Contemporary Engineering Sciences, Vol. 7, 2014, no. 7, 345 - 350
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ces.2014.4212

Distributed Arithmetic Based Non Recursive Filter for High Throughput and Low Power Applications

R. Sathiyapriya and K. Hariharan

SASTRA University, India

Abstract

The proposed work demonstrates a design of adaptive nonrecursive filter design based on distributed arithmetic technique. First, the efficient pipelined distributed arithmetic technique has been proposed here, it achieves low power and reduced switching activity than the conventional one. Second, to attain high throughput the parallel processing based distributed technique is designed. Analysis has been made between the first and second model experimental result shows pipelined DA technique increasing the speed with reduced switching activity and to achieve better area, parallel processing technique will be the efficient one. The proposed design is simulated in modelsim 6.5b by using VHDL, synthesized in Xilinx ISE 14.2 and implemented in FPGA Spartan-3E.

Keywords: Finite Impulse Response(FIR), Distributed Arithmetic(DA), VHSIC Hardware definition language(VHDL), Integrated Simulated Environment(ISE), Field Programmable Gate Array(FPGA).

1. Introduction

Finite Impulse Response (FIR) filters are rapidly used as one of the most significant components in real time Digital signal Processing applications such as cancellation of echo in signal processing, multipath delay compensation, etc. The coefficients of adaptive filters are varying with time in order to optimize a given standard. Least Mean Square (LMS) algorithm is the commonly used algorithm to adapt the coefficients. In a past few years, the distributed arithmetic based multiplier less FIR filter achieves high-throughput, reduced area and time efficient computing architecture [2]. For a large order Adaptive digital filter implementation, the area delay product will be prominently high. To avoid the delay, Block LMS algorithm has been suggested by Mohanty [1] in that a newly modified look up table sharing technique is used. In Digital Signal Processing algorithm the computation of inner product has long critical path, in order to reduce the critical path [3] distributed arithmetic technique has introduced for reconfigurable FIR filter. The paper [4-6] filtering and weight updating has done by using only one look up table for to update the weight of the filter using Least Mean Square algorithm.

The proposed FIR filter design using Pipelined and parallel Distributed arithmetic technique achieves,

1.1 Power consumption and switching activity will be reduced by using pipelined Distributed arithmetic technique.

1.2 Reduction of area consumption has been achieved by parallel processing based Distributed arithmetic technique.

In the forthcoming stage, a brief description of Adaptive LMS algorithm will be given, followed by the review of proposed pipelined DA technique to reduce power then the FIR filter using parallel processing based DA technique. In the following section, discussion about the results of the proposed structure and conclusion.

2. LMS Algorithm for adaptive filter

The LMS algorithm implements an adaptive FIR filter object that gives the output of the filter, the error vector and the weight of the filter. The error vector is the difference between the filter output and the desired response. In every training cycle, the calculated error vector will be used for weight updating. In the LMS adaptive FIR filter the weights are updated by the,

$$W(n) = \alpha\, W(n-1) + f(x(n), e(n), \mu) \qquad (1)$$

Where,

$$y(n) = W^{T}(n-1) \,.\, x(n) \qquad (2)$$

$$e(n) = d(n) - y(n) \qquad (3)$$

d(n) will be the desired response, y(n) will be the output of the filter and e(n) refers the error vector calculated at n^{th} iteration,μ denotes the convergence vector.

for LMS,

$$f(x(n), e(n), \mu) = \mu\, e(n) * x(n) \qquad (4)$$

for sign error LMS,

$$f(x(n), e(n), \mu) = \mu\, \mathrm{sign}\,(e(n) * x(n)) \qquad (5)$$

sign data LMS,

$$f(x(n), e(n), \mu) = \mu\, e(n)\, sign(u^*(n)) \qquad (6)$$

3. Pipelined Distributed Arithmetic Technique

For each cycle, to compute the inner products in LMS adaptive FIR filter which leads to the critical path. Distributed Arithmetic (DA) will be a competent technique for to compute the inner product and it is an efficient technique to implement on field Programmable Gate Array (FPGA). In the conventional distributed arithmetic based adaptive FIR filter of order length 4, it consists of product block for to compute the partial sums for $0<l<15$ and this sum of products are stored in an array of 15 registers. The 16*1 multiplexer will be used to select the partial sums from one of those registers. In that, the weights are given as the selection line i.e., the control bits to the multiplexer, according to the control bits it selects the input from the DA table figure.1.The partial sum terms compiled in a carry save accumulator and then a shift occurs and it gives a sum and a carry word in next cycle figure 2.Finally, the error signal is calculated by subtracting the desired signal with the output of the filter. Then the computed error signal is given to the sign and magnitude separator to get the sign error (sign (μ e (n-2)) and the control bits (t) for barrel shifter.The proposed FIR filter design based on pipelined distributed arithmetic technique as shown in Figure.3. pipelining will lead to a reduction in a critical path, either it increases the clock speed or it will reduce the power.In the proposed pipelined Distributed arithmetic table the dual multiplexer has been used to select the inner products i.e., the partial sums. The counter will be used as the selection line for the multiplexer corresponding to that counter value it gives the partial sum to the latch, the counter value will be incremented from 0 to 14. The output will be driven to the other multiplexer then to carry save accumulation adder.

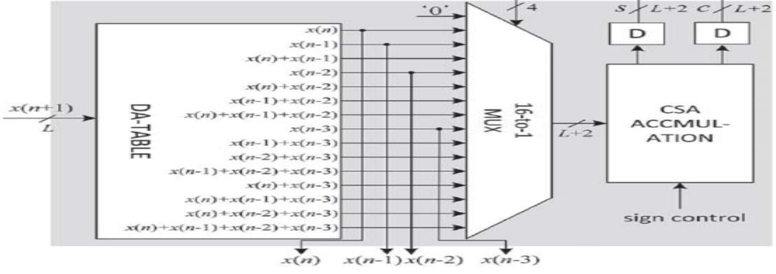


Figure.1.Conventional DA Table for the computation of partial sums

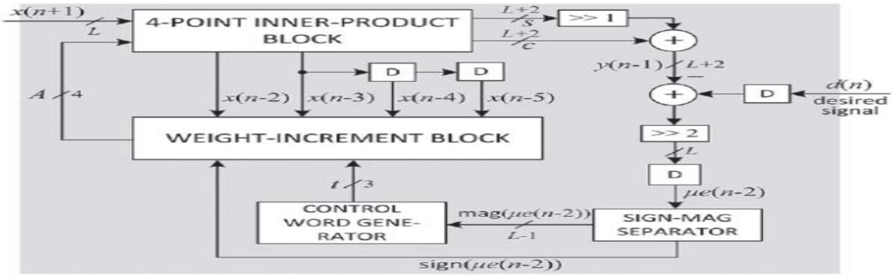


Figure.2. Conventional FIR Filter design based on Distributed Arithmetic Technique.

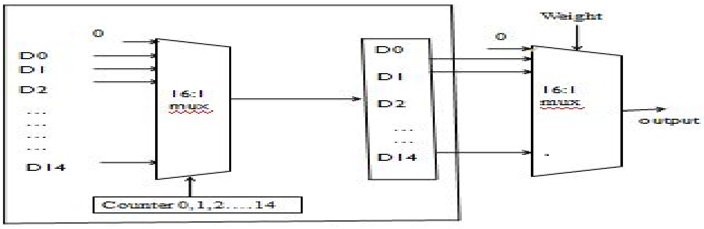


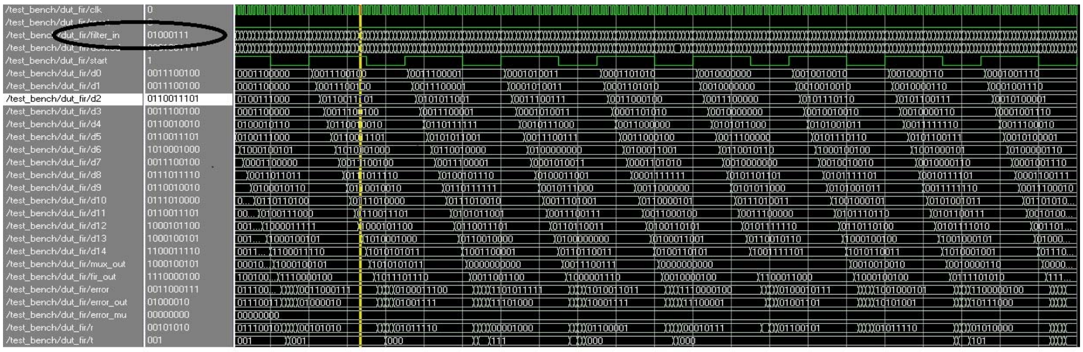
Figure.3. Proposed Distributed Arithmetic table with Pipelining

The pipelined DA table reduces switching activity and hence it reduces power.

4. Parallel processing based Distributed Arithmetic Technique

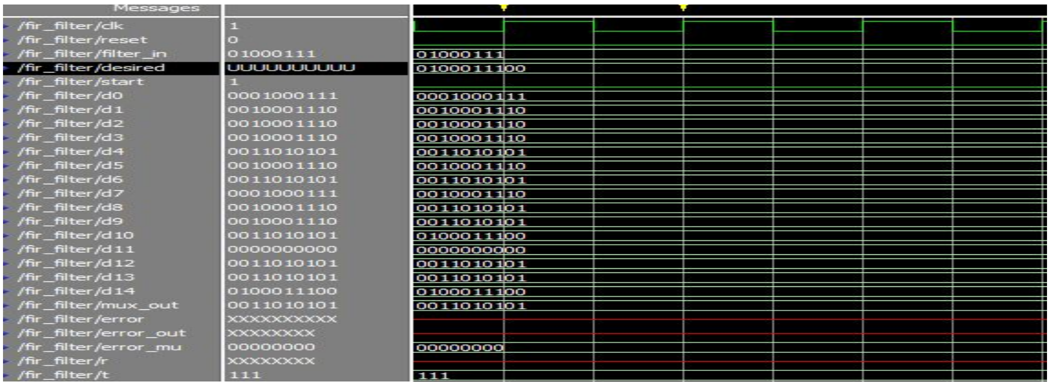
In a parallel processing technique the critical path of the design remains unaltered. The existing FIR filter design uses two clock cycles for the computation of inner products, but in the proposed parallel processing based DA technique uses a single clock cycle to calculate the partial sums from d0 to d14 instead of using 15 clock cycles. Slices of bits i.e.,{w0,w1,w2,w3} will be fed to the multiplexer here it selects the partial sum and the sum and carry bits are generated in a next clock cycle by using carry save accumulation adder(CSA). So that the proposed design attains reduced area with the expense of switching activity and power.

5. Result Analysis and Discussion



```
/test_bench/dut_fir/clk        0
/test_bench/dut_fir/reset      0
/test_bench/dut_fir/filter_in  01000111
/test_bench/dut_fir/desired    [illegible]
/test_bench/dut_fir/start      1
/test_bench/dut_fir/d0         0011100100
/test_bench/dut_fir/d1         0011100100
/test_bench/dut_fir/d2         0110011101
/test_bench/dut_fir/d3         0011100100
/test_bench/dut_fir/d4         0110010010
/test_bench/dut_fir/d5         0110011101
/test_bench/dut_fir/d6         1010001000
/test_bench/dut_fir/d7         0011100100
/test_bench/dut_fir/d8         0111011110
/test_bench/dut_fir/d9         0110010010
/test_bench/dut_fir/d10        0111010000
/test_bench/dut_fir/d11        0110011101
/test_bench/dut_fir/d12        1000101100
/test_bench/dut_fir/d13        1000100101
/test_bench/dut_fir/d14        1100011110
/test_bench/dut_fir/mux_out    1000100101
/test_bench/dut_fir/fir_out    1110000100
/test_bench/dut_fir/error      0011000111
/test_bench/dut_fir/error_out  01000010
/test_bench/dut_fir/error_mu   00000000
/test_bench/dut_fir/r          00101010
/test_bench/dut_fir/t          001
```

Simulation result.1.Distributed Arithmetic Table output with pipelining



```
Messages
/fir_filter/clk          1
/fir_filter/reset        0
/fir_filter/filter_in    01000111
/fir_filter/desired      UUUUUUUUUU
/fir_filter/start        1
/fir_filter/d0           0001000111
/fir_filter/d1           0010001110
/fir_filter/d2           0010001110
/fir_filter/d3           0010001110
/fir_filter/d4           0011010101
/fir_filter/d5           0010001110
/fir_filter/d6           0011010101
/fir_filter/d7           0001000111
/fir_filter/d8           0010001110
/fir_filter/d9           0010001110
/fir_filter/d10          0011010101
/fir_filter/d11          0000000000
/fir_filter/d12          0011010101
/fir_filter/d13          0011010101
/fir_filter/d14          0100011100
/fir_filter/mux_out      0011010101
/fir_filter/error        XXXXXXXXXX
/fir_filter/error_out    XXXXXXXX
/fir_filter/error_mu     00000000
/fir_filter/r            XXXXXXXX
/fir_filter/t            111
```

Simulation result.2. DA Table using parallel processing technique

	Pipelined DA	Parallel Processing in DA
Slices	233	111
LUT	345	196
FF	299	136
Frequency	163.811	123.986

Table.1. Synthesis result for pipelined and parallel processing Technique

The Proposed design is synthesized using Xilinx ISE 14.2, The pipelined Distributed Arithmetic Technique based FIR Filter achieves higher frequency and the parallel processing based DA technique attains low area when compared to pipelined technique.

6. Conclusion and Future work

Analysis has been made between the Distributed Arithmetic techniques with pipelining and parallel processing. For low area applications, the parallel processing technique will be used, for low power and high speed purpose pipelined DA technique accomplish better results than the parallel processing DA based FIR filter. The Future enhancement is to replace the carry save adder with other efficient adders i.e., koggestone adder or ling adder, etc to achieve better result.

References

[1] D. J. Allred, D. V. Anderson, H. Yoo, V. Krishnan, and W. Huang, "LMS adaptive filters using distributed arithmetic for high throughput," . *IEEE Trans.*

[2] Basant K. Mohanty, *Senior Member, IEEE*, and Pramod Kumar Meher, *Senior Member,* "A High-Performance Energy-Efficient Architecture for FIR Adaptive Filter Based on New Distributed Arithmetic Formulation of Block LMS Algorithm" *IEEE* TRANSACTIONS ON SIGNAL PROCESSING, VOL. 61, NO. 4, FEBRUARY 15, 2013.

Circuits Syst. I, Reg. Papers, vol. 52, no. 7, pp. 1323–1337, Jul. 2005.

[3] F.J.Ballester M. Martínez-peiró, R.Gadea, R.J.colom, V.Herrero"Multiplier less FPGA FIR Filter design using a New Signed Common Subexpression Algorithm"

[4] Konrad Moller, Martin Kumm and Peter Zipf "Reconfigurable FIR Filter Using Distributed Arithmetic on FPGAs" 978-1-4673-5762-3/13/$31.00 ©2013 *IEEE* TRANSACTIONS ON SIGNAL PROCESSING, VOL. 61, NO. 4, FEBRUARY 15

[5]L. S. DeBrunner and R. Guo"Two high-performance adaptive filter implementation schemes using distributed arithmetic," R. Guo and L. S. DeBrunner, *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 9, pp. 600–604, Sep. 2011.

[6] L. S. DeBrunner and Rui. Guo "A novel adaptive filter implementation scheme using distributed arithmetic,", in *Proc. Asilomar Conf. Signals, Syst., Comput.*, Nov. 2011, pp. 160–164.

[7] M.Vijaya Laxmi and P.Annapurna Bai, "Design of 128- bit Kogge-Stone Low Power Parallel Prefix VLSI Adder for High Speed Arithmetic Circuits", International Joural of engineering and advanced technology, Issue , 2013.

Received: February 1, 2014