International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 4, April 2014

Design of Area Delay-Power Efficient Adaptive Filter using Wallace Tree Multiplier

Sudha .S .A[1], Marimuthu C. N[2]

[1]PG student, Department of Electronics and Communication Engineering, Nandha Engineering College, Erode-638052, Tamilnadu, India

[2]Project Guide & Dean, Department of Electronics and Communication Engineering, Nandha Engineering College, Erode-638052, Tamilnadu, India

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 multiplications (MCM) are 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, Wallace tree architecture, RPAG algorithm, FIR filter, Pipelining, Partial product generation, Adder depth

1. 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, Fast Fourier Transforms (FFT), and Discrete Cosine Transforms (DCT). An important objective is the reduction of the adder depth (AD), which is defined as the number of adder stages needed to compute a coefficient. The Reduced Pipelining Adder Graph (RPAG) algorithm that minimizes the number of adders while keeping the number of shifts low so as to reduce delay product in Finite Impulse Response (FIR) filter and to reduce the area and power. 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 be forehand 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 operations [12] For the implementation of constant multiplications using addition/subtraction and shift operations, a straightforward method, generally known as the digit-based recoding [7], 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 add 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 problem [4]. Most work on implementation of digit-serial FIR filters has focused on implementation in FPGAs and without using multiplier blocks [11]–[13]. However in [14] the digit-size trade-off in implementation of digit-serial transposed direct form FIR filters using direct multiplier blocks was studied.

2. System Analysis

2.1 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 structure. 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 latency.

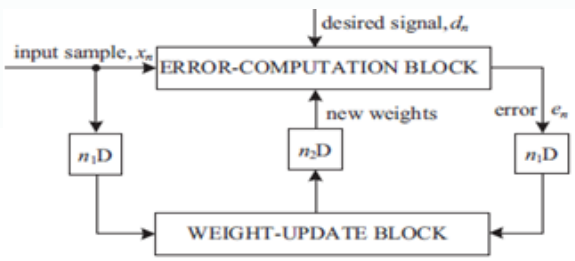

Figure 1: Structure of Delayed LMS Adaptive Filter

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 4, April 2014

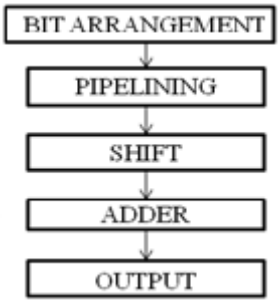

Figure 2: Block Diagram of Existing System

2.2 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. An advantage of the proposed is simplicity in implementation. It is stable and robust performance against different signal conditions.

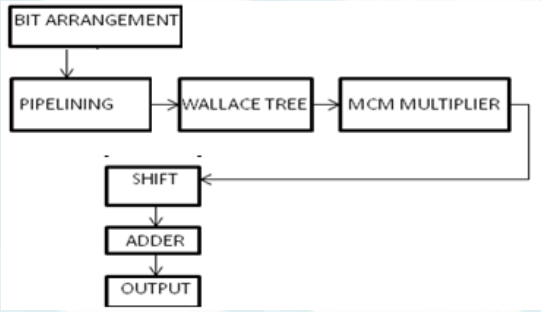

Figure 3: Block Diagram of Proposed System

2.2.1 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.

2.2.2 Multiple Constant Multiplication: The Multiplication of a variable by a set of constants, generally known as Multiple Constant Multiplications (MCM).It is essential in many Digital Signal Processing (DSP) applications,

- Digital Finite Impulse Response (FIR) filters,
- Fast Fourier Transforms (FFT) and
- Discrete Cosine Transforms (DCT).

2.2.3 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:

a) 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.
b) Reduce the number of partial products to two by layers of full and half adders
c) Group the wires in two numbers, and add them with a conventional adder.

A typical Wallace tree architecture is shown in figure 4below. In the diagram AB0-AB7 represents the partial products the partial products.

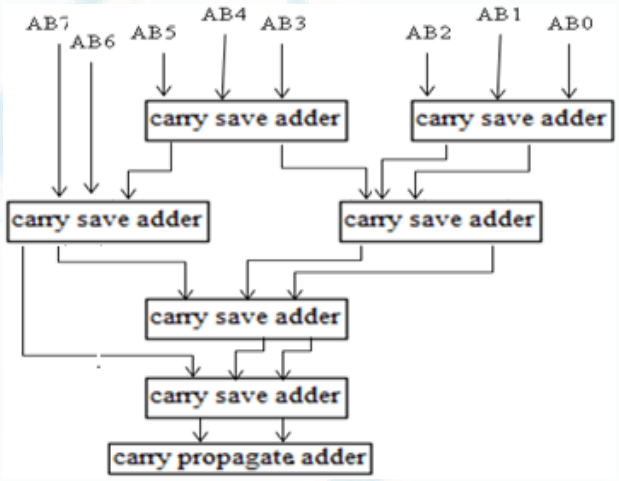

Figure 4: Wallace tree multiplier

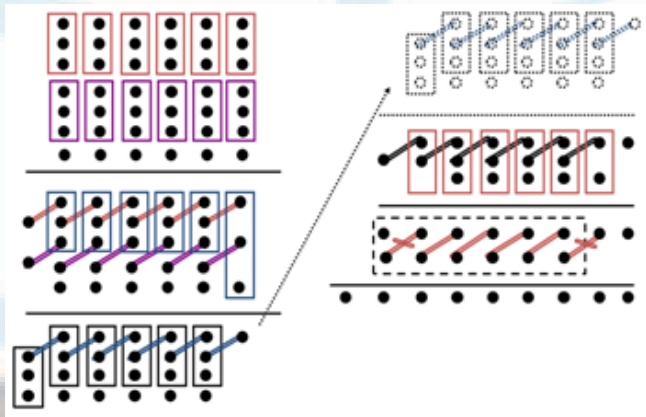

Figure 5: An Example of Wallace Tree Multiplier

2.3.4 Pipelined-Array Multiplier: Array multiplier is well known due to its regular structure. Multiplier circuit is based on add and shift algorithm. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial product are shifted according to their bit orders and then added. The addition can be performed with normal carry propagate adder. N-1 adders are required where N is the multiplier length. The latency of this architecture is low, which is important in many applications. Furthermore, it can easily be pipelined to an arbitrary degree to obtain the desired throughput. To prevent glitches further, pipelining can be introduced, which also increases the throughput. Note that less hardware is needed for pipelining in serial arithmetic compared to the parallel case of multiplier architecture.

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 4, April 2014

2.3.5 Partial Product Generation: The main operation in the process of multiplication of two numbers is addition of the partial products. Therefore, the performance and speed of the multiplier depends on the performance of the adder that forms the core of the multiplier. To achieve higher performance, the multiplier must be pipelined. Throughput is often more critical than the cycle response in DSP designs. In this case, latency in the multiply operation is the price for a faster clock rate. This is accomplished in a multiplier by breaking the carry chain and inserting flip-flops at strategic locations. Care must be taken that all inputs to the adder are created by signals at the same stage of the pipeline. Delay at this point is referred to as latency.

3. 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∈T} ADmin(t)
3    X_S := {odd(t) | t ∈ 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_s(P)
12       else
13         (p1, p2) ← best_msd_predecessor_pair(W, s)
14         P ← P ∪ {p1, p2}
15         W ← W \ A_s(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 XS 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 As (p) and the register set Rs (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… $2b_{max}+1$ with ADmin(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 = $\{2k \pm 1 \mid k = 2 \ldots b_{max}\}$.

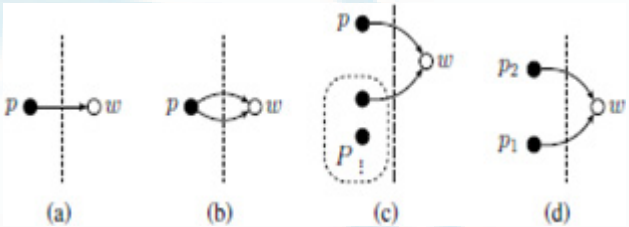

Figure 6: Predecessor graph topologies for (a)-(c) a single predecessor p, (d) a predecessor pair (p1,p2)

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 . . . 2bmax+1 with ADmin(p1) < s and all p2 2 A(p1;w) which fulfill ADmin(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.

4. Result

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.

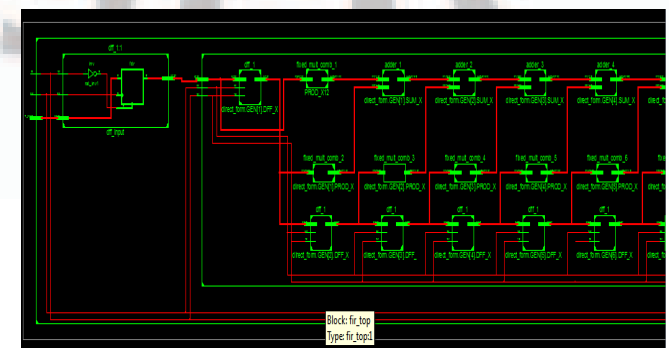

Figure 7: Existing System View RTL Schematic Diagram

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 4, April 2014

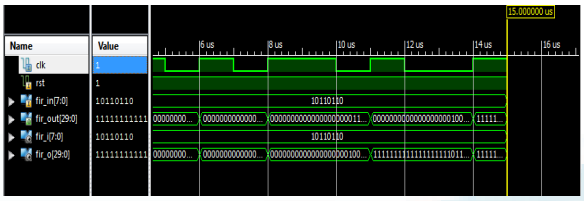

Figure 8: Existing System Simulation Output

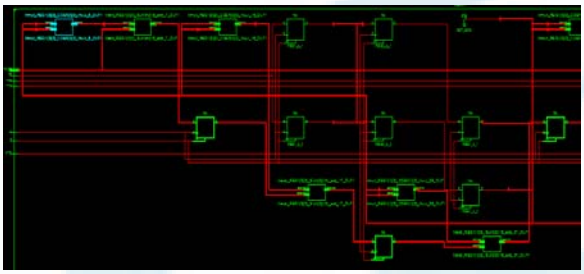

Figure 9: Proposed System View RTL Schematic Diagram

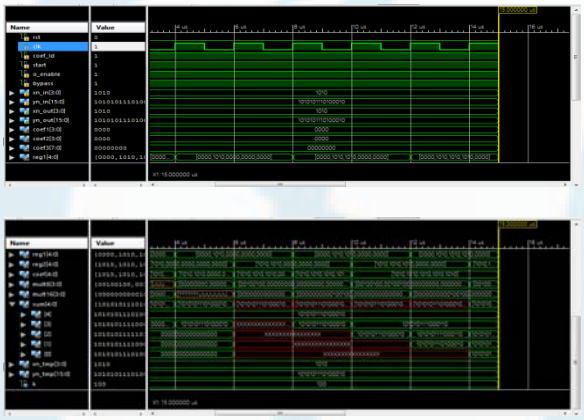

Figure 10.ProposedSystem Simulation Output

Table 1: Device Utilization Summary (Area) of Existing System

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	184	126,800	1%	
Number used as Flip Flops	158			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	26			
Number of Slice LUTs	145	63,400	1%	
Number used as logic	145	63,400	1%	

Table 2: Device Utilization Summary (Area) of Proposed System

Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	74	126,800	1%	
Number used as Flip Flops	74			
Number used as Latches	0			
Number used as Latch-thrus	0			
Number used as AND/OR logics	0			
Number of Slice LUTs	70	63,400	1%	
Number used as logic	59	63,400	1%	

```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

================================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 17.460ns (frequency: 57.272MHz)
  Total number of paths / destination ports: 80640614002948296000 0 / 294
--------------------------------------------------------------------------------
Delay:               17.460ns (Levels of Logic = 46)
  Source:            dff_input/q_5 (FF)
  Destination:       dff_output/q_29 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising
```

Figure 11: Delay and Speed Summary of Existing System

```
Timing Details:
---------------
All values displayed in nanoseconds (ns)

================================================================================
Timing constraint: Default period analysis for Clock 'clk'
  Clock period: 1.976ns (frequency: 505.996MHz)
  Total number of paths / destination ports: 634 / 71
--------------------------------------------------------------------------------
Delay:               1.976ns (Levels of Logic = 15)
  Source:            SUM_3_2 (FF)
  Destination:       SUM_2_15 (FF)
  Source Clock:      clk rising
  Destination Clock: clk rising
```

Figure 12: Delay and Speed Summary of Proposed System

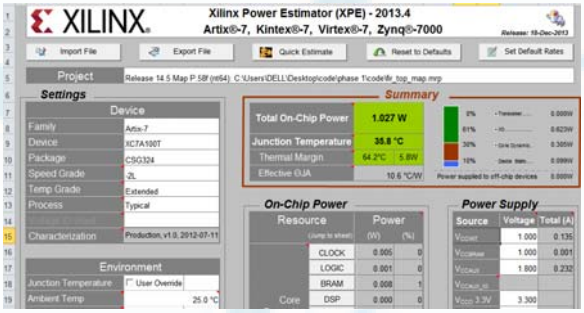

Figure 13: Existing system Xilinx power estimation

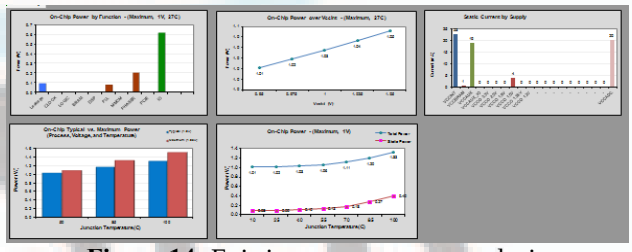

Figure 14: Existing system power analysis

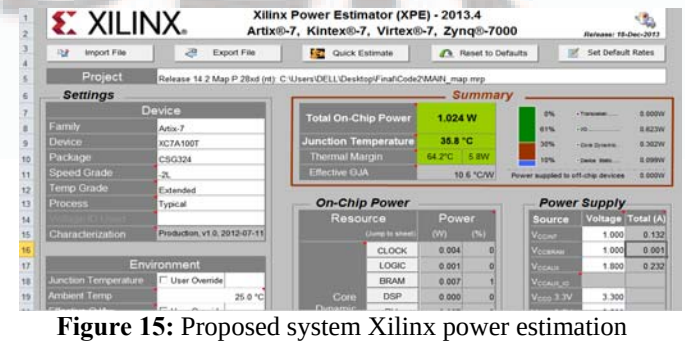

Figure 15: Proposed system Xilinx power estimation

International Journal of Scientific Engineering and Research (IJSER)
www.ijser.in
ISSN (Online): 2347-3878
Volume 2 Issue 4, April 2014

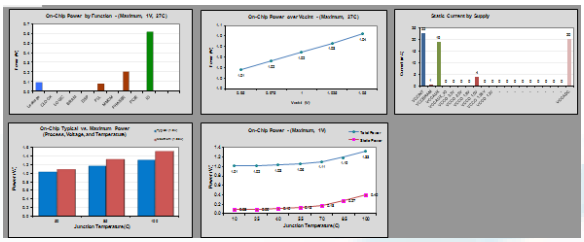

Figure 16: Proposed system power analysis

Table 3: Comparison of Existing System and proposed System

System	*Area (No. of slices and LUTs)*	*Delay in (ns)*	*Power in (mW)*
Existing	329	17.460	1027
Proposed	144	1.976	1024

5. 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 existing system..

References

[1] Y. Yi, R. Woods, L.-K. Ting, and C. F. N. Cowan, "High speed FPGA-based implementations of delayed-LMS filter," J. Very Large Scale Integr. (VLSI) Signal Process, vol. 39, nos. 1–2, pp. 113–131, Jan. 2005.

[2] L.-K. Ting, R. Woods, and C. F. N. Cowan, "Virtex FPGA implementation of a pipelined adaptive LMS predictor for electronic support measures receivers," IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 13, no. 1, pp. 86–99, Jan. 2005

[3] P. K. Meher and M. Maheshwari, "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., May 2011, pp. 121–124.

[4] P. K. Meher and S. Y. Park, "Low adaptation-delay LMS adaptive filter part-I: Introducing a novel multiplication cell," in Proc. IEEE Int. Midwest Symp. Circuits Syst., Aug. 2011, pp. 1–4.

[5] P. K. Meher and S. Y. Park, "Low adaptation-delay LMS adaptive filter part-II: An optimized architecture," in Proc. IEEE Int. Midwest Symp. Circuits Syst., Aug. 2011, pp. 1–4.

[6] FIRsuite, "Suite of constant coefficient FIRfilters,"2011.[Online].Available:http://www.firsuie.net

[7] K. Parhi, VLSI Digital Signal Processing Systems: Design and Implementation. John Wiley & Sons, 1999.

[8] G. Dempster and M. D. Macleod, "Use of minimum-adder multiplier blocks in FIR digital filters," IEEE Trans. Circuits Syst. II, AnalogDigit.Signal Process., vol. 42, no. 9, pp. 569–577, Sep. 1995.

[9] M. Kumm and P. Zipf, "High speed low complexity FPGA-based FIR filters using pipelined adder graphs," in Field Programmable Technology, International Conference on (ICFPT), 2011

[10] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Optimization of gatelevel area in high throughput multiple constant multiplications," in Proc. European Conf. on Circuit Theory and Design (ECCTD), Link¨oping, Sweden, Aug. 29-31 2011, pp. 609–612.

[11] S. Mirzaei, R. Kastner, and A. Hosangadi, "Layout aware optimization of high speed fixed coefficient FIR filters for FPGAs," International Journal of Reconfigurable Computing, vol. 3, pp. 1 – 17, January 2010

[12] U. Meyer-Baese, J. Chen, C. H. Chang, and A. G. Dempster, "A comparison of pipelined RAG-n and DA FPGA-based multiplierless filters," Circuits and Systems, 2006. APCCAS 2006. IEEE Asia Pacific Conference on, pp. . 1555–1558, 2006.

[13] Voronenko, "Spiral website," 2011. [Online]. Available: http://www.spiral.net/hardware/multless.html

[14] M. Faust and C. H. Chang, "Minimal logic depth adder tree optimization for multiple constant multiplication," in Proc. IEEE Int. Symp. On Circuits Syst., 2010. ISCAS 2010, Paris, France, May 30 - Jun. 2 2010,pp. 457–460.

[15] L. Aksoy, E. Costa, P. Flores, and J. Monteiro, "Optimization of area and delay at gate-level in multiple constant multiplications," in Digital System Design: Architectures, Methods and Tools (DSD), 2010 13thEuromicro Conference on, September 2010, pp. 3–10.

[16] K. Johansson, "Low power and low complexity shift-and-add based computations," Ph.D. dissertation, Link¨oping University, 2008, Link¨oping Studies in Science and Technology. Dissertations

[17] Y. Voronenko and M. P¨uschel, "Multiplierless multiple constant multiplication," ACM Trans. Algorithms, vol. 3, no. 2, p. 11, May 2007.