



Unique Journal of Engineering and Advanced Sciences

Available online: www.ujconline.net

Research Article

DESIGN OF ADAPTIVE FILTER BY PARTIAL PRODUCT DELAY MULTIPLICATION IN WALLACE TREE MULTIPLIER

Sudha SA^{1*}, Marimuthu CN²

¹PG student, Dept. of electronics and communication engineering, Nandha Engineering College, Erode-638052, Tamil nadu, India

²Project Guide & Dean, Dept. of electronics and communication Engineering, Nandha Engineering College, Erode-638052, Tamil nadu, India

Received: 23-02-2014; Revised: 21-03-2014; Accepted: 19-04-2014

*Corresponding Author: S.A Sudha

PG student, Dept. of electronics and communication engineering, Nandha Engineering College, Erode-638052, Tamil nadu, India

Email: srisudha.28@gmail.com

ABSTRACT

In this paper, the fir filter is proposed an efficient multiplication stage technique. To investigate the area, speed, power trade-offs for implementation of FIR filters using MCM and digit-serial arithmetic. Multiple constant multiplication(MCM) is an efficient way of implementing several constant multiplications with the same input data. The coefficients of multiplier are expressed using shifts, adders, and subtractions. To introduce the Wallace tree multiplier for reduce both the number of adders and subtractions as well as the number of shifts. This method can be used to reduce the amount of embedded multipliers in large MCM blocks. We use Xilinx 14.5 to provide VHDL coding for our architecture. Result shows the better performance rate of our proposed work than existing algorithms. This paper is used to reduce the delay product and reduce the area and power consumption in FIR filtering.

Keywords: Multiple constant multiplication, Multiplier block, Multiplier less, Digit-serial arithmetic, FIR filter, Low power, Adder graph, Shift-and-add multiplication, Adder depth

INTRODUCTION

Multiplication of a variable by a set of constants, generally known as Multiple Constant Multiplications (MCM), is essential in many Digital Signal Processing (DSP) applications such as, digital Finite Impulse Response (FIR) filters (illustrated in Figure 1), Fast Fourier Transforms (FFT), and Discrete Cosine Transforms (DCT). However, the implementation of a multiplication operation in hardware is considered to be expensive as it occupies significant area and has large delay. Since the constants in multiplications are determined before hand by the DSP algorithms, the full-flexibility of a Multiplier is not necessary and the constant multiplications can be replaced by addition/subtraction and shift operations¹⁻⁵.

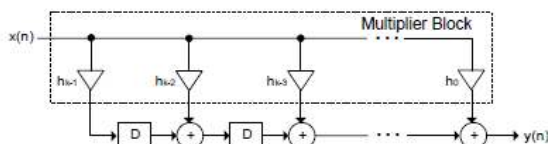


Figure 1: Transposed form of a digital FIR filter design.

For the implementation of constant multiplications using addition/subtraction and shift operations⁶, a straightforward method, generally known as the digit-based recoding⁷, initially defines the constants in multiplications in binary representation. Then, for each 1 in the binary representation of the constant, according to its bit position, it shifts the variable and adds up the shifted variables to obtain the result.

However the implementation of constant multiplications in a shift-adds architecture enables the sharing of common partial products among the constant multiplications, that significantly reduces the area and power dissipation of the MCM design. Hence, the MCM problem is defined as finding the minimum number of addition/subtraction operations that implement the constant multiplications, since shifts can be realized using only wires in hardware. Note that the MCM problem is an NP-complete problem⁸⁻¹⁰.

Most work on implementation of digit-serial FIR filters has focused on implementation in FPGAs and without using multiplier blocks¹¹⁻¹³. However in¹⁴ the digit-size trade-off in implementation of digit-serial transposed direct form FIR filters using direct multiplier blocks was studied.

SYSTEM ANALYSIS

Existing system: In existing system, they proposed fixed point adaptive filter with low adaptive delay for optimized

balanced pipelining across the time-consuming combinational blocks of the structure¹⁵. The adaptation delay of m cycles amounts to the delay introduced by the whole of adaptive filter structure consisting of finite impulse response (FIR) filtering and the weight-update process. Drawback of the existing is to Slow convergence (due to Eigen-value spread).This length of time might cause problems in these applications because the adaptive filter must work in real time to filter the input signals. The steady-state behavior of the LMS algorithm, the step size is small. The DLMS algorithm is reduced the convergence speed and poorer tracking performance. It increases output latency¹⁶.

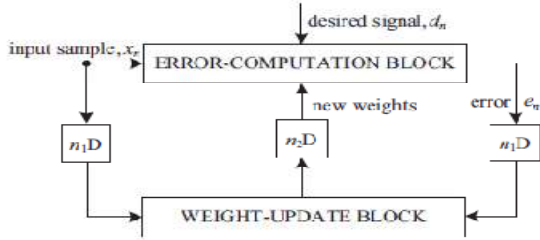


Figure 2. Structure of Delayed LMS Adaptive Filter

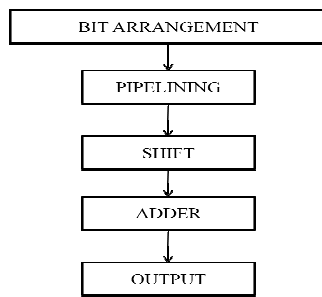


Figure 3. Block Diagram of Existing System

A variation of the direct FIR model is called the transposed FIR filter. It can be constructed from the direct form FIR filter by,

- Exchanging the input and output.
- Inverting the direction of signal flow.
- Substituting an adder by a fork, and vice versa.

Proposed system: In this paper, we propose an optimization method which is able to include a user-defined number of embedded multipliers into a fully pipelined add/shift (PAG) based MCM operation. Here we use Wallace tree algorithm in FIR filters. Beyond that, the method can be used to reduce the amount of embedded multipliers in large MCM blocks which typically appear in floating-point MCM operations. Result shows the performance of our proposed system. Advantages of the proposed is simplicity in implementation. It is stable and robust performance against different signal conditions.

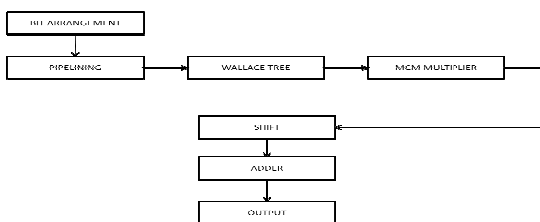


Figure 4. Block Diagram of Proposed System

PIPELINING: Important characteristic of multipliers - allow pipelining. The Pipelining is an implementation technique where multiple instructions are overlapped in execution. The long delay of carry-propagation addition must be minimized. The computer pipeline is divided in stages. Each stage completes a part of an instruction in parallel. The stages are connected one to the next to form a pipe - instructions enter at one end, progress through the stages, and exit at the other end.

Wallace tree multiplier: A Wallace tree is an efficient hardware implementation of a digital circuit that multiplies two integers. The Wallace tree has three steps:

1. Multiply (that is - AND) each bit of one of the arguments, by each bit of the other, yielding results. Depending on position of the multiplied bits, the wires carry different weights.
2. Reduce the number of partial products to two by layers of full and half adders
3. Group the wires in two numbers, and add them with a conventional adder.

A typical Wallace tree architecture is shown in figure.5 below. In the diagram AB0-AB7 represents the partial products.

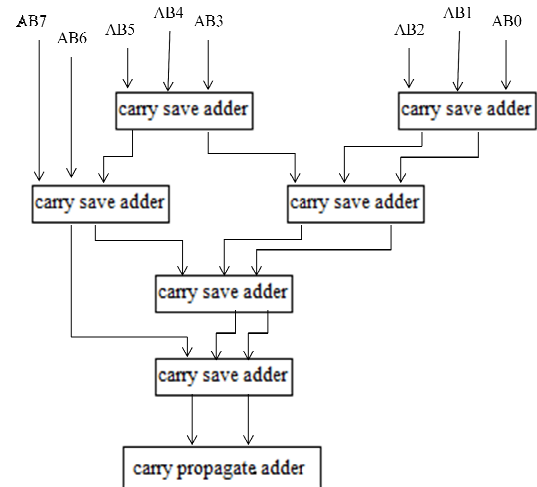


Figure 5: Wallace tree multiplier

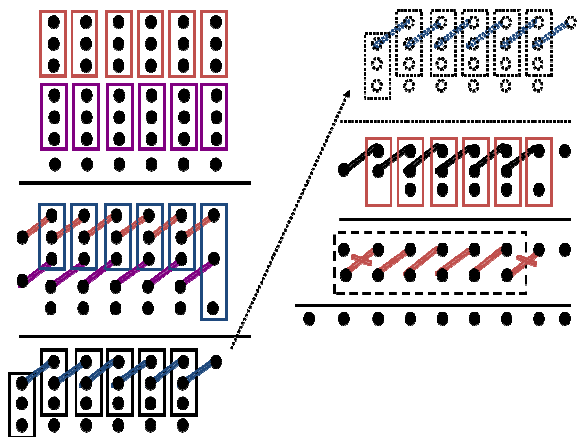


Figure 6: An Example of Wallace Tree Multiplier

PROPOSED ALGORITHM

As done in algorithms designed for the MCM problem given in Definition 1, in our RPAG algorithm. The advantages of

RPAG algorithm: The main factor of power consumption is adder depth. To reduce the adder depth, the power consumption is reduced. Simplicity in implementation, Stable and robust performance against different signal conditions, To reduce the area and delay the critical path is short and high speed performance.

Listing 1. RPAG Algorithm

```

1 RPAG(T)
2 S := max_{t in T} AD_min(t)
3 X_S := {odd(t) | t in T} \ {0}
4 for s = S...2
5   W := X_s
6   P := {}
7   do
8     p ← best_single_predecessor(P,W,s)
9     if p ≠ 0
10      P ← P ∪ {p}
11      W ← W \ A_*(P)
12    else
13      (p1,p2) ← best_msd_predecessor_pair(W,s)
14      P ← P ∪ {p1,p2}
15      W ← W \ A_*(p1,p2)
16  while |W| ≠ 0
17  X_{s-1} ← P
    
```

A. Construct MCM:

The pseudo code of the proposed reduced pipelined adder graph (RPAG) algorithm is shown in Listing 1 and is described below. In contrast to most existing MCM algorithms, the Wallace tree algorithm searches from the output nodes of the adder graph to the input node 1 in a greedy manner, i.e., starting from stage $s = S$, the algorithm searches locally for the best element(s) of the stage before. Initially, the number of stages S is set to the maximum of the minimal necessary AD of the target set T and the output set X_S is filled with the odd values (fundamentals) of T . The minimal AD of an integer x can be directly computed.

B. Evaluating predecessors:

To evaluate a predecessor p , a gain measure is defined which counts the elements of the adder set $A_s(p)$ and the register set $R_s(p)$ produced by p , weighted with the inverse cost of each element. This measure prefers predecessors that are able to produce the most elements in W but also respects predecessors which produce elements in W in a cheap way (like simple registers). The gain for a predecessor pair $(p1; p2)$ is defined in the same way but divided by two, as twice the amount of predecessors is necessary.

C. Search for the Best Single Predecessor:

Instead of evaluating all possible predecessor values $p = 1, 2, \dots, 2b_{max}+1$ with $AD_{min}(p) < s$, which would be very time consuming, the predecessors can be directly computed using the three possible graph topologies. These elements can be copied to P and are realized using pure registers. In topology (b), an element from W is computed from a single predecessor by multiplying it with a cost-1 coefficient of the set $C1 = \{2k \pm 1 | k = 2 \dots b_{max}\}$.

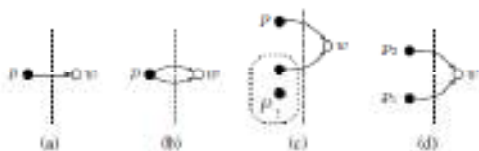


Figure 7: Predecessor graph topologies for (a)-(c) a single predecessor p, (d) a predecessor pair (p1,p2)

D. Search for the Best Predecessor Pairs:

The search for all possible predecessor pairs $(p1, p2)$ of a given number w as shown in topology (d) of Fig. 7 is not a trivial task. Again, all $p1 = 1 \dots 2b_{max}+1$ with $AD_{min}(p1) < s$ and all $p2 \in A(p1;w)$ which fulfill $AD_{min}(p2) < s$ could be evaluated. This method was originally used but is very time consuming. As the search for a single predecessor will find a valid predecessor in most cases, it was decided to limit the possible predecessor pairs to those that can be directly extracted from the MSD representation of w .

RESULTS

The RPAG algorithm was configured to randomly select the 1st or 2nd best solution whereas 50 runs per filter instance were performed. The Result of RPAG is compare to CSE algorithm in MCM system. This show the performance level of our propose system.

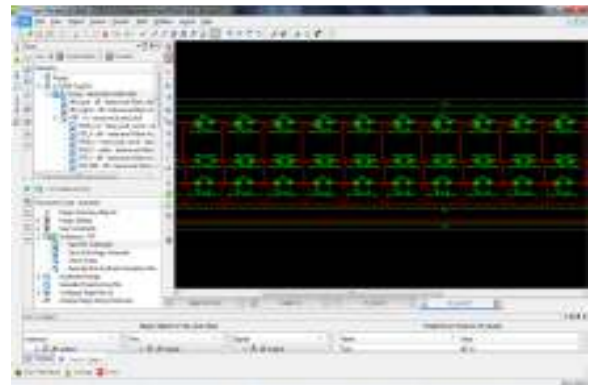


Figure 8: Existing System View RTL Schematic Diagram



Figure 9: Existing System Simulation Output

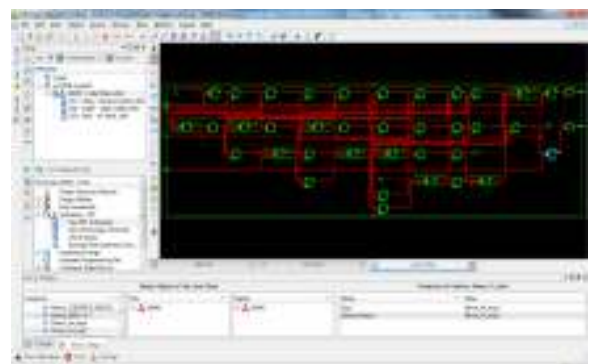


Figure 10: Proposed System View RTL Schematic Diagram

CONCLUSION

In this paper implementation of low power digit-serial FIR filters using multiple constant multiplication (MCM) techniques has been considered. Some conclusions regarding design guidelines for low power digit-serial multiplier blocks can be deduced. The actual complexity in terms of adder cost and number of shifts is not the main factor determining the power consumption. Instead the adder depth, as for parallel arithmetic, is a main contributor. Hence, an algorithm with low adder depth should be used. Furthermore, the shifts prevent glitch propagation through subsequent adders. For even coefficients the shifts can be placed either before or after the final additions. Hence, a heuristic for placing the shifts would be also useful. Better results were achieved using RPAG compared to optimally pipelined adder graphs for FPGAs [10] and the method presented in [9] targeting ASICs. Furthermore, it was shown that the algorithm often finds better solutions for minimal total AD compared to Hcub.

REFERENCES

1. Yi Y, Woods R, Ting LK, Cowan CFN. High speed FPGA-based implementation of delayed-LMS filters. *J. Very Large Scale Integr. (VLSI) Signal Process.* 2005; 39: 113.
2. Ting LK, Woods R, Cowan CFN. Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers. *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.* 2005; 13: 86.
3. Meher PK, Maheshwari M. A high-speed FIR adaptive filter architecture using a P. K. Meher and M. Maheshwari, A high-speed FIR adaptive filter architecture using a modified delayed LMS algorithm in *Proc. IEEE Int. Symp. Circuits Syst.* 2011; 121.
4. Meher PK, Park SY. Low adaptation-delay LMS adaptive filter part-I: Introducing a novel multiplication cell in *Proc. IEEE Int. Midwest Symp. Circuits Syst.* 2011; 1.
5. Meher PK, Park SY. Low adaptation-delay LMS adaptive filter part-II: An optimized architecture, in *Proc. IEEE Int. Midwest Symp. Circuits Syst.* 2011; 1.
6. FIR suite, Suite of constant coefficient FIR filters. Available: <http://www.firsuite.net>. 2011.
7. Parhi K. *VLSI Digital Signal Processing Systems: Design and Implementation.* John Wiley & Sons. 1999.
8. Dempster G, Macleod MD. Use of minimum-adder multiplier blocks in FIR digital filters, *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.* 1995; 42(9): 569.
9. Kumm M, Zipf P. High speed low complexity FPGA-based FIR filters using pipelined adder graphs, in *Field Programmable Technology, International Conference on (ICFPT).* 2011.
10. Aksoy L, Costa E, Flores P, Monteiro J. Optimization of gatelevel area in high throughput multiple constant

- multiplications, in *Proc. European Conf. on Circuit Theory and Design (ECCTD), Linkoping Sweden.* 2011; 29: 609.
11. Mirzaei S, Kastner R, Hosangadi A. Layout aware optimization of high speed fixed coefficient FIR filters for FPGAs, *International Journal of Reconfigurable Computing.* 2010; 3: 1.
 12. Meyer B, Chen J, Chang CH, Dempster AG. A comparison of pipelined RAG-n and DA FPGA-based multiplierless filters, *Circuits and Systems. APCCAS 2006. IEEE Asia Pacific Conference.* 2006; 1555.
 13. Voronenko. Spiral website, [Online]. Available: <http://www.spiral.net/hardware/multless.html>. 2011
 14. Faust M, Chang CH. Minimal logic depth adder tree optimization for multiple constant multiplication, in *Proc. IEEE Int. Symp. On Circuits Syst. ISCAS.* Paris, France. 2010; 457.
 15. Aksoy L, Costa E, Flores P, Monteiro J. Optimization of area and delay at gate-level in multiple constant multiplications,” in *Digital System Design: Architectures, Methods and Tools (DSD).* Euromicro Conference. 2010; 3.
 16. Johansson K. Low power and low complexity shift-and-add based computations, Ph.D. dissertation. Linkoping University. Linkoping Studies in Science and Technology. 2008.
 17. Voronenko Y, Puschel M. Multiplier less multiple constant multiplication, *ACM Trans. Algorithms.* 2007; 3(2): 11.



Figure 11: Proposed System Simulation Output

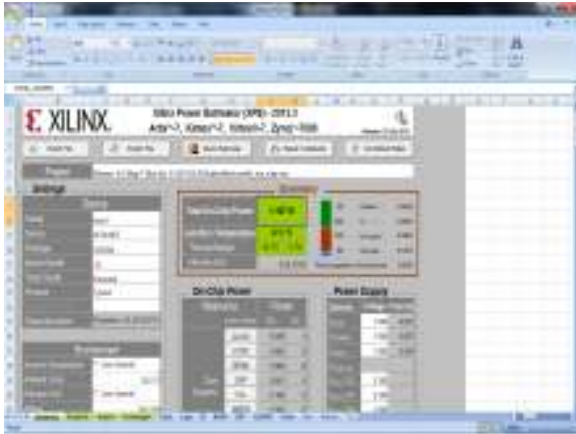


Figure 12: Existing system Xilinx power estimation

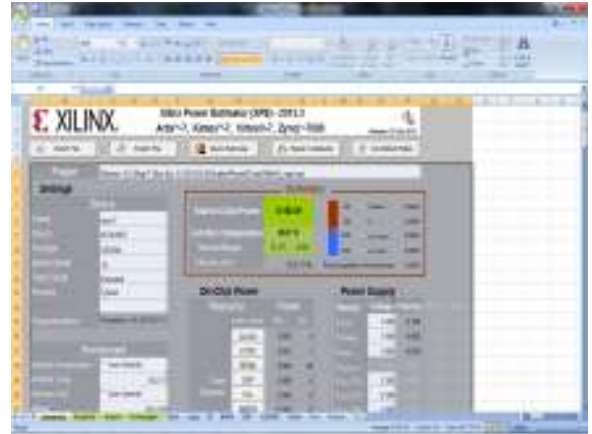


Figure 14: Proposed system Xilinx power estimation

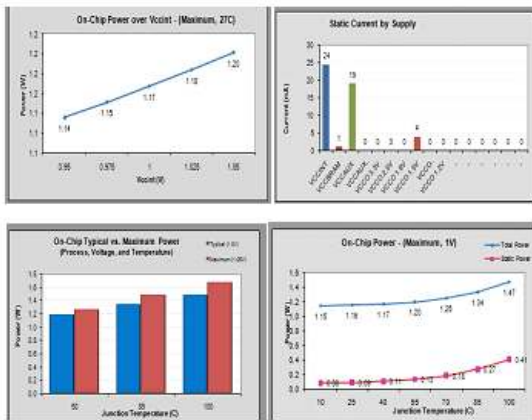


Figure 13: Existing system power analysis

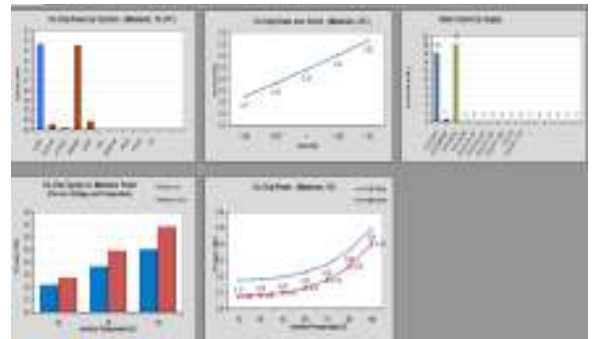


Figure 15: Proposed system power analysis

Table 1: Comparison of Existing System and proposed System

SYSTEM	AREA (No. of FFs)	DELAY in (ns)	POWER in (mW)
EXISTING	184	9.38 ns	1167
PROPOSED	176	6.83 ns	198

Source of support: Nil, Conflict of interest: None Declared