Middle-East Journal of Scientific Research 22 (4): 532-536, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.22.04.21918

Low Power Distributed Arithmetic Based Fir Filter

[1]*L. Murali,* [2]*D. Chitra and* [3]*T. Manigandan*

[1]Department of Electronics and Communication Engineering,
Hindusthan College of Engineering and Technology, Tamil Nadu, India
[2]Department of Computer science Engineering,
P A College of Engineering and Technology, Tamil Nadu, India
[3]Principal, P A College of Engineering and Technology, Tamil Nadu, India

Abstract: Datapath architectures are the critical components in computational intense applications and their architectural changes leads to changes in VLSI design constraints like area, performance and power. And in this modern automated world, the power constraint has been the major requirement; hence an effort has been applied regarding the necessary. This brief implements the Low power Finite Impulse Response (FIR) filter for the Digital signal processors (DSP) applications. Since it's a datapath arithmetic architectural change, the proposed architecture can be applied to any hierarchical architecture where power is the major constraint. Designs were developed and modeled with Verilog HDL and synthesized using Cadence RTL compiler by mapping to TSMC's 65nm technology. The proposed arithmetic has reduced the filter power by 10.62 % when benchmarked with standard ASIC design methodology.

Key words: Finite Impulse Response · Distributed Arithmetic · Low power arithmetic · Verilog

INTRODUCTION

Filters are used in signal processing applications like channel equalization, interference, echo cancellation system identification and noise cancellation, etc [1]. Filter output is the weighted sum of the past and present input samples, which is realized through the Multiply and Accumulate (MAC) unit in general DSP.

But the MAC units consume more area for multipliers to provide better system performance and it leads to high system cost. Hence the Distributed Arithmetic (DA) based architectures are the efficient techniques to realize the filters of higher order, as they can achieve high throughputs with multiplier-less architectures.

In DA, the pre-computed products will be stored in Look-up tables (LUT) and accessing & shift-accumulating the correct ones give the output of filter. The bit length of the input sequence decides the number of clock cycles required to get the filter output [2].

Several schemes were developed in the past to implement the filters using DA based architectures. A scheme named Offset binary coding (OBC) scheme is referred by the author in [2], where LUT size has been reduced to half to reduce the area and power consumption. In [1], author has proposed the novel LUT sharing method to generate the filter output to update the weights of the filter in Adaptive filters, which significantly reduces the overall area of the adaptive filter. Influence of dividing the larger LUT into smaller LUTs was introduced in [3], but additional adders were required to combine the smaller LUTs; which increases the dynamic power of the design.

Other than the DA based architectures common sub-expression elimination (CSE) methods were also used for low area and low power constraints in designing the filters. In CSE method, the number of adders required for the filter implementation, its complexity, power consumption, circuit speed and the depth of the adders are of concern. Adder depth is the number of adders that the input signal passes before the delay element. Algorithm based on the mixed integer linear programming (MILP) requiring minimum number of adders, was used for the linear FIR filter implementation in [4]. Author of [5], has reported that design implementations with respect to FPGA architectures; along with modification in CSE method and use of shift-add method in fixed-co-efficient multipliers reduces the number of adders required for the filter implementation. In [6], Martin *et al*. has suggested

Corresponding Author: L. Murali, Department of Electronics and Communication Engineering,
Hindusthan College of Engineering and Technology, Tamil Nadu, India.

that graph based methods provide better solutions than the CSE based methods and illustrated that the pipelined adder graphs yields high speed for the FIR filter implementation.

Author of [7], has demonstrated that the change in the datapath implementations improves the speed of addition by exploiting the benefits of pre-computed summations in optimized LUTs using efficient compressors.

Limitations of the State of the Art Methodologies:

- Methods deal at the algorithmic level and are in the abstract method for datapath architectures. But the low level datapath architectures impact the design largely as they are the most copiously used components.
- CSE, modified CSE and graph based methods tries to reduce the number of adders, but limits to optimize the adder architectures as per the design requirement like area, speed and power constraints.
- In [7], author has utilized the compressors in the addition stage to reduce the speed but limits itself for the power constraint designs or applications.

Hence there is a need for the development of power aware architectures which suits the power constrained designs, as the current technology designs are in need. Thus in this brief we propose the low power arithmetic adders for the distributed arithmetic based filter design, which consume less power than the regular standard architectures. This enables the design to be analyzed as the low power corner and triggers the development of architectures as per the design requirements. In this brief, other sections are arranged as in the following. Section II describes the DA based architecture used for filtering process. Section III characterizes the datapath arithmetic architectures and the design methodology. Results and discussions are evaluated in sections IV and conclusion is provided in section V, while references are given in last section.

DA Architecture: In digital signal processing, filtering is one of the most widely computed processes. Filtering is the linear convolution of the weights "wn" and inputs "xn", for an Nth order filter the output "yn" is given by

$$y\,(n) = \textstyle\sum_{k=0}^{N} x\,(n-k)\,w_k \tag{1}$$

As per equation (1), "N + 1" numbers of MAC units are required for the generation of each output sample "yn". As the area consumed by mutlipliers are more, MAC/filters are frequently replaced by the multiplier-less architectures.

DA's are used to design serial-bit-level architectures for multiplier designs and filters implemented using DA method uses input samples as the address to access serially the contents of LUT, which stores the pre-computed sums of co-efficients [8].

Denoting each of the samples x (n – k) in 2's complement form, we have

$$x(n-k) = x_{n-k} = -b_{k},B-1 + \textstyle\sum_{j=1}^{B-1} b_{i},B-1-j2^{-j} \tag{2}$$

Now putting (2) in (1) and re-arranging

$$y\,(n) = \textstyle\sum_{j=0}^{B-1} c_{B-1-j}2^{-j} \tag{3}$$

Where, $c_{B-1-j} = \sum_{k=0}^{N-1} w_i\, b_{i,B-1-j}$ and

$$c_{B-1} = -\sum_{k=0}^{N-1} w_i\, b_{i,B-1}$$

For a given set of weights "w_i", C_{B-1-j} takes only one of 2^N combinations which are pre-computed and stored in the LUT. Figure 1 shows the 4-tap FIR filter implementation based on DA.

For higher order filters, the LUT size would be larger and reducing it is of a concern. In such a case the larger LUTs are decomposed into smaller ones as described in [3]. Different from the ROM decomposition, offset binary coding is applied; where the LUT combinations can be reduced from "2^N" to "2^{N-1}" without increasing the LUTs.

Rewriting the equation (2), as

$$\begin{aligned} x(n-k) &= \tfrac{1}{2}[(x_{n-k} - (-x_{n-k})] \\ &= \tfrac{1}{2}[-(b_{k,B-1} - \bar{b}_{k,B-1-j})2^{-j} - 2^{-(B-1)}] \end{aligned} \tag{4}$$

Choosing,

$$d_{k,j} = \begin{Bmatrix} -(b_{k,j} - \bar{b}_{k,j}), & j \neq B-1 \\ -(b_{k,B-1} - \bar{b}_{k,B-1}), & j \neq B-1 \end{Bmatrix} \tag{5}$$

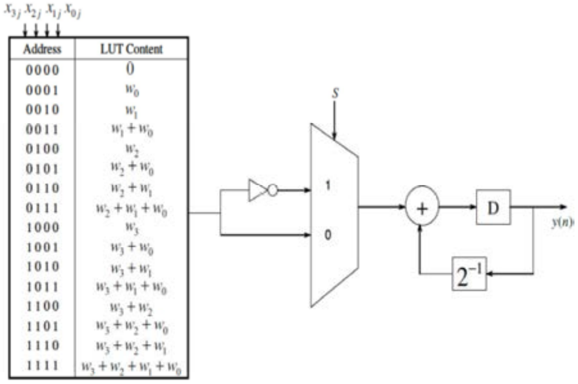
Putting equations (4) and (5) in (1)

$$y(n) = \sum_{j=0}^{B-1}\left(\sum_{k=0}^{N-1}\frac{1}{2}w_k d_{k,B-1-j}\right)2^{-j} - \left(\frac{1}{2}\sum_{k=0}^{N-1} w_k\right)2^{-(B-1)}$$

Defining

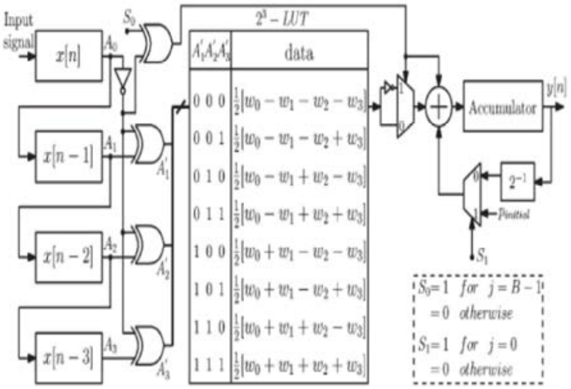
$$p_j = \sum_{k=0}^{N-1}\frac{1}{2}w_k d_{k,j}\ 0 \leq j \leq B-1$$

$$p_{initial} = -\tfrac{1}{2}\textstyle\sum_{k=0}^{N-1} \tag{6}$$



Address	LUT Content
0000	0
0001	w_0
0010	w_1
0011	$w_1 + w_0$
0100	w_2
0101	$w_2 + w_0$
0110	$w_2 + w_1$
0111	$w_2 + w_1 + w_0$
1000	w_3
1001	$w_3 + w_0$
1010	$w_3 + w_1$
1011	$w_3 + w_1 + w_0$
1100	$w_3 + w_2$
1101	$w_3 + w_2 + w_0$
1110	$w_3 + w_2 + w_1$
1111	$w_3 + w_2 + w_1 + w_0$

Fig. 1: DA based 4-tap FIR Filter implementation



$A'_1 A'_2 A'_3$	data
0 0 0	$\frac{1}{2}[w_0 - w_1 - w_2 - w_3]$
0 0 1	$\frac{1}{2}[w_0 - w_1 - w_2 + w_3]$
0 1 0	$\frac{1}{2}[w_0 - w_1 + w_2 - w_3]$
0 1 1	$\frac{1}{2}[w_0 - w_1 + w_2 + w_3]$
1 0 0	$\frac{1}{2}[w_0 + w_1 - w_2 - w_3]$
1 0 1	$\frac{1}{2}[w_0 + w_1 - w_2 + w_3]$
1 1 0	$\frac{1}{2}[w_0 + w_1 + w_2 - w_3]$
1 1 1	$\frac{1}{2}[w_0 + w_1 + w_2 + w_3]$

Fig. 2: OBC scheme based 4-tap DA FIR Filter implementation [2]

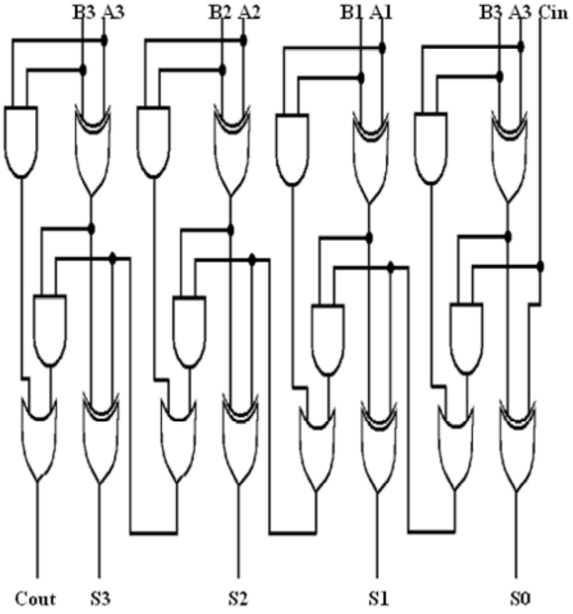


Fig. 3: Regular 4-bit Ripple carry adder [9]

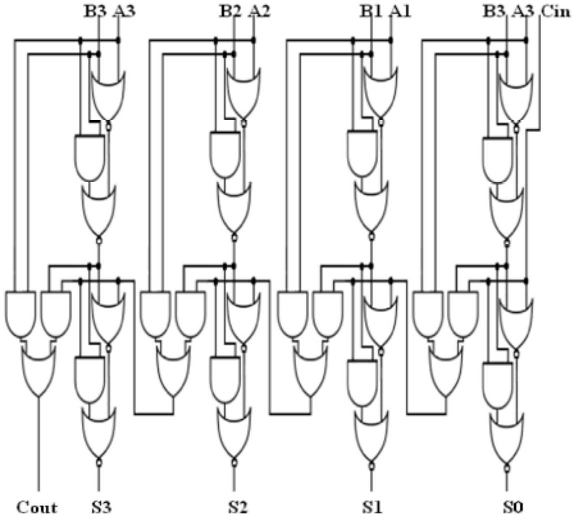


Fig. 4: Proposed 4-bit Ripple Carry Adder

$$y(n) = \sum_{j=0}^{B-1} p_{B-1-j} 2^{-j} - p_{initial} 2^{-(B-1)} \tag{7}$$

Now for "w_k", the "p_{B-1-j}" will take one of 2N combinations, half of them are symmetric about the other half. Thus now only 2^{N-1} combinations are stored in LUT and address to access them are obtained through Exclusive-OR logic of all the LSBs with newest sample LSBs [2]. OBC scheme based 4-tap DA filter implementation is shown in Figure 2.

Datapath Arithmetic Architecture: Datapath components used in DA filtering are multiplexers, inverters, shifters and adders. Among these components, adders are the important elements which decide the performance as they are in the critical path. Due to its presence in large numbers in the datapath, efficiency of such a component would impact largely at the datapath arithmetic and in filtering when DA based method is approached. Hence in this brief, low power adder architecture is proposed for the datapath arithmetic which impacts largely at the filter level.

Figure 3 and Figure 4 shows the architectures of the regular and proposed adders which are inferred in the DA based filter design. Figures (3, 4) contain 4-bit ripple carry adders for architecture illustration, but the actual design differ in bit-width. The Filter was designed for 4 tap, with 4-bit input samples and 8-bit weights.

Proposed adder architectures minimize the inverter cells in the critical path to avoid the power consumption due to frequent logic transition. Complex cells (like AND-AND-OR and OR-AND) with higher transistor stacks are used to reduce the leakage power; since the higher transistor stacks increases the ON stack resistance and reduces the leakage current between the power supplies (Vdd and Vss) during the standby mode. Complex cells avoid interconnects between the smaller cells and reduces the possible glitches and delays resulting in reduced power consumption.

Since the optimizations are at the architectural level, the proposed adder can be utilized at any level for any design abstractions depending on the constraints. Architectures designed as per the design constraints are more efficient than the regularly used architectures.

RESULTS AND DISCUSSIONS

4-tap DA based Filter was designed as per the ASIC design methodology. Adder and filter designs with existing and proposed adders were modeled using Verilog HDL and verified their functionality using model-sim simulator in waveform editor. Designs were synthesized with Cadence RTL compiler synthesis tool by mapping to TSMC 65nm technological library node. Results of the 8bit ripple carry adder with existing and proposed architectures are tabulated in Table 1. The impact of proposed adder architecture in filter has been reported in Table 2 and is discussed further.

Table 1 shows the results of the existing and proposed adder architecture. From Table I, it can be observed that the proposed architecture outperforms the regular architecture in all the aspects of the design metrics – area, performance and power consumption. As mentioned in datapath arithmetic section, the use of complex cells, results in reduced leakage power and similarly the delay is also reduced. Proposed architecture has reduced the leakage power consumption by 41.39% and delay by 17.43%. This suggests that the architectural changes are important and their impact would be higher when utilized in the hierarchy. The proposed architecture also requires 23.68% less area to implement the functional equivalent architecture and hence suggests that the architecture suits to any design with any constraints.

Table 2 shows the impact of adder architecture at the filter level and suggests that it is efficient than the regular adder architecture. It proves that the architectural changes can impact largely and datapath optimizations can change the efficiency of the design at the higher hierarchical level. The proposed DA based filter design requires 5% less area, provides 8.37% high performance and consumes 10.62% less power than the existing DA based filter architecture. It also suggests that the architectural changes provide efficiency at the individual standalone (adder) level and also at the hierarchical level (filter). Thus the datapath architectural optimizations are generic and can be implemented at any abstract levels.

Table 1: Results of 8-bit adder architecture using regular Full adder cell and Proposed Full adder cell

	Adder		
Design	[9]	Proposed	% gain
Area (Sq. microns)	109.44	83.52	23.68
Delay (ns)	0.998	0.824	17.43
Dp (micro watt)	4.814	3.546	26.34
Lp (micro watt)	1.138	0.667	41.39
Tp (micro watt)	5.952	4.213	29.22

Note: “Dp” is Dynamic power; “Lp” is Leakage power and “Tp” is Total power

Table 2: Results of 4-Tap DA based Filter using regular Full adder cell and Proposed Full adder cell

	DA_Filter		
Design	[2, 9]	Proposed	% gain
Area (Sq. microns)	553.32	525.24	5.07
Delay (ns)	2.174	1.992	8.37
Dp (micro watt)	21.262	19.069	10.31
Lp (micro watt)	4.242	3.727	12.14
Tp (micro watt)	25.504	22.796	10.62

Note: “Dp” is Dynamic power; “Lp” is Leakage power and “Tp” is Total power

CONCLUSION

Low power distributed Arithmetic based 4-Tap filter was implemented in this brief. Proposed DA based filter architecture has reduced the delay by 8.37% and consumed 10.62% less power with 5% less area. The importance of datapath optimizations and its impact at the filter level is addressed in this brief. Proposed results suggest that constraint specific architectural level optimizations can be utilized at any level for any design abstractions.

REFERENCES

1. Mohanty, Basant, K. and Pramod Kumar Meher, 2013. A high-performance energy-efficient architecture for FIR adaptive filter based on new distributed arithmetic formulation of block LMS algorithm, *Signal Processing, IEEE Transactions,* 61(4): 921-932.

2. Prakash, M.S. and R.A. Shaik, 2013. Low-Area and High-Throughput Architecture for an Adaptive Filter Using Distributed Arithmetic, Circuits and Systems II: Express Briefs, IEEE Transactions on , 60(11): 781-785.
3. Kumm, Martin, Konrad Moller and Peter Zipf, 2013. Partial LUT size analysis in distributed arithmetic FIR Filters on FPGAs, *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on*. IEEE.
4. Shi, Dong and Ya Jun Yu, 2011. Design of linear phase FIR filters with high probability of achieving minimum number of addersm, *Circuits and Systems I: Regular Papers, IEEE Transactions on.* 58(1): 126-136.
5. Mirzaei, Shahnam, Ryan Kastner and Anup Hosangadi, 2010. Layout aware optimization of high speed fixed coefficient FIR filters for FPGAs, *International Journal of Reconfigurable Computing* 2010.
6. Kumm, Martin and Peter Zipf, 2011. High speed low complexity FPGA-based FIR filters using pipelined adder graphs. *Field-Programmable Technology (FPT), 2011 International Conference on*. IEEE.
7. Sharifi, Fazel, 2014. A Flexible Design for Optimization of Hardware Architecture in Distributed Arithmetic based FIR Filters, *arXiv preprint arXiv, 1403-4554*
8. Rui Guo and L.S. DeBrunner, 2011.Two High-Performance Adaptive Filter Implementation Schemes Using Distributed Arithmetic, *Circuits and Systems II: Express Briefs, IEEE Transactions on*, 58(9): 600-604.
9. Weste, N., 2008. CMOS VLSI Design- A Circuits & System Perspective", Pearson Education.
10. Chandra Mohan, U., 2004. High Speed Squarer, Proceedings of the 8th VLSI Design and Test Workshops, VDAT.