HIGH PERFORMANCE FIR FILTER USING CARRY SELECT ADDER TO REDUCE ARE A-DELAY-POWER

A. Anline Reeta[1], N. Kaleeswari [2]

[1]*Department of Electrical and Electronics, B.E, Ketti (India)*

[2]*Assistant Professor, Department of Electronics and Communication, ME, (India)*

ABSTRACT

Ripple carry implementation allows adder tree to minimize hardware cost, unfortunately it sacrifice timing and gives low speed operation. To outperform this high speed adder is proposed and analyzed for real time speech signal applications. Multiple constant multiplication scheme is the most effective common sub expression sharing technique which is used for implementing the transposed FIR filters. The resource minimization problem in the scheduling of adder tree operation based Mixed integer programming (MIP) algorithm for more efficient MCM based implementation of FIR filters are identified. The proposed adder tree consists of carry select adder to improve the speed of the FIR filter.

***Keywords*: *Carry Select Adder Tree Implementation, Multiple Constant Multiplication, Transposed Fir Filter*.**

I. INTRODUCTION

Digital signal processing technology and its advancements have dramatically impacted our modern society everywhere. Without DSP, we would not have digital audio and speech, Digital telephone, Automobile industry, Electronic communications, Medical imaging equipment, Multimedia applications. The signals are usually processed in digital representation. , so speech processing can be regarded as a special case of digital signal processing the digital filter is the most important system in speech processing. It is used to reduce the noise in an information bearing signal. Based on the impulse response filters are classified as two.

1. Finite impulse response filter (FIR)
2. Infinite impulse response (IIR).

FIR filter is designed using finite number of impulse response. IIR filter does not provide the stable output, So that a design of an efficient transposed FIR filter which produces stable output is produced. The complexity of the FIR filter is dominated by the multiplication of the input samples with filter coefficients. Filters employ a large number of multipliers that lead to excessive area and power consumption. But the filter coefficients are constant for a given filter, so that multiplications are implemented by a network of adders and sub tractors. Where the number of additions and subtractions are minimized by a constant multiplication scheme. In the transposed fir filter, the recent

most input sample at any given clock period is multiplied with all the filter coefficients. A set of intermediate results are generated in this case, and shared across all the multiplications in order to minimize the total number

of additions and subtractions using multiple constant multiplication techniques. Each such intermediate result in an MCM process corresponds to one of the common sub-expressions of the set of constants to be multiplied.

The common sub expression elimination for MCM reveals that the number of operators used to form the adder tree networks is very significant. The number of operators on an adder tree is determined by the number of input terms the coefficient uses from the network. For an N input adder tree, N-1 operators are required. Identify the resource minimization problem in the scheduling of adder tree operations for the FIR MCM block. The area and power consumption of the filters MCM blocks can be calculated and apply the MIP based algorithm for exact bit level resource optimization.

1.1 Related Work

The number of additions used to implement the coefficient multiplications. It determines the complexity of digital filters. Many approaches have been proposed in literature for reducing the number of adders in the multipliers of digital filters. Using coefficient partitioning method to implement low complexity digital filters with minimum number of full adders. While the optimization criterion in conventional low complexity filter implementation method is the number of adders, the focus of this method is to minimize the number of full adders required for each adder. The coefficient partitioning algorithm is combined with the pseudo floating point coefficient coding scheme and applied to optimize the common sub expression elimination methods. The full adder reduction achieved using this method is substantially higher for higher order filters.

The number of adders and critical paths in a multiplier block of a multiple constant multiplication based implementation of a finite impulse response filter can be minimized through common sub expression eliminate techniques. A two bit common sub expression can be located recursively in a non-canonic sign digit representation of the filter coefficients. To improve the elimination of a CS from the multiplier block of an FIR filter. It can be realized with fewer adders and logical depths as compared to the existing methods. This algorithm shows average logical operator with a comparative logic depth requirement.

The two reconfigurable FIR filter architectures, namely Constant shift method (CSE) and Programmable sub expression elimination (PSE). Among these approaches, techniques the best hardware reduction since it deals with the multiplication of one variable (input signal) with several constants (coefficients).The CSE techniques focus on eliminating redundant computations in multiplier blocks by employing the most common sub expressions consisting of two-nonzero bits. The PSM approach is based on the common sub expression elimination algorithm used. Unlike the CSM method where constant shifts are used, the PSM employs programmable shifters. The advantage of PSM over CSM is that the former architecture always ensures the minimum number of additions and thus minimum power consumption. The latency of the adder tree increases by using these approaches.

A new algorithm for digit serial FIR filter using CAD tool proposed in [2]. Little attention has been given to the digit-serial MCM design that offers alternative low complexity MCM operations at the cost of an increased delay. Designing digit-serial MCM operation with optimal area at the gate level by considering the implementation costs of digit-serial addition, subtraction, and shift operation. Since there are still instances with which the exact CSE algorithm cannot cope, the number of adders and logical operators still increases the latency and reduces the throughput of the multiple constant multiplications.

The resource minimization problem in the scheduling of adder tree operations for the FIR multiple constant multiplication block and used the mixed integer programming based algorithm for exact bit level resource

optimization. The power and area reduced in this approaches. After the problem identification of all these implementations to overcome that only we have to implement an efficient FIR filter and proposed the high speed carry select adder in the scheduling of adder tree MCM block for speech processing applications.

II. PROPOSED METHOD

Digital Signal Processing (DSP) deals with the manipulation of digital signals using complex signal processing systems built from basic building blocks like filters. The proposed work implementation in the FIR filter in adder parts. This work evaluates the performance of the improved design in terms of delay and power. So we need to design high speed adders. The ripple carry adder part is replaced with the regular and improved carry select adder and the FIR Filter is implemented and the performance of the design is evaluated in terms of delay and power. The carry-select adder is a particular way to implement an adder, which is a logic element that computes the (n+1)-bit sum of two n-bit numbers. Which is the simple and fastest adder compared to all adders. The carry-select adder generally consists of two ripple carry adders and a multiplexer. Adding two n-bit numbers with a carry-select adder is done with two adders in order to perform the calculation twice, one time with the assumption of the carry being zero and the other assuming one. After the two results are calculated, the correct sum, as well as the correct carry, is then selected with the multiplexer once the correct carry is known.

In the carry select adder initially when en=1, the output of the RCA(Ripple carry adder) is fed as input to the D Latch and the output of the D latch follows the input and given as an input to the multiplexer. When en=0, the last state of the D input is trapped and held in the Latch and therefore the output from the RCA is directly given as an input to the MUX without any delay.

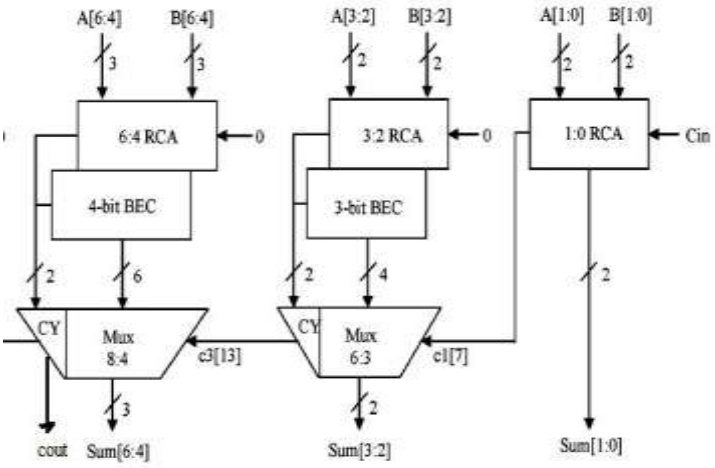

Fig 1. Carry select adder

Now the MUX selects the sum bit according to the input carry which is the selection bit and the inputs of the MUX are the outputs obtained when en=1 and 0. The Adder tree formed using this carry select adder is explained in the below sections.

2.1 Adder Tree Scheduling

In the transposed FIR filter, the recent most input sample at any given clock period is multiplied with all the filter coefficients. A set of intermediate results are generated in this case, and shared across all the multiplications in order to minimize the total number of additions/subtractions using multiple constant multiplication (MCM) techniques. Each such intermediate result in an MCM process corresponds to one of the

common sub-expressions (CS) of the set of constants to be multiplied. The common practice of handling the summation of CS terms of each coefficient is to use the tree height minimization algorithm to produce a height optimum adder-tree. Tree-height minimization algorithm iteratively collapses the pair with smallest delays using an ADD/SUB to form a new term with delay, until a single term is reduced.

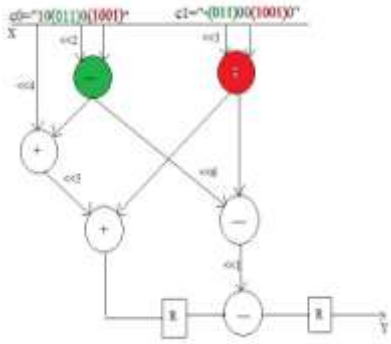

Fig 2. Carry select adder tree scheduling

Note that either a positive or negative sign is associated with each input term, which denotes whether the corresponding term should be added to or subtracted from the summation. These signs also determine whether an addition operation or a subtraction operation should be used when the algorithm collapses a pair of terms in the adder-tree based on the following rules.

(1) If two input edges are of the same sign, an ADD will be used; otherwise, it will be a SUB.

(2) The sign of the output edge is always the same as that of the "left" input edge.

Using these two rules, it is possible that the final term producing the summation result may carry a negative sign, such that a negation is needed after the adder-tree to correct the value. For an FIR filter, results from multiple adder-trees are accumulated by a structural adder register line. So the negation can be eliminated by replacing the structural adder with a sub tractor.

2.2 Cost Model

In order to quantify and minimize the hardware cost of the adder-tree, we model the cost of ADD/SUB operations in this section based on the carry select adder implementation, which is most area efficient and speedup process of scheduling the adder tree, it will be picked up by the hardware compiler whenever the timing allows. Without loss of generality, for a single ADD/SUB operation, the pair of its input operands may be of different bit-widths, and one of them is to be left shifted by certain bit positions. After the common sub expression terms are determined and the ADD/SUB network of non-redundant sub expressions (or terms) is formed, the product value corresponding to each of the coefficients is computed by an adder-tree that sums up its relevant terms. Two adder-trees are formed, for computing the product of a pair of coefficients using shifted versions of unique CS terms „1", „10-1" and „1001" from the term-networks or sub expression-networks. We have developed the cost model of the ADD/SUB network by bit-level analysis, which could be reduced by suitable scheduling of operations on the adder-tree. We find that significant area, latency and power reduction. The cost calculation is done separately in three bit-segments. Starting from the least significant bit (LSB), the 1st segment covers the bit positions up to but not including the first bit of the shifted operand; the 3rd segment covers the bits corresponding to the sign extension bits of the sign extended operand; the 2^{nd} segment takes the rest of the bit positions. The clock performance of the entire FIR filter is decided by the largest of the delays of all coefficients.

Assuming the delay of an ADD/SUB operator to be 1 unit, the delay of the constant multiplication by a coefficient can be simply measured by the number of ADD/SUB steps on a maximal path in the part of the network corresponding to the coefficient. The structural operator(s) of the adder-tree are not modeled on the binary tree. In linear phase FIR filters, which are used in most cases, a coefficient is accumulated twice at symmetric tap positions, thus corresponding to two structural operators. In a general FIR filter, an adder-tree corresponds to one structural operator. For a structural operator, the adder-tree output edge always Inputs to it as the right side operand, while the left input and output edges are on the accumulation line and their bit widths can be predetermined according to the upper/lower bound of accumulated values up to this coefficient tap. The left input edge is always "+" signed and carries 0 shift. The cost model for the logic depth and structural operator are calculated.

2.3 Mixed Integer Programming

We generally use Logic depth to describe the required ADD/SUB steps. For a coefficient whose logic depth is less than the filters logic depth, incrementing (relaxing) its logic depth may reduce the resource consumption. Given an algorithm which computes the adder-tree of the minimum resource on a given depth for a coefficient, if is less than the filters logic depth, one can always try increasing by 1 and rescheduling onto a depth adder-tree for possible reduction of resource without degrading the filters clock performance. We describe the schedule of an adder-tree of an individual coefficient to minimize the hardware resource. As linearity is required in MIP, various techniques to transform modeling friendly non-linear expressions into linear equations and inequalities are indispensable.

III. IMPLEMENTATION OF FIR FILTER USING CSA TREE

The performance of the multiple constant multiplication scheme is evaluated by implementing an FIR filter using carry select adder tree scheduling. The ripple carry adder replaced with the carry select adder to improve the performance of the filter. Which is the simple and fastest adder compared to all adders. The carry-select adder generally consists of two ripple carry adders and a multiplexer. The ripple carry adder part is replaced with the carry select adder and the FIR Filter is implemented and the performance of the design is evaluated in terms of delay and power.

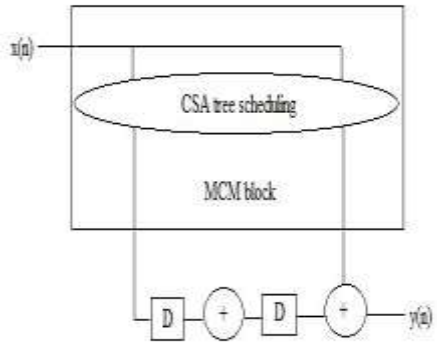

Fig. 3: Tranposed FIR filter

3.1 Single-Stage CSLA

The general expression to calculate the AOI gate counts of the n-bit proposed CSLA and the BEC-based CSLA of

[6] and CBL-based CSLA of [7] and [8] are given in Table 1 of single- stage design. We have calculated the AOI gate counts on the critical path of the proposed n-bit CSLA and CSLAs of [6]–[8] and used those AOI gate counts in (5b) to find an expression for delay of final-sum and output-carry in the unit of Ti (NOT- gate delay). The delay of the n-bit single-stage CSLA is shown in Table 1 for comparison.

Table 1

General Comparison o f gate counts and delay o f the proposed and existing CSLAS for single -stage design n: input bit-width

Design	AND-gate (N_a)	OR-gate (N_o)	NOT-gate (N_i)	final-sum (T_{fs})	output-carry (T_{cout})
Conventional	$14n-4$	$7n-3$	$5n-1$	$\max(t,3.5n+2)+4.5$	$\max(t,3.5n+1)+4.5$
CSLA [6]	$11n-2$	$5n-1$	$7n$	$\max(t,3.5n+8.3)+4.5$	$\max(t,3.5n+8.3)+4.5$
CSLA [7]	$7n$	$4n$	$5n$	$4.5n+1.8$	$4.5n+1.8$
CSLA [8]	$14n-7$	$8n-4$	$9n-4$	$9n+1$	$9n+1$
Proposed	$8n-2$	$5n-1$	$4n$	$\max(t,3.5n+2.7)+8$	$\max(t,3.5n+2.7)+3.5$

t stands for delay of *input-carry*. $t=0$ for single stage adder design. Delay expressed in the unit of T_i (NOT-gate delay).

VI. CONCLUSION

The adder tree scheduled in bit-level using CSA (carry select adder) are designed and are implemented in vhdl using Xilinx 13.2 ISE tool and the results are showed in terms of delay and power. The CSA proves to be the High Speed and Low Power multiple constant multiplication block. It is also implemented with FPGA. The MIP based algorithm used for bit level analysis. The performance of the MCM in terms of delay and power is evaluated by implementing an FIR Filter by using the CSA in the adder part and again it proves to be the High Speed and Low Power system. Thus the designed high speed FIR filter is used for speech signal processing applications.

REFERENCES

[1] Yu Pan and Pramod Kumar Meher, Senior Member, IEEE, “Bit-Level Optimization of Adder-Trees for Multiple Constant Multiplications for Efficient FIR Filter Implementation,” ieee transaction on circuits and systems-1: Regular papers, vol. 61, no. 2, Feb 2014.

[2] P. K. Meher and Y. Pan, “Mcm-based implementation of block fir filters for high-speed and low-power applications,” in Proc. VLSI and System-on-Chip (VLSI-SoC), 2011 IEEE/IFIP 19th Int. Conf., Oct. 2011, pp. 118–121.

[3] L. Aksoy, C. Lazzari, E. Costa, P. Flores, and J. Monteiro, “Design of digit-serial FIR filters: Algorithms, architectures, and a CAD tool,” IEEE Trans. Very Large Scale Integration (VLSI) Syst., vol. 21, no. 3, pp. 498–511, Mar. 2013.

[4] P.Prashanti, Dr. B.Rajendra Naik, “Design and Implementation of High Speed Carry Select Adder” International Journal of Engineering Trends and Technology (IJETT) – Volume 4 Issue 9- Sep 2013.

[5] Paulchamy balaiah and Dr.Ila Vennila, “A Proficient Design of Hybrid Synchronous and Asynchronous Digital FIR Filter using FPGA”, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 4, No 1, July 2012 , ISSN (Online): 1694-0814

[6] M. B. Gately, M. B. Yeary, and C. Y. Tang, “Multiple real-constant multiplication with improved cost model and greedy and optimal searches,” in Proc. IEEE ISCAS, May 2012, pp. 588–591.

[7] M.Kumm, P.Zipf, Faust, and C.H.Chang “Pipe line dadder graph for MCM” ieee , May2012.