

Implementation of Efficient Serial and Parallel Distributed Arithmetic Algorithm for FIR Filter

B.kiran kumar

Assistant Professor, Lords Institute of Engineering & Technology, Hyderabad.

Abstract:-this paper describes the implementation of highly efficient 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. Traditional parallel filter implementations cause linear increase in the hardware cost with respect to the block size. Recently, an efficient parallel FIR filter implementation technique requiring a less-than linear increase in the hardware cost was proposed. Modified FIR filter design using distributed Arithmetic (DA) also provides an approach for multiplier-less implementation of DSP systems. It can completely replace all multiplications and additions by a look up table (LUT) and a shifter-accumulator thereby it can save considerable amount of hardware resources. The proposed algorithm has been synthesized with Xilinx ISE 10.1i and implemented as a target device of Spartan3E FPGA.

Keywords- *Distributed Arithmetic (DA); FPGA; look-up table(LUT),multiply-and-Accumulate(MAC),Serial,Parallel.*

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:

$$y[n] = \sum_{k=0}^{L-1} w[k]x[n - k]$$

Where $y[n]$ is the FIR filter output, $x[n - k]$ is input data and $w[k]$ 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 multipliers-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/subtractions 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 $2K$ – 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.

1. Distributed Arithmetic (DA)

Raditional implementations of the finite impulse response (FIR) filter equation is

$$y[n] = \sum_{k=0}^{L-1} w[k]x[n - k]$$

typically employ L multiply-accumulate (MAC) units. Implementing multipliers using the logic fabric of the FPGA is costly due to logic complexity and area usage, especially when the filter size is large. Modern FPGAs have dedicated DSP blocks that alleviate this problem, however for very large filter sizes the challenge of reducing area and complexity still remains. An alternative to computing the multiplication is to decompose the MAC operations into a series of lookup table (LUT) accesses and summations.

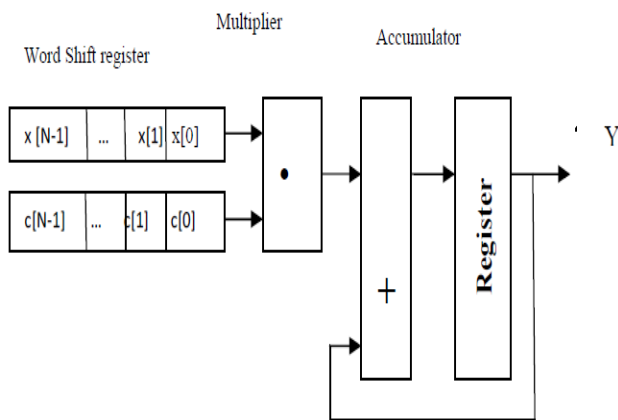


Fig: conventional MAC.

2. Implementation of FIR filter using Distributed Arithmetic (Existing)

The DA solves the computation of the inner product equation when the coefficients are pre knowledge, as happens in FIR filters.

An FIR filter of length K is described as

$$y = \sum_{t=1}^T A_t x_t \quad \text{-----(1)}$$

In this equation (1) the A_t are the fixed coefficients, T is the number of filter-taps and x_t are the input data words. These ones have a standard fixed-point format number, which is a two's-complement fractional representation with x_t limited in the range $-1 \leq x_t \leq 1$:

$$x_t = -x_{t0} + \sum_{n=1}^{N-1} x_{t,n} 2^{-n}$$

where N is the bit number of the data, being x_{t0} the sign bit and $x_{t,n}$ the bit "n" of x_t .

$$y = \sum_{t=1}^T A_t (-x_{t0} + \sum_{n=1}^{N-1} x_{t,n} 2^{-n}) = -\sum_{t=1}^T A_t x_{t0} + \sum_{n=1}^{N-1} [\sum_{t=1}^T A_t x_{t,n} 2^{-n}]$$

$$y = -[A_1 x_{1,0} + A_2 x_{2,0} + A_3 x_{3,0} + \dots + A_T x_{T,0}] + [A_1 x_{1,1} + A_2 x_{2,1} + A_3 x_{3,1} + \dots + A_T x_{T,1}] \cdot 2^{-1} + [A_1 x_{1,2} + A_2 x_{2,2} + A_3 x_{3,2} + \dots + A_T x_{T,2}] \cdot 2^{-2} \vdots + [A_1 x_{1,N-1} + A_2 x_{2,N-1} + A_3 x_{3,N-1} + \dots + A_T x_{T,N-1}] \cdot 2^{-(N-1)}$$

where $x_{1,0}$ is the bit 0 of the first sample, $x_{2,0}$ is the bit 0 of the second sample, etc... Each term in brackets is located into a ROM, RAM or Look-Up Table (LUT) memory. Their $2(N-1)$ different values are addressed by $x_{t,n}$.

Address					Address Content
$x_{t,N-1}$...	$x_{t,2}$	$x_{t,1}$	$x_{t,0}$	
0	...	0	0	0	0
0	...	0	0	1	A_1
0	...	0	1	0	A_2
0	...	0	1	1	$A_2 + A_1$
...
1	...	1	1	1	$A_1 + A_2 + A_1$

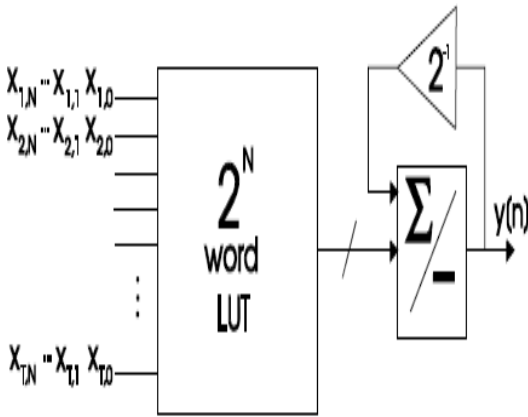


Fig: Block Diagram representation of the Existing Distributed FIR is shown in the following figure.

The architectural implementation of Distributed arithmetic is shown in the following figure. The blocks presented in this architecture are Shift register block, LUT unit, add and shift (accumulator) unit.

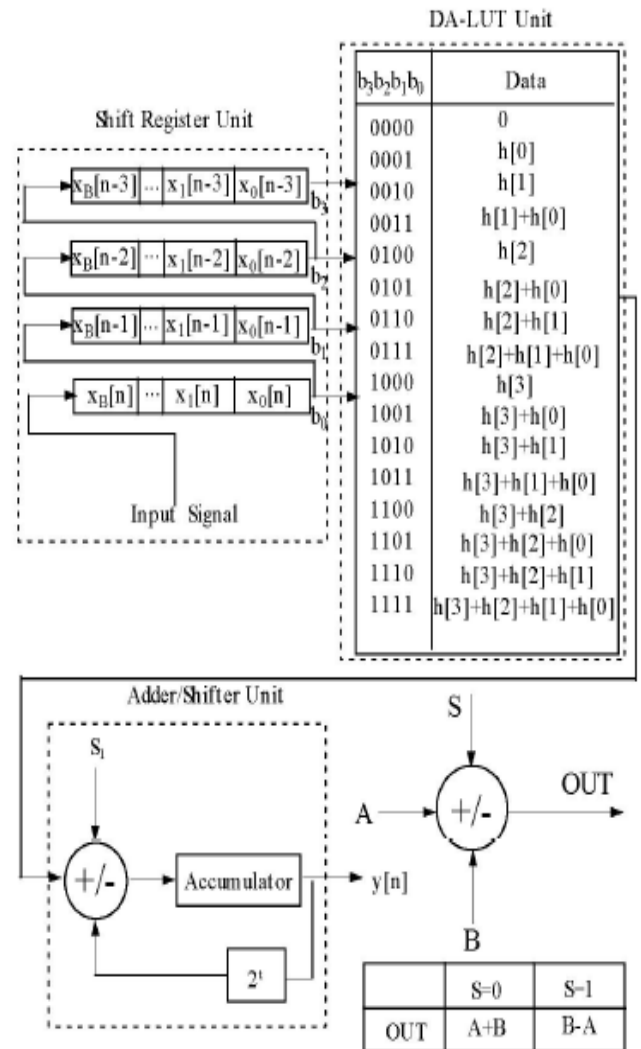


Figure1. Original LUT-based DA implementation of 4-tap FIR filter

3. Parallel Distributed Arithmetic Architecture

A basic DA architecture, for a length Nth 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].

4. 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 programmed directly from Xilinx ISE in configuration logic blocks interconnected with switching matrix. Spartan 3E has a microbar 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.

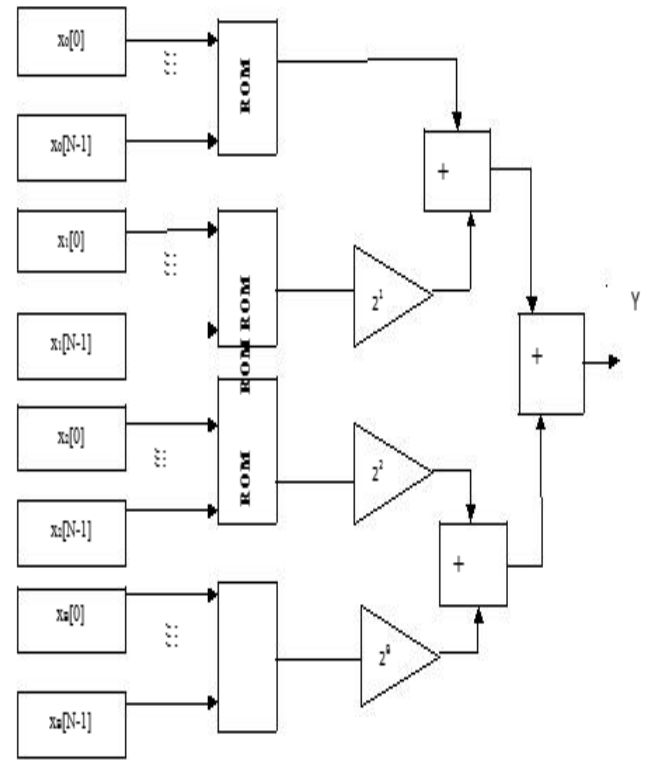


Fig 3: Parallel Distributed Arithmetic Architecture

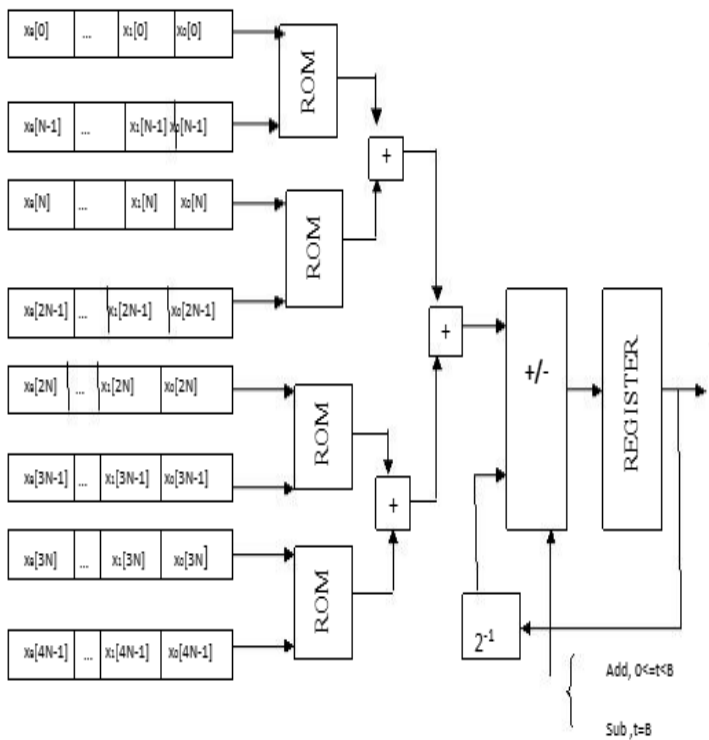


Fig 2: DA with Table partitioning to yield a reduced Size

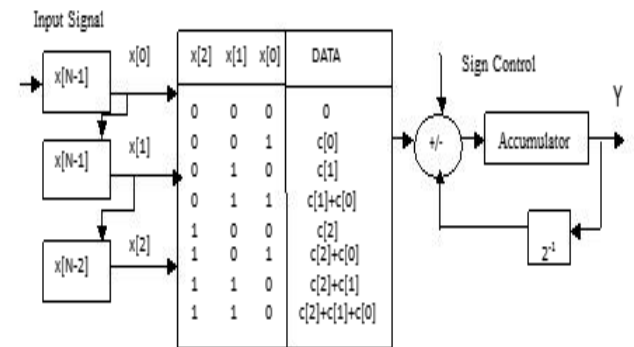


Fig 4: 2³xB LUT based DA FIR filter

5. 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 23 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.

For 16-tap DA FIR filter, partitioning the input into 4 bit units, reduces the single look up table 216 memory locations to 4x24 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 24 memory address location.

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

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

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

6. 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,4-tap and 16-tap FIR filter using partitioned input based LUT on Xilinx 10.1i as a target of SPARTAN-3EFPGA 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.

7. REFERENCES

[1] Mohamed al mahdi Eshtawie and Masurie BinOthman, "An Algorithm Proposed For FIR Filter Coefficient Representation" International Journal of Mathematics and computer Sciences 2008, pp24-30.
 [2] John G. Proakis, 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.Vigneswari 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.

[13] A Reusable Distributed Arithmetic Architecture for FIR Filtering Haw-Jing Lo, Heejong Yoo, and David V. Anderson

[14] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters," IEEE Transactions on ASSP, vol. 22, no. 6, pp. 456–462, 1974.

[15] T.-S. Chang and C.-W. Jen, "Hardware-efficient Implementations for Discrete Function Transforms using LUT-based FPGAs," IEE Proceedings –Computers and Digital Techniques, vol. 146, no. 6, pp. 309–315, 1999.



B.KIRAN KUMAR completed B.Tech in 2006 from (MTIST) JNTUH & M.Tech in 2010 from (AEC)JNTUH. Having 9 years of teaching Experience. Field of interest is VLSI System Design, wireless communication, Digital Electronics. Presently working as Assistant Professor in Department of Electronics and Communication Engineering, LORDS Institute of Engineering and Technology, Hyderabad.