

Distributed Arithmetic Unit Design for Fir Filter

B. Ayyappa Reddy
M.Tech Student,
MITS-Madanapalle.

G. Sambasiva Rao
Asst.prof,
MITS-Madanapalle.

ABSTRACT:

In this paper different distributed Arithmetic (DA) architectures are proposed for Finite Impulse Response (FIR) filter. FIR filter is the main part of the Digital Signal Processing. In Digital Signal Processing we can use Multiply Accumulator Circuit (MAC) and DA for filter design. MAC consumes more power and area because of multiplier and adder circuit. The design distributed arithmetic is run time reconfigurable. The implementation results are provided to demonstrate a high speed and low power architectures. The different DA architecture are implemented in verilog and verified via simulation. In the 16-tap FIR filter design of distributed arithmetic gives better results, 50% of power dissipation and area can be achieved by LUT less2 architecture. 49% of delay can be achieved by separated LUT DA architecture.

Keywords:

Distributed Arithmetic (DA), Finite impulse response (FIR) Filter, Multiply-Accumulate-Circuit (MAC).

Introduction:

In the recent years, there was a developing tendency in order to implement a digital signal processing functions in Field Programmable gate array (FPGA). Finite impulse Response (FIR) filters are most frequent a digital signal processing system unit. FIR filter with exactly linear phase can easily be design. It can be realized in both recursive and non-recursive structure. Generally, direct implementation of an N-tap FIR filter requires multiply and accumulate (MAC) blocks, which are extravagant to implement in FPGA because of resource usage and logic complexity [1]. To determination this issue, first present Distributed Arithmetic, which may be multiplier less architecture? Implementing multipliers while using the reasoning materials from the FPGA will be high-priced because of logic complexity as well as area use, especially when the filter size will be large.

Modern FPGAs have got focused DSP blocks which relieve this concern, but also for substantial filter sizes the battle associated with decreasing area as well as complexity even now remains.

Distributed Arithmetic was introduced by croisier in FIR filters to overcome the difficulties of MAC. DA is a multiplier less architecture based on 2's complement binary representation of data which will pre compute and stored in LUT and bit position reordering [2]. Distributed Arithmetic implementation can be classified in to two ways. Those are RAM-based and ROM-based methods. The 2 multiplier-less techniques are conversion based approach and memories (RAMs, ROMs) or Look-Up table (LUT) techniques.

The LUTs are used to store pre-computed values of coefficient operations [1]. Pre-computation and pre-calculation values are the states stored in LUT in ROM-based which have an impact of low power design. These types of memory based structures are hugely useful in power usage which often uses area in expense of pre-defined in addition to set filter coefficients therefore limits the application scopes. RAM based scheme is usually a substitute way of implement the particular FIR filter. Inside RAM model set filter coefficients usually are stashed because articles which allows adjusting in addition to changing the particular coefficients through the runtime of the filter for several applications. Power consumption and area are the major motivation factors for researchers when compared to ROM-based design.

In the FIR filter design and performance measures three basic boundaries are there. Those are Speed or runtime clock frequency, power as well as area. Power and area improved in the DA comparing to MAC unit. This accomplishment is likewise focused with reconfigurable or adaptable filter configuration to have both of the ROM-based execution and RAM based adaptability. The proposed DA architecture endeavors the circuit exchanging action on the most dynamic and power hungry units.

II. Background Concept:

A. Multiply and Accumulate:

The MAC operation is common in Digital Signal Processing Algorithms. In Digital Signal Processing MAC is the one major unit to design the filter. The MAC unit computes the multiplication of 2 numbers and adds that product to an accumulator [1].

$$p \leftarrow p + (q \times r) \quad (1)$$

The above equation (1) symbolizes the MAC numerical function. The place where p symbolizes the out of accumulator, q is the input and r is the coefficient.

B. Distributed Arithmetic:

Distributed Arithmetic is the extension of multiply and accumulate unit (MAC). It is efficient technique for calculation of inner product or sum of products or multiplies and accumulates. Distributed Arithmetic is a technique that is bit serial in nature. Efficiency of mechanization is the advantage of Distributed Arithmetic (DA).

For sum of product the general equation is:

$$y = \sum_{i=1}^k A_i X_i \quad (2)$$

Where A_i is usually fixed coefficients, X_i is the suggestions info terms, as well as the i is usually volume of inputs. Permit X_i always be N-bits scaled 2's complement number wherever b_{k0} is the sign bit.

$$1 \leq i \leq k, X_i = \{b_{i0}, \dots, b_{i(N-1)}\} \quad (3)$$

We can express X_i as

$$X_i = -b_{i0} + \sum_{n=1}^{N-1} b_{in} 2^{-n} \quad (4)$$

In order to exhibit y in terms of the actual bits of X_i we could substitute equation 4 throughout equation 2. The result as following:

$$y = \sum_{i=1}^k A_i (-b_{i0} + \sum_{n=1}^{N-1} b_{in} 2^{-n})$$

$$= -\sum_{i=1}^k A_i b_{i0} + \sum_{n=1}^{N-1} (\sum_{i=1}^k A_i b_{in}) \cdot 2^{-n} \quad (5)$$

The above equation (5) describes a DA computation. Consider the bracketed term $\sum_{i=1}^k A_i b_{in}$, due to the fact every single trash can will take the actual values involving 0 in addition to 1 only, consequently only 2i achievable values tend to be major. We can easily calculate these types of values on-line (using any RAM), as well as pre-compute the actual values in addition to shop them in the ROM. That input details needs to be used to directly handle the memory and also the output result. Immediately after N like series, the memory affects the output result [3].

C. FIR Filter Implementation:

By using MAC as well as DA units we can implement FIR Filter. Involving of which DA is just about the nearly all recognized methods. The K-length FIR filter can be represented as:

$$y[n] = \sum_{k=0}^K h[k].X[n - k] \quad (6)$$

In which $x[k]$ may be the input information [1, 4] as well as $h[k]$ may be the filter coefficient. Generally, direct implementation of the K-tap FIR filter requires K MAC blocks that's proven within Fig. 1 [1].

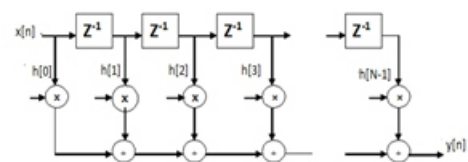


Fig.1. Block diagram of Conventional tapped FIR Filter

The actual down below Fig. 2, one more implementation involving FIR filter will be based upon DA approach. The actual DA architecture consists of 3 parts. DA-LUT, shift register and adder/shifter. The filter coefficients pre-stored and addressed by input data in DA-LUT [1].

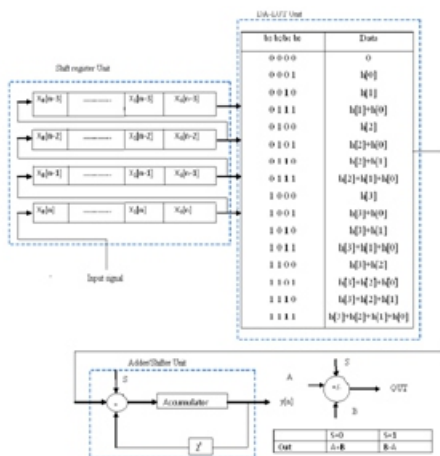


Fig. 2. Original LUT based DA representation of 4-tap FIR filter.

III. Proposed Distributed Arithmetic Unit Design:

A. LUT-Less1 DA Architecture:

In Fig. 2. We can easily see that the lower half of the particular LUT could be the similar using the sum the upper connected with LUT and $h[3]$. By using a 2:1 Multiplexer and an adder can be reduced to half of DA-LUT unit [1, 2]. To reduce the delay carry save adder is replaced with carry look ahead adder.

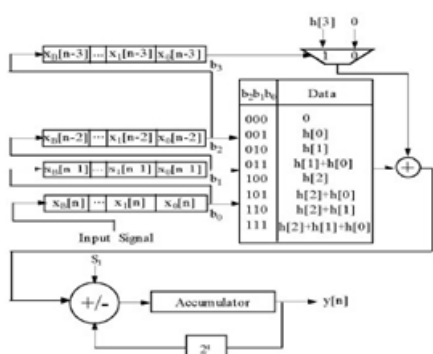


Fig.3.LUT- Less1 DA Architecture for 4-tap FIR filter

B. LUT-Less2 DA Architecture:

From the same LUT decrease process, we have got LUT-Less2 DA architecture. The LUT Less2 DA structures dramatically reduce the actual memory. In this architecture every one of the LUT models are generally replaced simply by mux and adders.

Simply by use of combinational logic circuit the filter efficiency will be damaged. Fig.4. represents that the LUT-Less2 DA architecture for 4-tap FIR Filter.

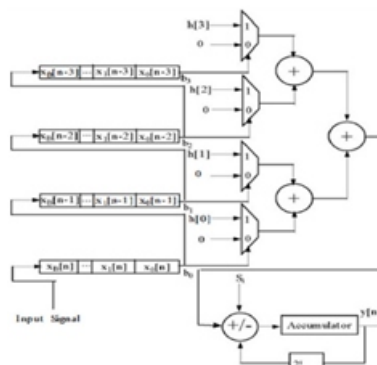


Fig.4. LUT- Less2 DA Architecture for 4-tap FIR filter

C. Separated-LUT DA Architecture:

As the filter size improves the components setup cost of memory in DA architecture develops exponentially. We can break down this k-map FIR straight into N small FIR filters. Hence LUT size reduced to $N \times 2^m$ words. The below figure shows the design of 4-tap FIR filter for separated LUT-DA architecture.

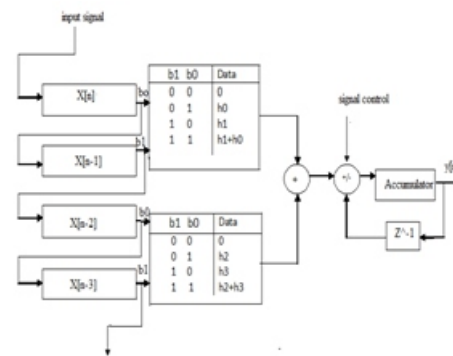


Fig.5.Separated-LUT DA Architecture for 4-tap FIR filter

IV. Synthesis Results:

In order to compare the performance of the various LUT-DA architecture for FIR filter design are described in section 3. Mainly this filter code is usually wrote inside verilogHDL regarding each of the architectures and then synthesize is conducted with cadence design compiler with this purposed this 4-tap, 8-tap&16-tap FIR filter with conventional DA, LUT-Less1, LUT-Less2 and separated-LUT are implemented.

with cadence design compiler with this purposed this 4-tap, 8-tap&16-tap FIR filter with conventional DA, LUT-Less1, LUT-Less2 and separated-LUT are implemented.

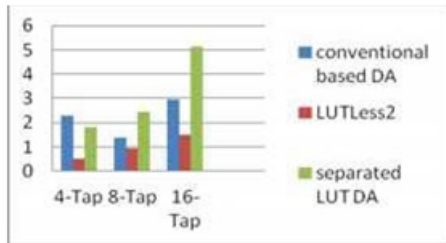


Fig.6. Power Report

B.Delay Mechanism:

The result of delay is shown in below figure. From the Synthesis report delay has reduced in separated LUT DA based architecture.

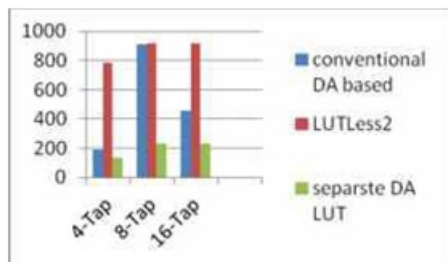


Fig.7. Delay Mechanism

C.Area Report:

The result of area report is shown in fig.8.from the synthesis report area is reduced in LUT Less2 architecture compared with conventional DA based architecture.

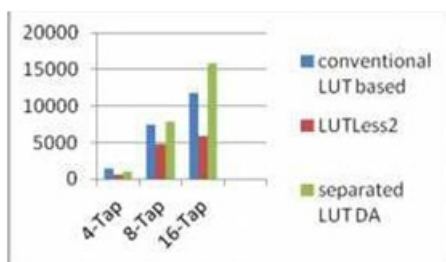


Fig. 8. Area Report

The below table shows the comparison between the 4-Tap, 8-Tap and 16-Tap architecture of FIR filter.

Table 1. Comparisons of power dissipation in mW

Table 1. Comparisons of power dissipation in mW

FIR Filter design	4-TAP	8-TAP	16-TAP
Conventional DA based	2.28	1.362	2.97
LUT Less2	6.58	0.924	1.48
Separated LUT DA	1.8	2.44	5.15

Table 2. Comparison of Delay in ps

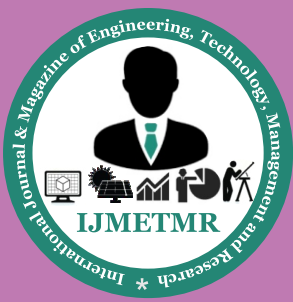
FIR Filter design	4-TAP	8-TAP	16-TAP
Conventional DA based	190	911	459
LUT Less2	782	920	920
Separated LUT DA	551	234	234

Table 3. Comparison of Area in μm²

FIR Filter design	4-TAP	8-TAP	16-TAP
Conventional DA based	1402	7504	11832
LUT Less2	583	4790	5854
Separated LUT DA	995	7983	15777

V. Conclusion:

MAC and DA are commonly used in digital signal processing and filter design. Different DALUT architectures design for FIR filters. These three architectures reduce in different aspects such as power, delay and area. Thus to reduce LUT Size higher order filters divided into several group of small filters. The design of distributed arithmetic unit has the run-time coefficient configurability. The target architecture is design, verified and simulated with verilog HDL for power, delay and area target architecture is synthesise in cadence digital lab. In the 16-tap FIR filter design of our distributed arithmetic gives the better results, LUT-Less2 architecture power dissipation and area improvements are 50%.and separate LUT DA architecture gives 49% of delay reduction.



VI. REFERENCES:

- [1] Wang Sen, Tang Bin, Zhu Jun, "Distributed Arithmetic for FIR Filter Design on FPGA" International conference on communication. Circuits and systems, October 2007, pp.620-623
- [2] AM. AL-Haj, "An FPGA-Based Parallel Distributed Arithmetic Implementation of the 1-D Discrete Wavelet Transform," vol 29, pp. 241-247, February 2004
- [3] N. S. Pal, H.P. Singh, P. I. Sarin, S. Singh, "Implementation of High Speed FIR Filter using Serial and Parallel Distributed Arithmetic Algorithm", International Journal of Computer Applications, July 2011 vol. 25, no. 7, pp. 26-32.
- [4] S. F. Ghamkhari, M. B. Ghaznavi-Ghouschi "A Low-Power Low-Area Architecture Design for Distributed Arithmetic (DA) Unit", 20th Iranian Conference on Electrical Engineering, (ICEE2012), May 15-17, 2012, Tehran, Iran.
- [5] LI Nian-giang, Hou Si-Yu Cui Shi-Yao, "Application of Distributed FIR filter based on FPGA in the analyzing of ECG signal" international conference on intelligent system design and engineering application,(2010IEEE). october,11,2009
- [6] D.J. Allred, H. Yoo , V. Krishnan, W. Huang, and D.V. Anderson, "LMS adaptive filters using distributed arithmetic for high 237 throughput", IEEE Transactions on Circuits and Systems, vol. 52, no. 7,pp. 1327-1337,2005.