



Realization Of Fir Filter By An Efficient And Flexible Systemization Using Distributed Arithmetic

A.Aruna Sree *1, Dr. J V R Ravindra *2

M.Tech (Decs) Student Department Of Ece, Vce, Shamshabad, R.R (Dist), Ap, India.

Professor, Department Of Ece, Vce, Shamshabad, R.R (Dist), Ap, India

Abstract

This paper presents the realization of area efficient architecture using Distributed Arithmetic (DA) for implementation of Finite Impulse Response (FIR) filter. The performance of the bit-serial and bit parallel DA along with pipelining architecture with different quantized versions are analyzed for FIR filter design. The systolic decomposition scheme is found to offer a flexible choice of the address length of the lookup tables (LUT) for DA-based computation to decide on suitable area time tradeoff. It is observed by using smaller address length for DA-based computing units it is possible to reduce the memory size, but on the other hand that lead to increase of adder complexity and the latency. Pipelined DA architecture has achieved double the maximum frequency of operation when compared to their non-pipelined implementations with an increase in hardware. For efficient DA-based realization of Fir filters of different orders, the flexible linear systolic design in implemented on Xilinx Vertex - E XCV2000E FPGA using a hybrid combination of handled -C and parameterizable VHDL cores. Analyses of the result obtained indicate that performance metrics of the proposed implementation is broadly in line with theoretical expectations. More over the proposed FPGA implementation is found to involves significantly less area- delay complexity compared with the existing DA- based implementations of FIR filter.

Index terms: DA, FPGA, FIR, Linear convolution, systolic array.

IJMTARC

1. Introduction

Finite impulse response (FIR) digital filters are common components in many Digital Signal processing (DSP) system. Throughout the years with the increasing development in very large scale integration (VLSI) technology, the real time realization of FIR filter with less hardware requirement and less latency has become more and more important. As the complexity of implementation increases with the length of filter several algorithms have been mapped into effective architecture using ASIC's and FPGA's one of them being distributed arithmetic (DA).

The main portion of DA- based computation is lookup table (LUT) that stores the precompiled values and can be read out easily which makes DA- based computation well suited for FPGA realization, because the LUT is the basic component of FPGA. DA technique can be designed to meet various speed requirements for example it can be designed for high speed implementation where several bits of one word are processed per clock. Memory decomposition can be reduced the memory significantly but on the other side, it also

lead to the increasing of latency and number of adders. Presently high speed realization of FIR filter with low latency and chip area are in high demand. This paper addresses the issue of extending DA based FPGA implementation of FIR filter to an area- time optimized implementation. The multiplier less distributed arithmetic (DA)- based techniques has gained substantial popularity in recent years for its high throughput processing capability and increased regularity which results in cost effective and area-time efficient computing structures. The main operations required for DA- based computation of inner product are a sequence of lookup table (LUT) accesses followed by shift-accumulation operation of the LUT output. All these structure however are not suitable for implementation of the FIR filters in systolic hardware since the partial products available from the partitioned memory modules are summed together by a network of output adders. A new tool for the automatic generation of highly parallelized FIR filters based on PARO design methodology is presented in, where the authors have performed hierarchical partitioning in order to balance the amount of local memory with external communication and they have achieved higher throughput and smaller latencies by partial localization.



2. Related work

The rest of the paper is organized as follows. The formulation of the algorithms for flexible DA- based realization of FIR filter is described in the next section; and the systolic structures are derived from the dependence graphs of the algorithms in section III. The FPGA implementation methodology is described in section IV, and simulation results pertaining to the FPGA implementation are presented and discussed in section 5. conclusion along with the scope for future work is presented in section 6.

2.1 Formulation of the Algorithm

Description of conventional distributed arithmetic approach for inner- product computation and thereafter derives a decomposition scheme for flexibility DA- based sys-toleration of FIR filters.

2.1.1 Conventional DA Approach for inner –product computation

Let us consider the inner- product of two N-point vectors A and B given by

$$C = \sum_{k=0}^{N-1} A_k \cdot B_k \quad (1)$$

Where A is constant vector, while B may change from time to time. Assuming L to be the word length, each component of B may be expressed in two's complement representation

$$B_k = -b_{k0} + \sum_{l=1}^{L-1} b_{kl} \cdot 2^{-l} \quad (2)$$

Where b_{kl} denotes the l th bit of B_k .

$$C = \sum_{k=0}^{N-1} A_k \cdot b_{k0} + \sum_{k=0}^{N-1} A_k \cdot [\sum_{l=1}^{L-1} b_{kl} \cdot 2^{-l}] \quad (3)$$

To convert the conventional sum-of-products from of inner product of (1) into a distributed from the order of summations over the indexes k and l in the second term of (3) can be interchanged to have

$$C = \sum_{k=0}^{N-1} A_k \cdot b_{k0} + \sum_{l=1}^{L-1} 2^{-l} \cdot [\sum_{k=0}^{N-1} A_k \cdot b_{kl}]. \quad (4)$$

Without loss of generality for simplicity of discussion we may assume the signal samples to be unsigned words of size L, although the DA decomposition algorithm can be used for two's complemented coding and offset binary coding also. The inner – product given by (4) then can be expressed in a simpler from

$$C = \sum_{l=0}^{L-1} 2^{-l} \cdot C_l \quad (5a)$$

Where

$$C_l = \sum_{k=0}^{N-1} A_k \cdot b_{kl} \quad (5b)$$

Since vector A is assumed to be constant, and each element of the N-point bit-sequence $\{b_{kl} \text{ for } 0 \leq k \leq N-1\}$ can either be zero or one any of the partial sum C_l for $l=0, 1, \dots, L-1$ can have 2^N possible values. All the 2^N possible values of C_l can therefore, be precompiled and scored in a ROM, such that while computing the inner product the partial sums C_l can be read out from the ROM using the bit sequence $\{b_{kl} \text{ for } 0 \leq k \leq N-1\}$ as address bits.

2.1.2 Decomposition scheme for DA-based implementation of FIR filter

The output of an FIR filter of order can be computed as inner products of the impulse response vector $\{h(k), \text{ for } k=0, 1, \dots, N-1\}$ and an input vector $\{S_n(k), \text{ for } k=0, 1, \dots, N-1\}$ given by

$$y(n) = \sum_{k=0}^{N-1} h(k) \cdot S_n(k) \quad (6)$$

where $S_n(k) = x(n-k)$, and $x(n)$ is the current input sample. $\{h(k)\}$ is a fixed sequence while the input sequence $\{S_n(k)\}$ is derived from serially shifted input samples using a window size N, such that it receives a fresh input samples using and leave its oldest sample. Comparing (6) with (1) the filter output can be computed according to (5) as

$$y(n) = \sum_{l=0}^{L-1} 2^{-l} \cdot C_l \quad (7a)$$

Where

$$C_l = \sum_{k=0}^{N-1} h(k) \cdot (S_n(k))_l \quad (7b)$$

with $(S_n(k))_l$ for $l=0, 1, \dots, L-1$, being the l th bit of $S_n(k)$. Equation (7) can be directly used for straightforward DA- based implementation of FIR filter using a ROM containing of 2^N possible values of C_l . for large values of N, however, the ROM size becomes too large and so also the ROM access time consequently becomes large. The straightforward DA implementation is therefore, not suitable for large filter orders. When N is a composite number given by $N=PM$, one can map the index k into $(m + pM)$ for $m=0, 1, \dots, M-1$ and $p=0, 1, \dots, P-1$, in order to express in equation (7).

2.2 Derivation Of The Structures

This approach suggested in we derive here the DA-based 1-dimensiona; (2-D) systolic array for FIR filter from dependence graph (DG) representation of DA-based composition.

2.2.1 D systolic array for FIR Filters

The DG for computation of FIR filter output according to Fig.1 is shown. It consists of L rows, where each row consists of p number of node-A and one boundary node-B. The function of node-A and node-B are deposited in Figs. 1(b) and 1(c), respectively. A bit vector $(b_n)_{l,p}$ consisting of a sequence of M bits of the sequence as given is fed to the node-A on $(l+1)$ -th row and $(p+1)$ -th column. The node uses the sequence of M input bits of



the input bit-vector as address for an LUT and reads the content stored at the location specified by the address.

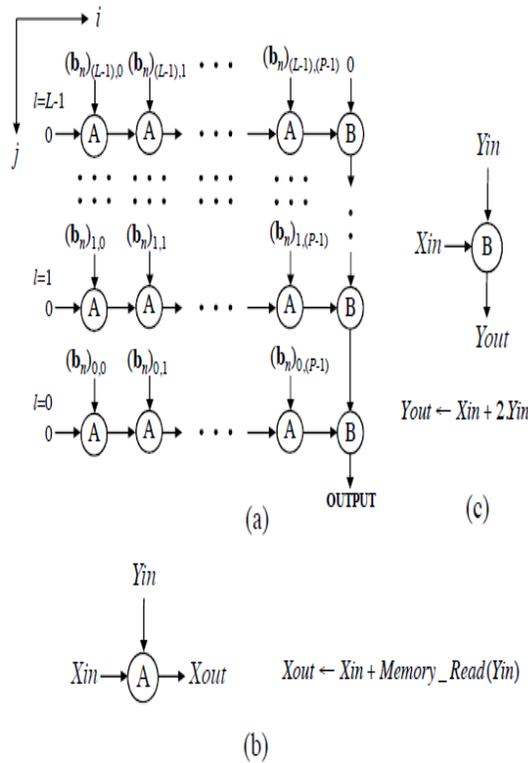


Fig.2.2.1.1 The DG for DA-based implementation of FIR filters. (a) The DG. (b) Function of node A. (c) Function of node B.

The value read from the LUT is then added with the input available from its left, and the sum is passed to the node on its right. Nodes-B performs a shift-add operation such that it makes a left-shift of the bits of the input available for the top, then adds the input available from the left to the left-shifted values and passes the result down to its adjacent node. The DG can be noticed vertically along the projection direction $[0 \ 1]^T$ with default schedule to derive a linear array consisting of P number of PEs and an output-cell as shown in Fig.2.

The input sequence $\{x(n)\}$ is fed to a serial-in parallel-out input register, where content of the register is serially- right-shifted by one position and transformed in parallel to the bit-serial word-parallel converter in every L cycles. The bits of vector $(b_n)_{l,p}$, are derived from the bit-serial word-parallel converter and fed to the $(p + 1)$ -th PE in most significant bits (MSBs) to least significant bits (LSBs) order cycle period such that $(L-1)$ -th bits of input values are fed to the PE at first, and the zeroth bits are fed at the end. The value read from the ROM is then added to the input available to the PE from its left. During every cycle period, the sum is then

transferred as output to its right. Function of the output cell is shown in fig.2©. Each output cell contains a shift register and an adder. During a cycle period it shifts the content of its register left by one position and then adds the available input to the recently shifted content in its register. After L cycles it derives a desired filter output.

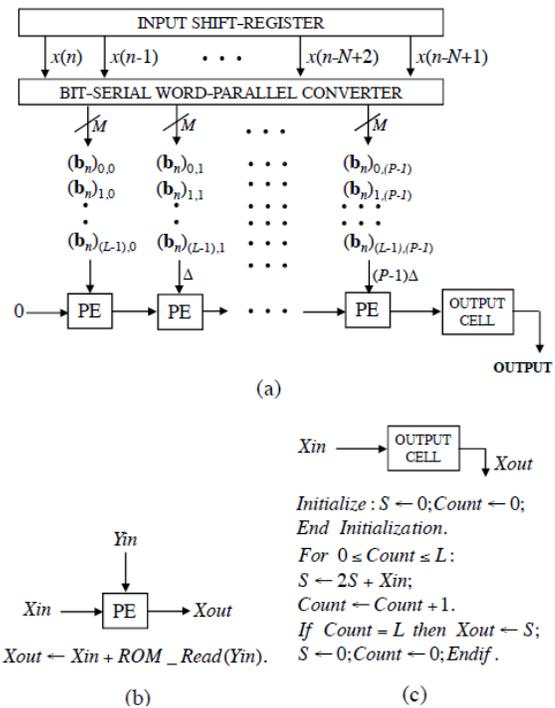


Fig.2.2.1.2 The 1-D array for DA-based implementation of FIR filters. (a) the linear systolic array. (b) Function of PE. (c) Function of output cell.

While the successive output becomes available in every L cycles. For high throughput application one may, however a structure with N number of 1-D arrays which would yield N convolved output in every L cycles duration.

2.3 2-D Systolic Structure for FIR filters

For high- throughput implementation of FIR filters, each node of DG of Fig.1 can be assigned to a PE exclusively to obtain a 2-D systolic array of L rows and $(P + 1)$ columns as shown in fig.3. The input samples are fed to a bit-parallel word-serial converter which receives a new input sample in every cycle period, and generates L number of bits of the input sample and feeds one bit each to L number of bit levels serial-in parallel-out shift register associated with each row of PEs as shown in Fig.3(a). Each SIPOSR contains a bit-stream of the corresponding bits of all the input words, such that the SIPOSR on upper- rows contain the



more significant bits compared to that of the lower rows.

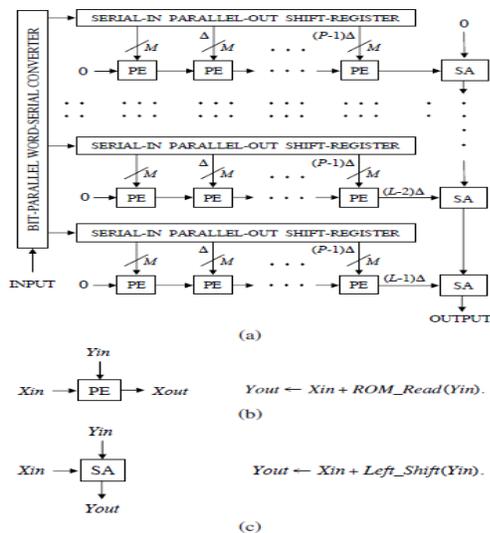


Fig.2.3.1 The 2-D array for FIR filters. (a) The 2-D systolic array. (b) Function of PE. (c) Function of SA cell.

Each of the SIPOSRs of the structure shifts its content to right by one bit- location and receives a new bit in every cycle, open arrival of a fresh sample to the bit serial wors-parallel converter. The bit vector $(b_n)_{l,p}$ consisting of M number of bits from the $(l+1)$ -th SIPOSR is loaded to the $(p+1)$ -th PE of the $(l+1)$ -th row. Each PE shown in Fig.3(b) uses the bit vector $(b_n)_{l,p}$ as address for its LUT to read a partial result. The PE then adds the input available from the left with its recently read partial result. The PE then adds the input available from the left with its recently read partial result, and passes that out to its right. Each row of the structure is terminated with an SA cell. The function of SA is depicted in Fig.3(c). Each SA during a cyclic period makes a left shift of its input available from the top and adds that input to its input available from the left. The sum is then passed downward to its adjacent SA.

3. Methodology

This part is concerned with the description of the proposed system and methodology for FPGA implementation of the FIR filter based on systolic decomposition of DA-based computation discussed in the previous section. By using a systematic design flow a power-aware area-time-efficient optimized FPGA realization has been obtained here for the systemized filter.

3.1 Design environment

The proposed computing system is designed by a hybrid combination of Handel-C and parameterized VHDL codes. The

main advantage of handle-c over the other hardware description language (HDLs) is its rapid prototyping capabilities. Several works have shown that Handel –C shorten design time by a factor of 3-4 times with approximately the same operating speed compared to traditional HDLs. The VHDL based cores are generated using Xilinx Coregent and are used for small frequency required blocks such as shifters and accumulators. Handle-C is used at the top level for architecture description and integration of the cores. This is particularly important because non optimal place and route tends to use long nets that consume more power than short nets. Finally XPower is used to obtain the power estimation from which various energy consumption measures can be calculated.

3.2 Platform details

In order to estimate the performance of the flexible DA-based systemized FIR filter the design has been prototyped on the Celoxica RC1000 board containing the Xilinx XCV2000E FPGA. The RC1000 also has four memory banks which communicate with the host by DMA data transfer mechanism. The proposed prototyping platform is pictorially presented in Fig.4. The host application obtains the input vector of arbitrary length and transfers the data to SRAM bank-0 by means of DMA transfer after taking control of the FPGA memory banks. The control is then released and a pre-selected bit stream file is downloaded to the FPGA for configuration. The FPGA takes control of the memory banks there after performs the FIR filtering and wires the result to SRAM Bank-1. The control of memory bank is finally passed back to the host application which reads the filter output from Bank-1.

I would like to note here that we have used DRAMs instead of BRAMs for implementation of the architecture, because, BRAMs are specialized RAMS which are dependent on the architecture of the FPGA used and consequently are not suitable to have a fully parameter sable and scalable architecture for implementing DA.

3.3 Simulation Results and Discussion

The result of FPGA implementation of the flexible 1-D systolic design Fig.2 in terms of area and maximum usable frequency metrics with respect to the filter order N and address-length M are presented in Table I.

It can be seen that for a given filter order N the case for M=4 yields the most area time efficient architecture when compared to the case for M=2 and 8. This can be explained by the fact that the increase in control logic and number of delay element out weight the gains made by reduction of LUT size for M=2 while for M=8, the memory requirement of LUTs is too high.

TABLE I. The key performance metrics of the proposed FPGA implementation of the DA-based Fir filter (L=8)



Also it is worth mentioning that four input LUTs are the basic building blocks of the vertex-E configurable logic block (CLB) structure and this account for the most efficient mapping of the DA-LUT to the available resources for the case of M=4.

| Order (N) | Address size(M) | Area (slices) | Frequency (MHz) |
|-----------|-----------------|---------------|-----------------|
| 8 | 2 | 144 | 71.788 |
| | 4 | 133 | 74.025 |
| | 8 | 149 | 62.181 |
| 16 | 2 | 287 | 65.807 |
| | 4 | 260 | 67.222 |
| | 8 | 286 | 60.114 |
| 32 | 2 | 555 | 62.771 |
| | 4 | 524 | 63.131 |
| | 8 | 553 | 55.313 |
| 64 | 2 | 1057 | 61.244 |
| | 4 | 1061 | 64.049 |
| | 8 | 1094 | 54.750 |

Table II. Comparison and performance of the proposed implementation and the existing implementation of DA-Based FIR filter.

| filter order | proposed | | | Yoo <i>et al</i> [12] | | |
|--------------|---------------|-----------|------------|-----------------------|-----------|------------|
| | area (slices) | MUF (MHz) | gate count | area (slices) | MUF (MHz) | gate count |
| 8 | 133 | 74.025 | 2512 | 146 | 70.552 | 3365 |
| 16 | 260 | 67.222 | 4998 | 283 | 62.775 | 6337 |
| 32 | 524 | 63.131 | 11128 | 547 | 61.166 | 12235 |
| 64 | 1061 | 64.049 | 23878 | 1076 | 57.192 | 24801 |

Hence we have faithfully re-implemented the architecture presented on the same platform that has been used in this project for word length L=8. It may be noticed that the same design tools and identical P&R settings have been used and the same amount of optimization has been applied for Yoo's design as has been for the case of the proposed implementation.

It is clear that the proposed implementation significantly outperforms the existing implementations in terms of three important key metrics namely the area occupied maximum usable frequency and gate-count for all the values of N. Moreover in our design only a single shift-accumulate operation is needed, irrespective of filter order N as a result of systolization.

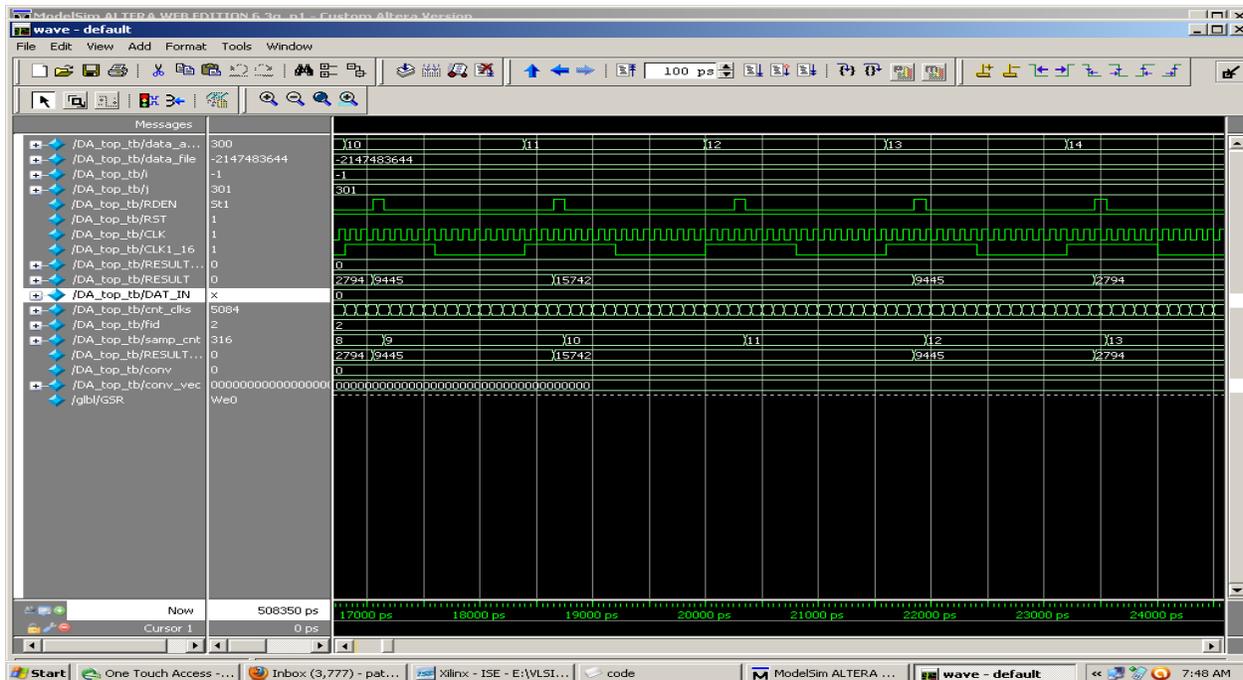


Fig. 3.3.1 Comparison and performance of the proposed implementation and the existing implementation of DA-Based FIR filter.



4. Conclusion

Finite impulse response filter plays an important role in many Digital Signal Processing applications. Flexible designs of 1-D and 2-D systolic computation structure are derived for area delay power efficient implementation of FIR filter by added decomposition of DA-based inner-product computation. The 1-D systolic structure is implemented on a Xilinx Virtex-E XCV 2000E FPGA by a hybrid combination of Handled-C and parameterizable VHDL cores. It is found that the 1-D structure with address-length $M=4$ yield the best of area –delay – power –efficient implementations for various filter orders, moreover , it is found that the proposed implementation involves significantly less area-delay complexity compares with the existing DA- based implementation of FIR filters. For high speed applications, a 2-D systolic array consisting of L number of linear systolic arrays could be used to process the individual bit- stream of input signal separately in different systolic arrays. Further work may be carried out to develop an efficient DA- based adaptive filter, and the two-factor DA-decomposition may be extended further for three-factor and four-factor decomposition of DA-based computation of FIR filtering.

5. References

- [1] J. G. Proakis and D. G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications. Upper Saddle River, NJ: Prentice- Hall, 2011.
- [2] Jiafeng Xie, Jianjun He, Guanzheng Tan, “FPGA Realization of FIR filters for high-speed and medium-speed by using modified distributed arithmetic architectures”, Microelectronics journal 41(2010) 365-370.
- [3] P. K. Meher, Shrutisagar Chandrasekaran and Abbas Amira, .FPGA realization of FIR filters by efficient and flexible systolization using distributed arithmetic., IEEE Trans. Signal process., vol. 56, no. 7, pp. 3009-3017, July 2008.
- [4] B.K.Mohanty, P.K.meher .Cost effective novel flexible cell-level systolic architecture for high throughput implementation of 2D FIR filters. IEE 1996. Technical note.
- [5] P. K. Meher, .Hardware-efficient systolization of DA-based calculation of finite digital convolution., IEEE Trans. Circuits Syst. II, Exp. Briefs, vol. 53, no. 8, pp. 707.711, Aug. 2006.
- [6] S.-S. Jeng, H.-C. Lin, and S.-M. Chang, “FPGA implementation of FIR filter using M-bit parallel distributed arithmetic,” in Proc. 2006 IEEE International Symposium on Circuits and Systems, ISCAS 2006, May 2006, p. 4.
- [7] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, “LMS adaptive filters using distributed arithmetic for high throughput,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 52, no. 7, pp. 1327–1337, July 2005.
- [8] H. Ruckdeschel, H. Dutta, F. Hannig, and J. Teich, “Automatic FIR filter generation for FPGAs,” in Proc. 5th International Workshop on Systems, Architectures, Modeling, and Simulation,

SAMOS 2005, T. D. H. et al., Ed., vol. LNCS 3553, July 2005, pp. 51–61.

[9] P. K. Meher, “Hardware-efficient systolization of DA-based calculation of finite digital convolution,” IEEE Trans. Circuits Syst. II: Express Briefs, vol. 53, no. 8, pp. 707–711, Aug. 2006.

[10] B. K. Mohanty and P. K. Meher, “Cost-effective novel flexible celllevel systolic architecture for high throughput implementation of 2-D FIR filters,” IEE Proceedings-Computers and Digital Techniques, vol. 143, no. 5, pp. 436–439, Nov. 1996.



Dr. J V R Ravindra, He is working as a professor in department of Electronics and Communication Engineering in Vardhaman College of Engineering (Autonomous), Kacharam, Shamshabad in Andhra Pradesh, India. His special field of Interest in Modeling of Ultra Low power Interconnects, High-speed and Low power Arithmetic Circuits, Low power DSP Architectures, Hardware Security, Wireless Sensor Networks. He is Dr. form Electronics and Communication Engineering in IIIT-H, Hyderabad.



A.Aruna sree was born in Hyderabad in the Andhra Pradesh, India on September 02. She graduated (B.Tech) from Lords College of engineering & technology under JNTU University and pursuing M.Tech (DECS) from Vardhaman college of engineering under JNTU University. Her special field of interest in Digital Signal Processing.