Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

Area and Power Efficient Adaptive Fir Filter using Modified Distributed Arithmetic

Priyanka Pandey
SENSE Department
Vellore Institute Technology, Vellore
Tamil Nadu, India

***Abstract*— In this paper, Distributed Arithmetic block consists of look up table, shifter and accumulator instead of multiplier. Addition and shift operation is used for multiplication. As the order of the filter out will increase the LUT size additionally improved, to avoid this problem changed distributed arithmetic method is applied wherein the LUT is subdivided to reduce the memory size. Traditional adder primarily based shift accumulation operation for DA primarily based internal product computation is changed by using conditional signed bring-save accumulation with a purpose to meet the timing and area complexity. The proposed design requires less than half of place and synthesis consequences suggests that the location decreased by means of thing and additionally the LUT partitioning primarily based DA for the adaptive filter gives less electricity intake for filter tap, N=4. The adaptive filter design is implemented in Modelsim Altera and for synthesizing results using Synopsys DC Compiler saed 90nm technology used .**

Keywords*— *Adaptive Filter, Distributed Arithmetic, weight-increment, Look Up Table, area efficiency.

I. INTRODUCTION

In signal processing and telecommunication filters are the basic element and they may be broadly used in programs including noise reduction, channel equalization, audio and video processing, biomedical signal processing, image processing, radar and many others. Filters are used for frequency selection of a network. Primarily based on the input and the output of the clear out they're categorized as analog and digital clear out. And based on their impulse response they're labeled as finite Impulse response (FIR) and infinite impulse reaction (IIR) filter out [1]. The main distinction among FIR and IIR filter is that FIR filter has linear phase it is always stable and it has no analog history whereas IIR filter does not have linear phase and it is difficult to control, IIR is unstable and is derived from analog filter. IIR relies upon on each input and output while FIR depends best on present inputs. IIR includes zero's and poles and require less memory whereas FIR includes best zero's. With advances in digital technology they are unexpectedly changing analog filters wherein RLC components are used. For virtual sign processing operation primary filters are virtual filters. With the usage of special DSP devices they're carried out as multiply and accumulate (MAC) algorithm. The DSP gadgets are based on RISC structure and which includes fast array multipliers. Usually the multipliers are inside the range of 1 to 4 and facts are sequenced by using the microprocessor to pass via it for multiply and different functions and intermediate end result is stored in the accumulator. Performance is progressed by increasing the clock speed used for multiplication. The most powerful operation multiplication is carried out by way of repeated addition which require element hardware portion and chip region and power consumption is likewise excessive. While in comparison with multiply collect structure Memory- based structure are greater normal and having more ability for excessive throughput and has much less latency with less power consumption. For many digital signal Processing (DSP) algorithms, memory-based structures are nicely appropriate for multiplication with constant set of coefficients. For this FIR filter out is designed the use of distributed arithmetic which reduces resource utilization in FPGA implementation.

II. OVERVIEW OF LMS ADAPTIVE ALGORITHM

In this algorithm the filter output and the error value are computed for each cycle which is equal to the difference between the desired output and the filter output.

The equation for weight updating of an n^{th} iteration is –

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \cdot e(n) \cdot \mathbf{x}(n) \qquad (1a)$$

Where,

$$e(n) = d(n) - y(n) \qquad (1b)$$
$$y(n) = \mathbf{w}^{\mathrm{qT}}(n) \cdot \mathbf{x}(n). \qquad (1c)$$

w(n) is the weight vector, e(n) is the error signal, d(n) is the desired response, y(n) is the output signal, μ is convergence factor and x(n) is the input vector.

The weight-update equation for delayed LMS adaptive filter.

$$\mathbf{x}(n) = [x(n), x(n-1), \ldots, x(n-N+1)]^{\mathrm{T}} \qquad (2a)$$
$$\mathbf{w}(n) = [w_0(n), w_1(n), \ldots, w_{N-1}(n)]^{\mathrm{T}}. \qquad (2b)$$

For pipelined structure, error signal is achieved after certain number of cycles, which is called "adaptation delay".

Delayed error e(n-m) is used for updating weight-increment block where m is the adaptation delay.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

III. FIR FILTER WITH DISTRIBUTED ARITHMETIC ARCHITECTURE

The LMS Adaptive Filter computes the inner-product block for each cycle to evaluate the critical path. Let the inner-product of (1c) be –

$$y = \sum_{k=0}^{N-1} w_k \cdot x_k \qquad (4)$$

Changing the order of the summations over the indices k and l, we get

$$y = -\sum_{k=0}^{N-1} x_k \cdot w_{k0} + \sum_{l=1}^{L-1} 2^{-l} \cdot \left[\sum_{k=0}^{N-1} x_k \cdot w_{kl}\right] \qquad (7)$$

Thus, we obtain

$$y = \left[\sum_{l=1}^{L-1} 2^{-l} \cdot y_l\right] - y_0, \qquad \text{where } y_l = \sum_{k=0}^{N-1} x_k \cdot w_{kl}. \quad (8)$$

Any element in N-point bit sequence $\{w_{lk}\}$ for $0 \le k \le N-1$ can be either zero or one, the partial sum y_l for l=0,1,.....L-1 can have 2^N possible values. If all the possible values of y_l are pre-computed and stored in LUT the partial sum y_l can be read by the bit sequence $\{w_{kl}\}$ as address bits for computing inner-product .

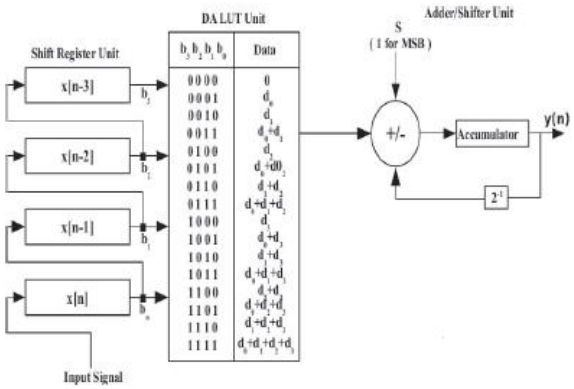

Fig. 1: LUT based Inner Product block

The inner product is calculated of shift accumulation in L cycles which is proceeded by LUT-read operations corresponding to L bits $\{w_{kl}\}$ for $0 \le l \le L-1$.

Since the LUT design consumes larger area for inner product computation hence it is replaced by multiplexers for area efficiency.

The conventional shift adder is replaced with enhanced compressor adder to further reduce area. In the proposed design one enhanced compressor adder is sufficient to compute the inner product instead of three conventional adders for the same purpose.

IV. PROPOSED LUT PARTITIONED BASED DA

The DA-based Adaptive filter design is of length N=4.
It has four-point inner product block and a weight increment block and extra circuitry for computation of error signal e(n) in conjunction with control phrase (t) that is given to barrel shifter .

LUT stores all the possible combos of the sums of the coefficients in 3 dimensional orders. Addition operation is to implement the clear out capability by the use of LUT contents The check in is used to shift the values once more to arithmetic operation block .

The major drawback of this method is that as the order of filter increased then look up table size also increased.

Since we use three dimensional lookup table large amount of values are stored when compared to conventional lookup table.

Let us consider the third order filter, if the number of coefficient is N the memory size required is 2N.

If the number of coefficients = 4
Number of inputs = 4
Then LUL size = 24 = 16 memory locations.

In this method possible outputs are pre computed and stored in LUT .

LUT addressed through input of the filter. For 4 tap filter, 4 tap represents the number of co-efficient of the filter as well as it represents the no. of inputs to the filter and address bit for the LUT.

Each location has different output for the corresponding inputs. The possible inputs for this filter is 0(0000) - 15(1111).For each input the computation of output is easy by using this technique.
If Input = 1011 means then
Output = 1.h0 + 0. h1 + 1. h2 + 1.h3 = h0+h2+h3
If Input = 1111 means then
Output = h0+ h1+h2+h3
If Input = 0101 means then
Output = h1+h3
If Input = 1010 means then
Output = h0+h2

It represents the addition of high level input co-efficient. We can easily find 16 output for corresponding input without any mathematical calculation.

The most significant bit of the error signal is ignored so that the multiplication of input x_k by the error is done by right shifter. The control word for the barrel shifter is obtained from the magnitude part of the error signal .

The convergence factor μ =1/N and is taken to be O(1/N). Here for N=4 tap filter, $4=2^2$ the power 2 determines the no. of shifts that is input to the barrel shifter .

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

The weight increment block has four-barrel shifter along with four adder/subtract or unit. It shifts the input values x_k for k=0,1,.....N-1. The barrel shifter produces the incremented value that might be added to or subtracted from the current weight. The sign bit of error signal is used as the control bit which determines whether to perform addition/subtraction with the current value in the register. Fig. 3 shows the conventional DA based LMS Adaptive filter with N=4 .

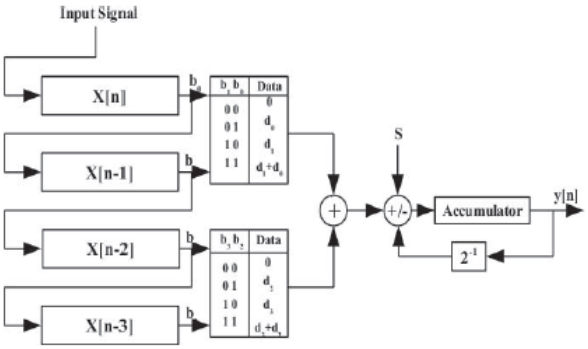

Fig. 2: Partitioned LUT based DA Block

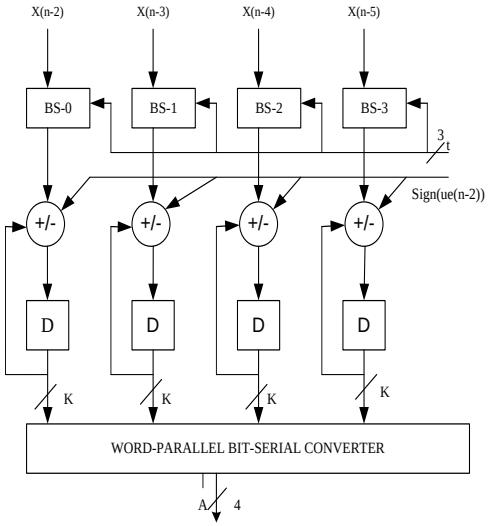

Fig 2: Weight increment block for N=4

```
if r6=1 then t=”000";
else if r5=1 then t=”001";
else if r4=1 then t=”010";
else if r3=1 then t=”011";
else if r2=1 then t=”100";
else if r1=1 then t=”101";
else if r0=1 then t=”110";
   else  then t=”111";

   r=abs(ue(n-2))
ri: ith bit of 7-bit word r
```

Fig. 4: Control word generation Logic for Barrel Shifter

The computation of inner- product in equation (4) can be decomposed into N/P (assume that N=PQ, where Q is assumed to be 2^n, and n is a positive integer) small adaptive filtering blocks of filter length P as

$$y = \sum_{k=0}^{P-1} w_k \cdot x_k + \sum_{k=P}^{2P-1} w_k \cdot x_k \cdots + \sum_{k=N-P}^{N-1} w_k \cdot x_k. \quad (10)$$

The filter coefficients are updated by the corresponding weight increment block. The updated coefficient values are connected to the selection line of multiplexers present in the inner product block. 4-point inner product blocks each of length P=4.

The (L+2) bit sum of products of the filter coefficients and input vectors are produced by the four blocks of inner product blocks.

The four output of the four inner product blocks are summed up together using enhanced compressor adder and the sum thus obtained is compared with the desired input value.

The error, e(n) (difference between actual result and the desired result) is then treated for generating the updated filter coefficients.

Since μ is of the order of 1/N i.e., μ=O(1/N), it can be assumed that μ =1/N. for the filter tap N=16, we truncate the four LSB of e(n) to make the word length to be sign-magnitude to be L bit. Performance of the filter does not get affected much, since the truncation is happened from the LSB.

V. SYNTHESIS RESULT

Power, Area and timing have been anticipated for the proposed design and comparative evaluation is accomplished with the present structure. The proposed design contains best subdivision LUT, that is the main issue for reduce reminiscence size. Conventional adder based totally shift accumulation operation for DA based inner product computation is changed with conditional signed carry-save accumulation on the way to meet the timing and area complexity. The existing model and the proposed design has been synthesized by Synopsys Design Compiler using 90nm SAED CMOS technology library for filter tap N=4 compute the area, delay and power .

Table 6.1 Summary of Area and Power Report

Comparison of Design		
Technique of DA	Area (sq.um)	Power (mW)
DA (Prakash & Shaik (2013))	20981	2917
Modified LUT Partitioned DA	10490	1.33

VI. CONCLUSION

Finite Impulse Response filter performs an important function in lots of Digital Signal Processing applications. On this approach, the multiplier less FIR filter is implemented the usage of disbursed arithmetic which includes look Up

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 8 Issue 05, May-2019

desk and then partitioning is involved. Memory get right of entry to time is much less than multiplication time. LUT partition reduces memory requirements. This approach reduces the delay, area, power consumption. The overall performance can be in addition progressed by means of pipelining all the partial tables. This architecture affords an efficient area-time power implementation which includes appreciably much less latency and less area – delay complexity whilst as compared with current systems for FIR filter

VII. REFERENCES

[1] "Low-Power, High-Throughput, and Low-Area Adaptive FIR Filter Based on Distributed Arithmetic" Sang Yoon Park, Member, IEEE, and Pramod Kumar Meher, Senior Member, IEEE."

[2] "LUT optimization for Distributed Arithmetic based least Mean Square Adaptive Filter" by Basant K. Mohanty, senior member,IEEE, 2015

[3] Mohd Tasleem Khan , "A new High Performance VLSI Architecture for LMS Adaptive filter using Distributed Arithmetic"

[4] "Least Mean Square Adaptive Filters" by S.Haykin and B.Widrow, Hoboken K J, USA: Wiley,2003.

[5] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, "LMS adaptive filters using distributed arithmetic for high throughput," IEEE Trans. Circuits Syst. I, Reg. Papers, vol. 52, no. 7, pp. 1327–1337, Jul. 2005."