

Flexible Medium Speed Systolic Architecture for FIR Filter Based on Distributed Arithmetic

¹K.G. Shanthi, ²J. Faritha Banu, ¹S. Sessa Vidhya and ¹A. Manikandan

¹Department of Electronics and Communication Engineering,

²Department of Computer Science and Engineering, R.M.K College of Engineering and Technology,
601206 Chennai, India

Abstract: This study proposes an innovative architecture for medium speed Finite Impulse Response (FIR) filter based on Distributed Arithmetic (DA). This architecture composed of linear systolic arrays is derived by a two factor decomposition of both the order of the filter and the length of the input sequence. Many variants of medium speed systolic structures are possible that depend on how the input word length is factored. The design metrics, used to verify the effectiveness of the proposed architecture are the delay elements, speed in terms of maximum frequency and area as the number of slices. This design is implemented on the Xilinx Virtex-6 Field Programmable Gate Array (FPGA). A comparison of the performance metrics is made among the variants of medium speed architectures. The simulation results of the proposed design for different filter taps when related to the existing low speed and high speed systolic architectures shows that it requires intermediate area resources, latency and results in a medium speed. The implementation results also distinctly exhibit that the proposed architecture requires less hardware resources and also yields remarkably higher maximum frequency for various filter orders when compared with an existing DA based medium speed FIR filter.

Key words: Finite impulse response filter, DA, systolic array, LUT, area, speed

INTRODUCTION

The most powerful and frequently used operation in DSP systems is filtering. Finite Impulse Response digital filters are the crucial modules involved in many areas of Engineering such as digital communication, speech processing, audio and video processing spectral analysis, biomedical signal processing, etc. (Proakis and Manolakis, 1996). Designing FIR filters with reduced delay, less area and low power is very crucial in the current scenario (Parhi, 1999). Re-configurability, low cost, high speed performance and less manufacturing time are prominent features that have made FPGAs as the most preferred platform for realizing DSP operations. A discrete time invariant FIR filter that receives new input at every sampling period is characterized by the following Eq. 1 given by:

$$Z[n] = \sum_{i=0}^{T-1} B_i A[n-i] \quad (1)$$

Where:

T = The order of the filter

B_i = The known fixed filter coefficients

A[n-i] = The input data

Realizing the Multiply and Accumulate (MAC) operations of Eq. 1 with multipliers needs more hardware resources and computation time.

Amongst the multiplier-less architectures, bit serial DA is an effective algorithm that computes the dot product of two vectors in a definite number of steps. It involves storing all possible intermediate results in a Look Up Table (LUT)/Read Only Memory (ROM) and repeated shift accumulation of the stored values of the LUT that are selected by a sequence of input vectors (Parhi, 1999). The DA algorithm for digital filters was first formulated by Croisier and familiarized by Abraham and Bede. The major shortcoming of the DA method is that the memory requirement (2^T) increases exponentially as the number of filter taps (T) increases. Numerous methods to overcome this issue are suggested in the literature (Choi *et al.*, 2000; Eshtawie and Othman 2007; Sudhakar *et al.*, 2012; White, 1989; Yoo and Anderson, 2005; Shanthi and Nagarajan, 2013, 2014, 2015).

DSP applications that involve rigorous calculations can be effectively implemented using systolic designs with enriched features such as simplicity, higher throughput, uniformity and modularity (Kung, 1982; Mohanty and Meher, 1996; Meher, 2006; Parhi, 1999).

Recently, Meher *et al.* (2008) had suggested a DA based one dimensional (1D) and a two dimensional (2D) systolic architecture for the FIR filter. Jiafeng *et al.* (2010) had implemented an FIR filter for different speeds and Shanthi *et al.* (2014) had realized area-efficient and low latency architecture for high speed DA-FIR filter. The major shortcoming of existing medium speed FIR filter is that the number of adders as well as the depth of pipelined adder tree increases with the filter order consuming more hardware resources. In this study, we propose new medium speed architecture for DA based FIR filter that overcomes the above issue.

Mathematical formulation: The w bit two's complement binary number representation of the input sample is given by:

$$A[n-1] = -A_{i_0} + \sum_{j=1}^{w-1} A_{i_j} 2^{-j} \quad (2)$$

Where:

- $A_{i_j} \in \{0, 1\}$ = A_{i_j} the j bit of the input sample $A[n-i]$
- A_{i_0} = The sign bit
- $A_{i_{w-1}}$ = The Least Significant Bit (LSB)

Replacing $A[n-i]$ in the Eq. 1 by Eq. 2 and altering the order of summations yields the output expression as:

$$Z[n] = -\left(\sum_{i=0}^{T-1} B_i A_{i_0}\right) + \sum_{j=1}^{w-1} \left(\sum_{i=0}^{T-1} B_i A_{i_j}\right) 2^{-j} \quad (3)$$

The pre-calculated 2^T possible values of the terms within the brackets of the Eq. 3 for a set of fixed filter coefficients B_i ($i = 0, 1, 2, \dots, T-1$) are stored in the LUT/ROM. Each entry of the LUT is accessed, using the T bit input address sequence $\{A_{i_j} \text{ for } 0 \leq i \leq T\}$. One filter output $Z[n]$ is obtained by right shift accumulating the LUT results every w clock cycles.

Furthermore, for ease of understanding of the proposed architecture, the input data $A[n]$ is represented as unsigned binary numbers given by:

$$A[n-i] = \sum_{j=0}^{w-1} A_{i_j} 2^j \quad (4)$$

Substituting Eq. 4 in Eq. 1 and rearranging the summations, results in the following output response of an unsigned DA FIR filter given by:

$$Z[n] = \sum_{j=0}^{w-1} \left(\sum_{i=0}^{T-1} B_i A_{i_j}\right) 2^j \quad (5)$$

The limitation of higher memory requirements (2^T) and longer time to acquire the data from LUT/ROM for higher order filters is overcome by decomposing the order of

filter as $T = x \cdot y$ where 'x' denotes the number of smaller LUTs and 'y' is the number of address bits to each LUT. However, it requires additional adders to sum the intermediate partial values obtained from the LUTs. The output response of the unsigned DA FIR with decomposition is as follows:

$$Z[n] = \sum_{j=0}^{w-1} \left(\sum_{k=0}^{x-1} \left[\sum_{i=ky}^{(k+1)y-1} B_i A_{i_j}\right]\right) 2^j \quad (6)$$

Therefore, Eq. 6 can be represented as a LUT read operation with the y bit address for $0 \leq j \leq w-1$ and $0 \leq k \leq x-1$. Equation 6 is re-written as follows:

$$Z[n] = \sum_{j=0}^{w-1} \left(\sum_{k=0}^{x-1} M(u)_{j,k}\right) 2^j \quad (7)$$

$$M(u)_{j,k} = \sum_{i=ky}^{(k+1)y-1} B_i A_{i_j} \quad (8)$$

Where:

M = A LUT read operation

$(u)_{j,k}$ = A group of binary bits that is used as an address for the ROM/LUT

The address bit vector $(u)_{j,k}$ is given by:

$$(u)_{j,k} = \left[(A)_{ky,j}, (A)_{(1+ky),j}, \dots, (A)_{(y-1+ky),j} \right] \quad (9)$$

for $0 \leq j \leq w-1$ and $0 \leq k \leq x-1$, 'w' being the input word length. The Dependence Graph (DG) containing 'w' rows, demonstrating the calculations of the FIR filter of Eq. 7 is shown in Fig.1. Each row has 'x' number of nodes 'G' and a single node 'H'. The job of each node is shown in Fig. 1b and c. The address bit vector $(u)_{j,k}$ of 'y' bits applied to each node G, fetches the corresponding value stored in the LUT. If the operation of each row of DG is projected onto a single row of Processing Elements (PE) on a time delay basis, the resulting structure is the 1D systolic array with less speed. If each row of the DG is allotted separately to a linear systolic array, then the resulting structure is a 2D systolic architecture with high speed/throughput (Meher *et al.*, 2008).

The proposed design is a new systolic structure that results in medium speed obtained by decomposing the input word length 'w' as $w = R \cdot L$ where 'R' represents the number of linear systolic arrays and 'L' denotes the number of rows of the dependence graph that are projected onto a linear systolic array. Variants of the medium speed architecture are possible with different values of 'R' and 'L' that are chosen as per the hardware, time-delay and latency requirements of constraint based DSP systems and applications.

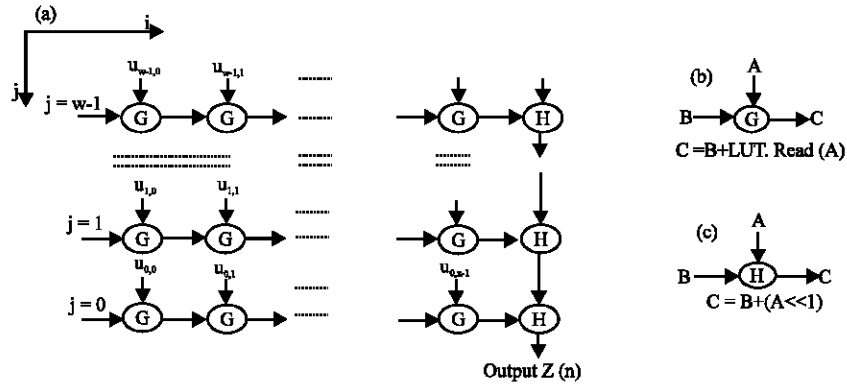


Fig. 1: Dependence graph for DA based FIR filter: a) Complete DG; b) Role of node G and c) Role of node H

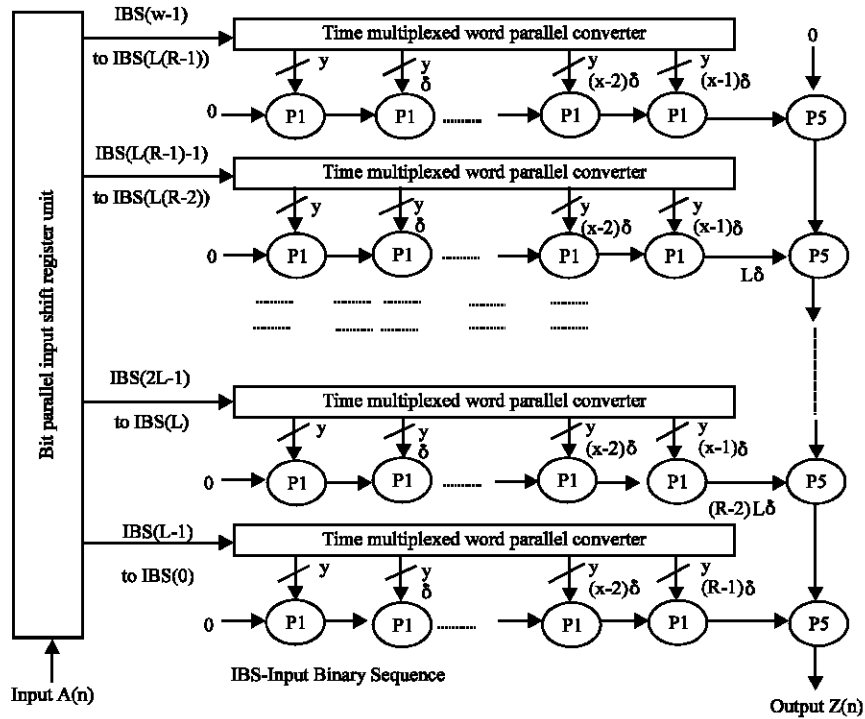


Fig. 2: Proposed medium speed systolic architecture for DA based FIR filter

MATERIALS AND METHODS

Proposed strategy: The proposed medium speed architecture for the DA based FIR filter is shown in Fig. 2. There are ‘R’ linear systolic arrays with each row consisting of ‘x’ number of P1 processing elements and one output cell P5. The bit parallel input shift register accepts an input A[n] in each cycle and produces ‘w’ words of ‘T’ bits each formed by the same positional bits of all the input words. The ‘w’ Input Binary Sequences (IBS) are divided into R groups of $L = w/R$ words as $IBS(L-1-0), IBS(2L-1-L), \dots, IBS((L(R-1)-1) \text{ to } IBS(L(R-2))$,

$IBS(w-1)-(L(R-1))$. The Most Significant Words (MSW) of each group ($IBS(w-1), IBS(L(R-1)-1), \dots, IBS(2L-1), IBS(L-1)$) are applied first to their corresponding linear systolic array for processing, continuing until the Least Significant Word (LSW). The input of each successive processing element P1 is delayed by one clock cycle (δ) with respect to its previous PE to satisfy the causality condition. Furthermore, to satisfy the data dependence criterion, the output cell P5 in the jth row is delayed by L clock cycles with respect to P5 in the (j+1)th row. The processing element P1 consists of a ROM/LUT and an adder. The processing element P1 reads the content of

Table 1: Analytical comparison of time-complexity and hardware requirements of the various speed architectures

Architectures	Low speed (1D) architecture	High speed (2D) architecture	Proposed medium speed architecture
Number of ROM/LUT words	$x \cdot 2^{Tx}$ (or) $x \cdot 2^y$	$w \cdot x \cdot 2^{Tx}$ (or)	$R \cdot x \cdot 2^{Tx}$ (or)
Number of adders	$x+1$	$w(x+1)$	$R(x+1)$
Number of delays	$x-1$	$R(x-1) + \sum_{i=1}^R (R-i)L$	$w(x-1) + \sum_{i=1}^{w-1} (w-i)$
Number of shift accumulators	1	w	R
Latency in cycles	$w+x$	$w+x$	$L+x+(R-1)L$
Throughput per cycle	$1/w$	1	$1/L$

Table 2: Comparison of design metrics of the medium speed architecture for an 8 tap FIR filter for different linear systolic arrays for a word length $w = 16$

No. of linear systolic arrays (R)	No. of input rows (L)	Maximum frequency (MHz)	Occupied slices	No. of ROMs used	Consecutive outputs/latency
2	8	367.377	53	4	Every $w/2 = 8$ cycles
4	4	367.377	93	8	Every $w/4 = 4$ cycles
8	2	367.377	173	16	Every $w/8 = 2$ cycles

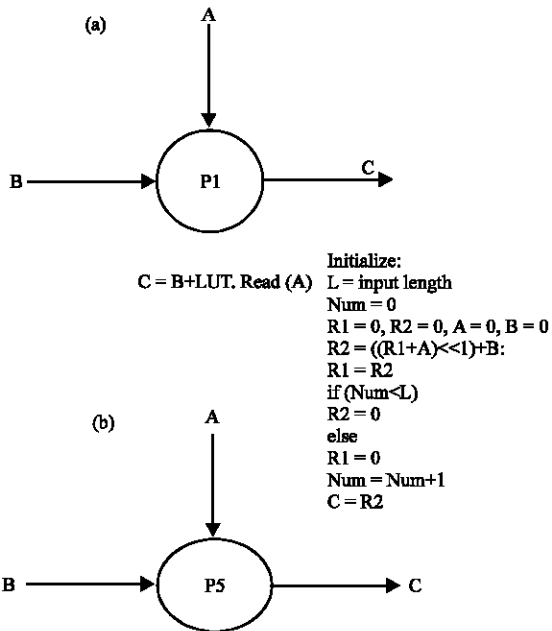


Fig. 3: a) Function of the processing element P1 and b) Function of output cell P5

LUT addressed by the input from the top and adds it to the input available from the left as shown in Fig. 3a. The output cell P5 performs the algorithm shown in the Fig. 3b. Generally, the output cell P5 adds its contents to the input available from the top, performs a left shift of the result and adds it to the input available from the left. The first output is obtained in $(L \text{ inputs} + \text{No. of PEs in a row} + (R-1) L \text{ delays})$ cycles and the consecutive filter outputs are obtained every L clock cycles. Analytical comparison of the proposed medium speed architecture with low and high speed architectures of Meher *et al.* (2008) are presented in Table 1.

RESULTS AND DISCUSSION

To make a fair comparison based on hardware requirements and time-delay complexities, the proposed

design 1D low speed and 2D high speed architectures (Meher *et al.*, 2008) and also the existing medium speed architecture of Jiafeng have been coded in Verilog HDL for various filter orders and implemented on the xilinx virtex ML605 kit with xc6vlx240t-1 ffg1156 FPGA device for an input word length of $w = 16$. The results presented in Table 2 are in line with the theory explained in the previous study. The hardware requirements in terms of the occupied slices and number of ROMs increase with number of linear systolic arrays. All the variants of the medium speed architecture for an 8-tap filter, yield their first output after 18 cycles but the successive outputs depend on the number of linear systolic arrays used in that particular design. Based on the throughput and hardware requirements of DSP systems, proper medium speed architecture is selected.

The comparison of the design metrics of Table 3 evidently shows that the proposed architecture results in an intermediate speed and hardware requirements when compared with the 1 and 2D systolic structures. The proposed medium speed architecture with four linear arrays has been used for comparison with other architectures. A comparison of the number of ROM units and slice LUTs of the proposed medium speed architecture with low and high speed architectures is shown in Fig. 4.

The outstanding performance of the proposed architecture than the existing one is due to pipelined systolic processing elements that yield higher speed as presented in Table 4. Moreover, the delay elements between the processing elements of the proposed structure reduce the critical path, thereby yielding a higher speed. Furthermore, there are no adder trees involved in the proposed systolic architecture resulting in less occupied slices. However, an eight tap filter of the proposed design shows a slight increase in the number of slices due to an additional adder and delay element involved that is overcome as the filter order increases.

Table 3: Comparison of design metrics of various speed architectures for different filter orders

No. of filter taps	1D Low speed architecture		2D High speed architecture		Proposed medium speed architecture	
	Maximum frequency (MHz)	Occupied slices	Maximum frequency (MHz)	Occupied slices	Maximum frequency (MHz)	Occupied slices
8	302.389	19	527.844	202	367.377	93
16	288.767	31	522.466	421	357.143	144
32	273.373	54	506.971	793	345.125	264
64	254.388	105	490.316	1444	329.598	459

Table 4: Comparison of design metrics of the proposed and the existing medium speed DA based FIR filter

No. of filter taps	Jiafeng Xie		Proposed medium speed architecture	
	Maximum frequency (MHz)	Occupied slices	Maximum frequency (MHz)	Occupied slices
8	140.845	68	367.377	79
16	102.648	164	357.143	144
32	81.466	344	345.125	264
64	64.914	646	329.598	459

CONCLUSION

A medium speed systolic architecture for the FIR filter using DA was designed and implemented by the flexible decomposition of input word length (w) and the order of the filter (T). Various ways of decomposing the word length resulted in numerous medium speed architectures. The hardware, speed and latency requirement of the DSP application leads to the selection of a particular medium speed architecture. The existing low speed (1D) and high speed (2D) structures have been elucidated to highlight the attractive features of the developed architecture and portray its capability in DSP applications where medium speed FIR filter operation is required. The implementation of the proposed FIR filter has resulted in increased speed of operation and lower area requirements, thereby exhibiting its superiority over the existing medium speed implementation.

REFERENCES

Choi, P., S.C. Shin and J.G. Chung, 2000. Efficient ROM size reduction for distributed arithmetic. Proceedings of the IEEE International Symposium on Circuits and Systems, May 28-31, 2000, IEEE, Geneva, Switzerland, ISBN:0-7803-5482-6, pp: 61-64.

Eshtawie, A.M. and M. Othman, 2007. An algorithm proposed for FIR filter coefficients representation. Intl. J. Appl. Math. Comput. Sci., 4: 24-30.

Kung, H.T., 1982. Why systolic architectures. IEEE. Comput., 15: 37-45.

Meher, P.K., 2006. Hardware-efficient systolization of DA-based calculation of finite digital convolution. Circ. Syst. II: IEEE Trans. Exp. Briefs, 53: 707-711.

Meher, P.K., S. Chandrasekaran and A. Amira, 2008. FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic. IEEE Trans. Signal Process, 56: 3009-3017.

Mohanty, B.K. and P.K. Meher, 1996. Cost-effective novel flexible cell-level systolic architecture for high throughput implementation of 2-D FIR filters. IEEE. Proc. Comput. Digital Tech., 143: 436-439.

Parhi, K.K., 1999. VLSI Digital Signal Processing Systems: Design and Implementation. 2nd Edn., Wiley-Interscience Publication, New York, USA., ISBN-13: 9780471241867, Pages: 784.

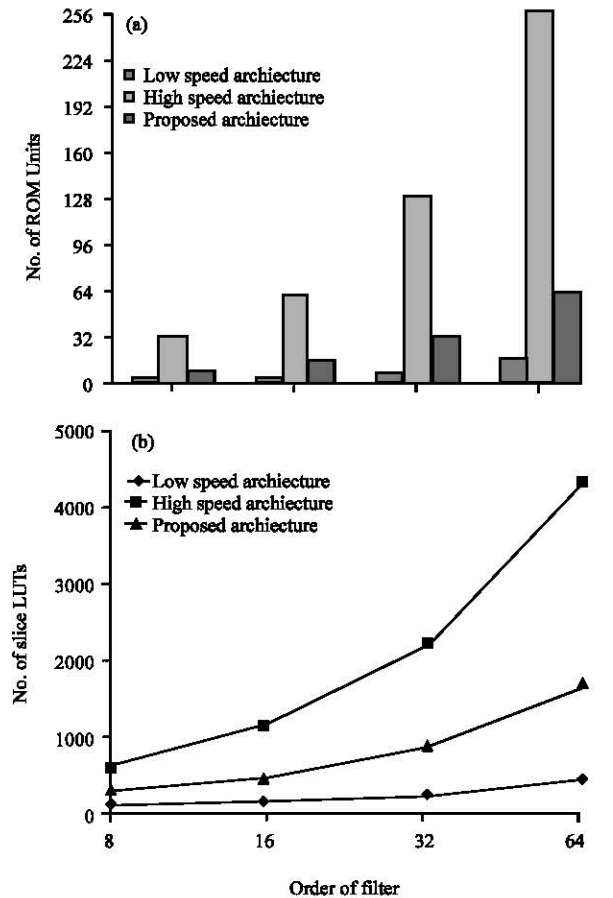


Fig. 4: Comparison of: a) ROM units and b) Slice LUTs of the proposed medium speed architecture with low and high speed architectures (set the caption)

- Proakis, J. and D. Manolakis, 1996. Digital Signal Processing: Principles, Algorithms and Applications. Prentice-Hall, Englewood Cliffs, New Jersey, USA.
- Shanthi, K.G. and N. Nagarajan, 2013. Memory based hardware efficient implementation of FIR Filters. *Int. Rev. Comput. Software*, 8: 1718-1733.
- Shanthi, K.G. and N. Nagarajan, 2014. High speed and area efficient FPGA implementation of FIR filter using distributed arithmetic. *J. Theoret. Appl. Inf. Technol.*, 62: 627-633.
- Shanthi, K.G. and N. Nagarajan, 2015. Area-efficient and low latency architecture for high speed fir filter using distributed arithmetic. *ICIC Express Lett. Part B Applic. Int. J. Res. Surv.*, 6: 2053-2058.
- Shanthi, K.G., N. Nagarajan and C. Kalieswari, 2014. FPGA realization of high speed FIR filter based on distributed arithmetic. *Int. J. Eng. Technol.*, 6: 1407-1414.
- Sudhakar, V., N.S. Murthy and L. Anjaneyulu, 2012. Area efficient pipelined architecture for realization of FIR filter using distributed arithmetic. *Proceedings of the International Conference on Industrial and Intelligent Information*, October 7-12, 2012, IACSIT Press, Singapore, pp: 169-173.
- White, S.A., 1989. Applications of the distributed arithmetic to digital signal processing: A tutorial review. *IEEE. ASSP. Mag.*, 6: 4-19.
- Yoo, H. and D.V. Anderson, 2005. Hardware efficient distributed arithmetic architecture for high-order digital filters. *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing Vol. 5*, March 23-23, 2005, IEEE, Philadelphia, Pennsylvania, ISBN:0-7803-8874-7, pp: 125-128.