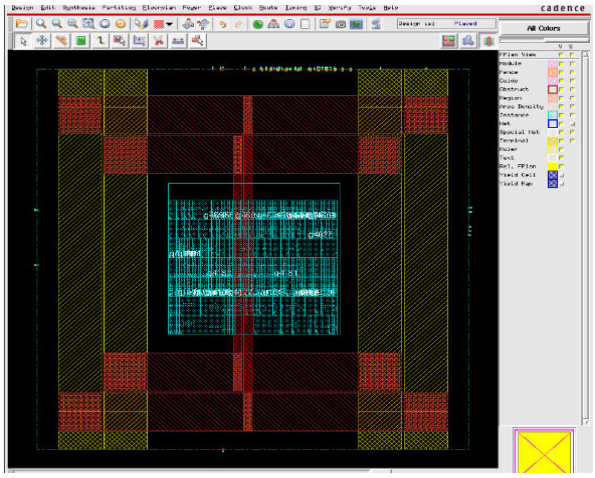


Fig 9: Lay out for 16 tap serial DA FIR filter

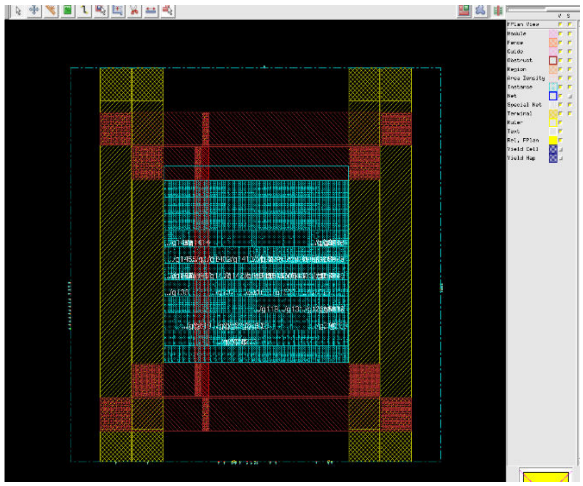


Fig 10: Lay out for 16 tap Parallel DA FIR filter

4. CONCLUSION & FUTURE WORK

The implementation of highly efficient serial and parallel DA algorithm was presented in this work. The results were analyzed for 3-tap and 16-tap FIR filter using partitioned input based LUT on Xilinx 10.1i as a target of SPARTAN-3E FPGA device .The speed performance of the Parallel DA FIR Filter was superior in comparison to all other techniques . For small tap filter less area, high speed and low power consumption is achieved after applying the Serial and Parallel DA technique. In large-tap FIR filter, speed of parallel DA FIR design technique become 3 times faster than that of conventional FIR filter. The proposed algorithm for FIR filters is also area efficient since approximately 50% of the area is saved with this technique as compared to conventional FIR filter design. Area efficiency and high speed is achieved with parallel DA technique at very slight cost of power consumption for large tap FIR filter. Since, distributed arithmetic FIR filters are area efficient and contained less delay, so these filters can be used in various applications such as pulse shaping FIR filter in WCDMA system, software design radio and signal processing system for high speed. In future the work to reduce the power consumption in large tap DA FIR filters could be performed.

5. REFERENCES

[1] Mohamed al mahdi Eshtawie and Masurie Bin Othman,"An Algorithm Proposed For FIR Filter Cofficent Representation" International Journel of Mahematics and computer Sciences 2008.pp24-30.

[2] John G. Prokis, Manolakis, "Digital Signal Processing" Principles ,algorithm and applications (Fourth Edition)-2008

[3] Antolin Agatep, "Xilinx Spartan-II FIR Filter Solution", WP116 (v1.0) April 5, 2000

[4] M. Yamada, and A. Nishihara, "High-Speed FIR Digital Filter with CSD Coefficients Implemented on FPGA", in Proceedings of IEEE Design Automation Conference , 2001, pp. 7-8.

[5] M.A. Soderstrand, L.G. Johnson, H. Arichanthiran, M. Hoque, and R. Elangovan, "Reducing Hardware Requirement in FIR Filter Design", in Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing 2000, Vol. 6, pp. 3275 – 3278

[6] Martinez-Peiro, J. Valls, T. Sansaloni, A.P. Pascual, and E.I. Boemo, "A Comparison between Lattice, Cascade and Direct Form FIR Filter Structures by using a FPGA Bit-Serial DA Implementation", in Proceedings of IEEE International Conference on Electronics, Circuits and Systems, 1999, Vol. 1,pp. 241 – 244.

[7] A. Croisier, D. J. Esteban, M. E. Levilion, and V. Rizo, "Digital Filter for PCM Encoded Signals", U.S. Patent No. 3,777,130, issued April, 1973

[8] H. Yoo, and D. Anderson, "Hardware-Efficient Distributed Arithmetic Architecture for High-Order Digital Filters", in Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005**,** Vol. 5, pp. 125 – 128.

[9] T.Vigneswarn and P.Subbarami Reddy"Design of Digital FIR Filter Based on DDA algorithm" Journal of Applied Science ,2007

[10] Stanley A. White,"Application of Distributed Arithmetic to Digital Signal Processing: A Tutorial Review" IEEE Acoustic speech signal processing Magazine, July 1989

[11] Attri, S.; Sohi, B.S.; Chopra, Y.C.**;** "Efficient design of application specific DSP cores using FPGAs" in Proceedings of 4th IEEE International Conference on application specific integrated circuits Oct. 2001 Page(s):462 – 466

[12] Samir Palnitkar,"Verilog HDL A guide to Digital Design and Synthesis"Second Edition-2007