ISSN 2394-3777 (Print)
ISSN 2394-3785 (Online)
Available online at www.ijartet.com
International Journal of Advanced Research Trends in Engineering and Technology (IJARTET)
Vol. 3, Special Issue 2, March 2016

Low Complexity Design for FIR Filter Based MCM Technique using Wallace Multiplier

Nandhini V
ECE Department,
Salem College of Engineering
and Technology, Salem, India
nandhini09ec@gmail.com

Dayalrai R
ECE Department,
Salem College of Engineering
and Technology, Salem, India
dayal@gmail.com

ABSTRACT

Finite Impulse Response Filters (FIR) are widely used in many digital signal processing applications. The main key requirement of FIR filters is low complexity. For low complexity FIR filter, two architectures was implemented, namely Multiple Constant Multiplication [MCM] technique and Wallace Multiplier Algorithm [WMA]. The MCM technique was introduced for implementing the reduced number of multipliers. The implementation of MCM technique was based on Wallace multiplier and half adder. Implementation of FIR filters using Multiple constant multiplications a 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. Introducing the Wallace tree multiplier was reduced both the number of adders and subtractions as well as the number of shifts. Multiple constant multiplication or accumulation in a direct FIR structure is implemented by means of an enhanced version of Wallace multipliers. In order to reduce the hardware which ultimately reduces an area and power, Energy Efficient full adders plays an important role in Wallace tree multiplier. The proposed Wallace multiplier will have fewer adders than Standard Wallace multiplier. In both multipliers, at the last stage half adders and full adders are used.

Index Terms – Design Blocks of FIR filter design, Finite Impulse Response,

I . INTRODUCTION

The increasing level of device integration and the growth in complexity of electronic circuits, reduction of power efficiency has come to fore as a primary design goal while power efficiency has always been desirable in electronic circuits. The finite impulse response digital filter is the fundamental element of digital signal processing systems. The implementation cost and power consumption are also high because of computational complexity. The complexity of the FIR filter is dictated by the complexity of the coefficient multipliers. The multipliers are the most expensive blocks in terms of area, delay and power in a FIR filter structure. As shifts are less expensive in terms of hardware implementation, the design problem can be defined as the minimization of the number of addition/subtraction operations to implement the coefficient multiplications. The complexity of the multiplier block in a FIR filter is reduced, if implemented as shift adders and sharing common sub expressions. In order to reduce the complexity of the filter, the filter coefficients are encoded using the pseudo random floating point method, however it is limited to filter lengths less than 40. The methods are only suitable for fixed logic filters where the coefficients are fixed. The architectures are appropriate only for relatively lower order filters and not suitable for channel filters in communication receivers. The idea is to precompute the values such as 0x, 1x, 2x, 3x, 4x, 5x, 6x and 7x, where x is the input signal and then reuse these precomputations efficiently using multiplexers.

In order to achieve both high performance and low complexity, by employing the scalable implementation scheme, propose an efficient design technique suitable for all types of sub band filter banks. In digital signal processing system sometimes it becomes necessary to convert the data to a new rate to make it easier to process or to achieve compatibility with another system. Therefore multi rate signal processing is used which is defined as the discrete time system that process data at more than one sampling rate to perform the desiredigital operations.

An efficient low power reconfigurable FIR filter architecture, where the filter order can be dynamically changed depending on the amplitude of both the filter coefficients and the inputs. In other words, when the data sample multiplied to the coefficient is so small as to mitigate the effect of partial sum in FIR filter, the multiplication operation can be simply cancelled. The filter performance degradation can be minimized by controlling the error bound as small as the quantization error or signal to noise power ratio of given system. The primary goal is to reduce the dynamic power of the FIR filter. The main contributions are summarized as follows. A new reconfigurable FIR filter architecture with real time input and coefficient monitoring circuits is presented. Since the basic filter structure is not changed, it is applicable to the FIR filter with programmable coefficients or adaptive filters. It provide mathematical analysis of the power saving and filter performance degradation.

As the scaling of silicon devices has progressed over the past four decades, semiconductor memory has become cheaper, faster and more efficient. According to the requirement of application environments, memory technology has been developed in a wide and diverse manner. To get the overall performance and to minimize the access delay and power dissipation, either the processor has been move to memory or the memory has been move to processor in order to place the computing logic and memory elements at closest proximity to each other. In addition to that, memory elements have also been used for a complete arithmetic circuit or a part of that in various applications. Memory based structures are well suited for many digital signal processing (DSP) applications. Memory elements like RAM or ROM are used as a part or whole of an arithmetic unit. Memory based structures are more regular compared with the multiply accumulate structure and have many other advantages like very greater potential for high throughput and reduced latency in implementation, (since the memory access time is much lesser than the usual multiplication time) and are expected to have less dynamic power consumption due to less switching activities for memory read operations when compared to the conventional multipliers.

II. DESIGN BLOCKS OF FIR FILTER DESIGN

As a modification 32 wide Multiple constant multiplication, Wallace tree multiplier algorithm can be estimated in Fig. 3.1 and also simple type of adder can be using for FIR filter design.

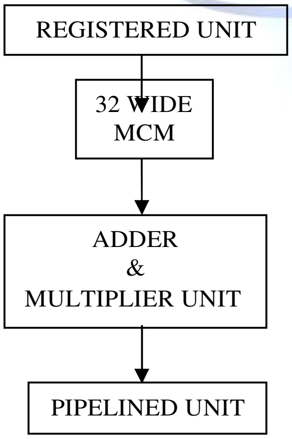


Fig.1 Design Blocks

The filter design have four main design blocks in it's architecture. The combination of all blocks are called as FIR filter design steps. Introducing the Wallace tree multiplier will reduce both the number of adders and subtractions as well as the number of shifts. Multiple constant multiplication/accumulation in a direct FIR structure is implemented by means of an enhanced version of Wallace multipliers.

III. FINITE IMPULSE RESPONSE

In signal processing, a finite impulse response filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. This is in contrast to infinite impulse response filters, which may have internal feedback and may continue to respond indefinitely (usually decaying). The impulse response of an Nth order discrete time FIR filter (i.e., with a Kronecker delta impulse input) lasts for N + 1 samples, and then settles to zero. FIR filters can be discrete time or continuous time and digital or analog.

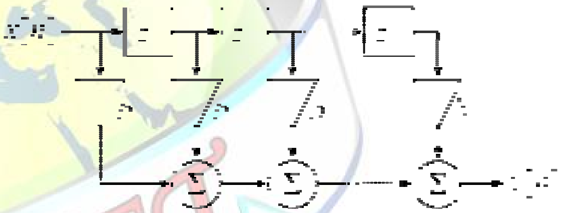


Fig.2 FIR filter

A discrete time FIR filter of order N. The top part is an N stage delay line with N + 1 taps. Each unit delay is a z^{-1} operator in Z transform notation. The output y of a linear time invariant system is determined by convolving its input signal x with its impulse response b. For a discrete time FIR filter, the output is a weighted sum of the current and a finite number of previous values of the input. The operation of the Fig. 3.2 is described by the following equation , which defines the output sequence y[n] in terms of its input sequence x[n].

$$Y[n] = b_0x[n] + b_1x[n\ 1] + . . . + b_N\,x[n\ N]$$
$$= \sum b_i\,x[n\ i]$$

Where,

x[n] is the input signal,
y[n] is the output signal,
b_i are the filter coefficients, also known as tap weights, that make up the impulse response,
b_i N is the filter order.

The conventional finite impulse response filters use multipliers, adders and delay elements to produce the required output. The multipliers which multiplies input with the fixed content significantly occupies more place to store their temporary values and also increases the power consumption.

The multipliers are replaced with memory based structures to reduce area and also to reduce the system latency. Several architectures have been reported for memory based implementation of discrete sinusoidal transforms and digital filters for digital signal processing applications. Many efforts for reducing power consumption of FIR filter generally focus on the optimization of the filter coefficients while maintaining a fixed filter order. FIR filter structures are simplified to add and shift operations and minimizing the number of additions/subtractions is one of the main goals of the research. However, one of the drawbacks encountered in those approaches is that once the filter architecture is decided, the coefficients cannot be changed therefore, those techniques are not applicable to the FIR filter with programmable coefficients. Approximate signal processing techniques are also used for the design of low power digital filters. In filter order dynamically varies according to the stop band energy of the input signal. However, the approach suffers from slow filter order adaptation time due to energy computations in the feedback mechanism.

FIR filtering operation performs the weighted summations of input sequences are called as convolution sum, which are frequently used to implement the frequency selective low pass, high pass, or bandpass filters. The amount of computation and the corresponding power consumption of FIR filter are directly proportional to the filter order, if it can dynamically change the filter order by turning off some of multipliers, significant power savings can be achieved. However, performance degradation should be carefully considered when change the filter order. The coefficients of a typical 25tap low pass FIR filter. The central coefficient has the largest value the coefficient Ck has the largest value in the 25tap FIR filter and the amplitude of the coefficients generally decreases ask becomes more distant from the centre tap. The data inputs x(n) of the filter, which are multiplied with the coefficients, also have large variations in amplitude. Therefore, the basic idea is that if the amplitudes of both the data input and filter coefficient are small, the multiplication of those two numbers is proportionately small thus, turning off the multiplier has negligible effect on the filter performance.

IV. IMPLEMENTATION TECHNIQUES

The implementation architecture of the four tap Finite Impulse response filter with the Multiple Constant Multiplication block. The architecture consists of following elements,

- MCM block
- Full Adder
- 8 bit Register

The Multiple Constant Multiplication block is implemented using the Digit based algorithm, CSA algorithm and Wallace multiplier algorithms. The digit serial adders are required to add the constants to produce the final filter output. The 8 bit register is used to store the output results of each adder, since the next adder will requires the previous output as one of the input. In the implementation architecture, x(n) represents the input data, y(n) represents the output data of the filter and D is not a flip flop which represents the storage element. Low power design of digital integrated circuits has emerged as a very active and rapidly developing field of CMOS design. In the past the major concerns of the VLSI designer were speed, the area and cost, power consideration was typically a secondary importance.

Applications of Digital Signal Processing involve a large number of multiplication using constants which lead to complex design and consumption of more area and power. FIR filters can be implemented with multiplier blocks through the use of multiple constant multiplication. MCM is an efficient way of implementing several constant multiplication with the same input data. The coefficients are expressed using shifts, adders and multipliers. The generation of multiplier block of set of constant is MCM block. A finite impulse response filter is usually implemented by using a series of delays, multipliers and adders to create the filters output. FIR filters can be easily designed to be “linear phase”. Linear phase has a constant group delay, where all the frequency components have equal delay time. So the filter does not cause any phase distortion or delay distortion. The FIR filter structure takes one input sample in each clock cycle, and produces one filter output in each cycle. The first filter output is obtained after a latency of three cycles

V. BASIC CONCEPTS OF MCM

Multiple Constant Multiplication (MCM) is an arithmetic operation that multiplies a set of fixed-point constants with the same fixed-point variable X. From a circuit point of view, MCM dominates the complexity of the whole category of Linear Time Invariant (LTI) systems, such as, FIR/IIR filters, DSP transforms (DCT, DFT, Walsh.,) LTI controllers, crypto-systems, etc. To be efficiently implemented, MCM must avoid costly multipliers. The hardware alternative must be multiplierless i.e., using only additions, subtractions, and shifts. Therefore, the MCM problem is defined as the process of finding the minimum number of addition/subtraction operations. The computational complexity of MCM is conjectured to be NP-hard. Because of the increasing demand in high-speed and low-power design, MCM problem has been the focus of important researches during these last three decades. As a result, a big number of MCM algorithms have been used, mostly based on the acyclic directed graphs, or common subexpression elimination or the combination of both together.

An extension of SCM is the problem of multiplying a variable x by several constants t1, . . . , tn in parallel in a so-called multiplier block. Since intermediate results of the constant decompositions may be shared, a multiple constant multiplier block may be decomposed into fewer operations than the sum of the single constant decompositions operation counts. The problem of finding the decomposition with the fewest operations is known as Multiple Constant Multiplication.

The potential savings from sharing intermediate results increase with the number of constants as illustrated. The plot compares the number of add/subtract operations (y-axis) for varying sizes n (x-axis) of sets of 12-bit constants using separate optimal SCM decompositions and using RAG-n, the heuristic MCM algorithm.

The MCM problem is particularly relevant for the multiplierless implementation of digital finite impulse response filters ,but also for matrix-vector products with a fixed matrix. The linear signal transforms, such as the discrete Fourier transform or the discrete cosine transform. In an n-tap FIR filter, every input sample is multiplied by all n taps. Discrete Fourier and trigonometric transform algorithms, on the other hand, involve 2x2 rotations, which require the simultaneous multiplication by two constants. A multiplier block, which implements the parallel multiplication by 23 and 81 using only 3 add/subtract operations, although the separate optimal decompositions of 23 and 81 each require 2 add/subtract operations. The different problem of multiplexed multiple constant multiplication was considered. In the multiplier block contains multiplexers that are switched by control logic to achieve multiplication by different constants. This way sequential multipliers can be fused.

Other optimization metrics are reducing the number of add/subtract operations, it is often desirable to optimize for other metrics, for example, the critical path of the MCM block, or the register pressure in the generated code. It does not consider this type of optimization however, the structure of our algorithm enables its adaptation to other target metrics. Avoiding costly multipliers is particularly important in hardware implementations, for example, of digital signal processing functionality such as filters or transforms. However, replacing constant multiplications with additions and shifts can also be relevant in software implementations. For example, as optimization for speed, since integer multipliers often have a significantly lower throughput than adders, but also for embedded processors, which may not feature a multiplication unit at all. The MCM problem can be considered as a fundamental problem in computer arithmetic.

The main contribution of is a new MCM algorithm that achieves significantly better results than previous methods are demonstrated for the cases most relevant in practice bit width $b \leq 32$ and $n \leq 100$ constants. However, asymptotically the new algorithm produces solutions with no known better complexity than O(nb) add/subtract operations, just like CSD and all the other algorithms. The A-distance computation and estimation framework development was very useful for further research in this area. One direction could be to improve the heuristic, which currently combines A-distances in a trivial way.

VI. CLASSES AND MERITS OF MCM

CLASSES

1. Digit-based recoding
2. Common subexpression elimination (CSE) algorithms
3. Graph-based algorithms
4. Hybrid algorithms.

MERITS

Fully Predictable

The upper-bound (Upb) in number of additions, adder-depth (Ath), and average (Avg), are known with exact analytic formulas. This feature not only allows designers to get a pre implementation picture on area, speed and power, but it also enables synthesis tools to rapidly satisfy user constraints and perform necessary tradeoffs without the "endless" feedbacks looking for the appropriate solution. Note that none of the existing MCM algorithms is predictable in Upb, Ath or Avg.

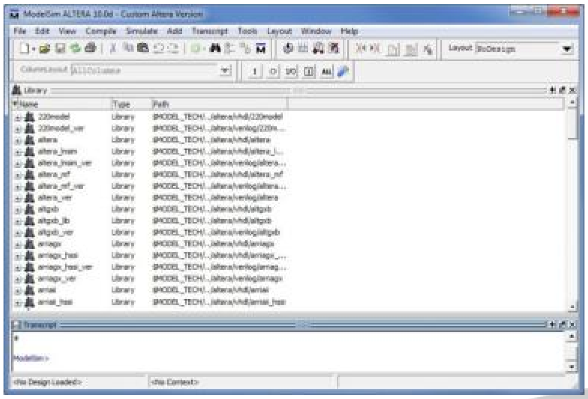
Sublinear

For a given number M of nonnegative constants with a bit-length N, RADIX-2r exhibits a sublinear runtime-complexity O(M×N/r), where r is a function of (M,N). This means that it has no limitation regarding the couple (M,N). For high-complexity problems (M×N>>), RADIX-2r is most likely the only one that is even feasible to run. Note that the best MCM heuristics have either a polynomial or exponential complexity.

High-Speed And Low-Power Solution

Adder-depth (Ath) is not only a measure of the critical-path (speed), but also a good indicator of the power consumption. It has been proved in many researches that a lower Ath results in lower power consumption due to the limitation of the glitch propagation. The existing MCM algorithms can not compete with RADIX-2r because it allows a logarithmic reduction of Ath, while it is not possible in the other algorithms due to the shared additions.

VII. SYNTHESIS RESULTS


ModelSim ALTERA 10.0d - Custom Altera Version

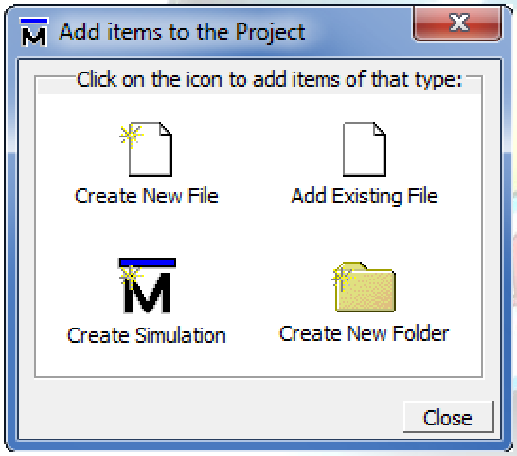
File Edit View Compile Simulate Add Transcript Tools Layout Window Help
Library
Transcript
ModelSim>

Name	Type	Path
220model	Library	$MODEL_TECH/../altera/vhdl/220model
220model_ver	Library	$MODEL_TECH/../altera/verilog/220m...
altera	Library	$MODEL_TECH/../altera/vhdl/altera
altera_lnsim	Library	$MODEL_TECH/../altera/vhdl/altera_l...
altera_lnsim_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_mf	Library	$MODEL_TECH/../altera/vhdl/altera_mf
altera_mf_ver	Library	$MODEL_TECH/../altera/verilog/altera...
altera_ver	Library	$MODEL_TECH/../altera/verilog/altera
altgxb	Library	$MODEL_TECH/../altera/vhdl/altgxb
altgxb_lib	Library	$MODEL_TECH/../altera/vhdl/altgxb
altgxb_ver	Library	$MODEL_TECH/../altera/verilog/altgxb
arriagx	Library	$MODEL_TECH/../altera/vhdl/arriagx
arriagx_hssi	Library	$MODEL_TECH/../altera/vhdl/arriagx_...
arriagx_hssi_ver	Library	$MODEL_TECH/../altera/verilog/arriag...
arriagx_ver	Library	$MODEL_TECH/../altera/verilog/arriagx
arriaii	Library	$MODEL_TECH/../altera/vhdl/arriaii
arriaii_hssi	Library	$MODEL_TECH/../altera/vhdl/arriaii_hssi

<No Design Loaded> <No Context>

Fig.3 Modelsim Window

The fig.3 shows the initial page opened in the modalism software. While opening the software we can get this page as output, from this page only the next step may process. It has more options like file, edit, view, add and help.


Add items to the Project

Click on the icon to add items of that type:

Fig.4 Add file to project

The fig.4 is the storage space of the project. In that the required project location will be added to the list of files. We can add files and create folders also. By adding the project into the file it will easy to reuse.

The Project Manager greatly reduces the time it takes to organize files and libraries. As you compile and simulate, the Project Manager stores the unique settings of each individual project, allowing you to restart the simulator right where you left off. Simulation properties allow you to easily re-simulate with pre-configured parameters. The fig.5 is the simulation window of the modelsim. In this simulation page can give our input and get the correspondence output for the input.

The given input will be changed and again find the output. By using this simulation window we can easily compare the output difference between the different inputs.

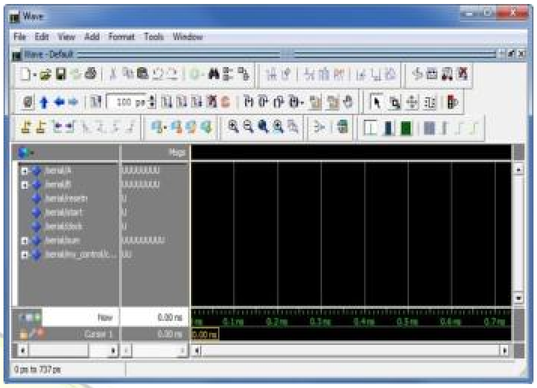


Fig.5 A Simulation Window

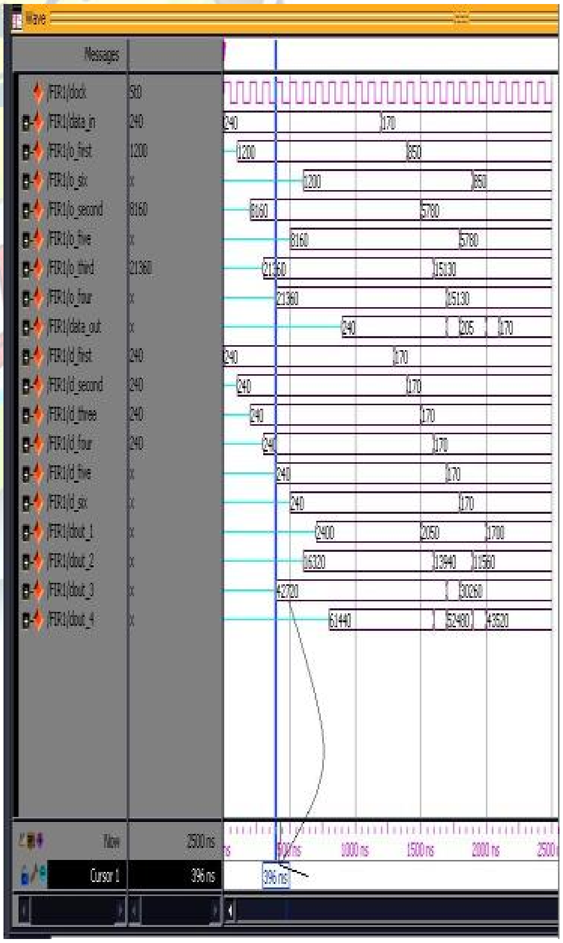


Fig.6 Output of FIR

By using the obtained ns in the simulation output fig.6 compare the results in the existing system methods and will get the output by using small calculations.

CONCLUSION

In this paper, we have explored the possibility of realization of block FIR filters in transpose form configuration for area delay efficient realization of both fixed and reconfigurable applications. A generalized block formulation is presented for transpose form block FIR filter, and based on that we have derived transpose form block filter for reconfigurable applications. We have presented a scheme to identify the MCM blocks for horizontal and vertical subexpression elimination in the proposed block FIR filter for fixed coefficients to reduce the computational complexity. Performance comparison shows that the proposed structure involves significantly less ADP and less EPS than the existing block direct-form structure for medium or large filter lengths while for the short-length filters, the existing block direct-form structure has less ADP and less EPS than the proposed structure.

REFERENCE

[1] Behrooz Parhami and Ding Ming Kwai "Parallel architectures and adaptation algorithms for programmable fir digital filters with fully pipelined data and control flows, 2002.

[2] M. Backia Lakshmi and D. Sellathambi "Reconfigurable FPGA implementation of FIR filter using modified DA method", 2011.

[3] A.Sirisha, P.Balanagu and N.Suresh Babu "Design of FIR filter using look up table and memory based approach", 2012.

[4] Ramesh and Nathiya "Realization of FIR filter using modified distributed arithmetic architecture", 2012

[5] K.Babulu And Mohammad Shaffi "FPGA implementation of multi-rate reconfigurable architecture with low complexity FIR filters", 2012.

[6] N.Durairajaa, J.Joyprincy and M.Palanisamy "Design of low power and area efficient architecture for reconfigurable FIR filter", 2013.

[7] N. Sriram and J. Selvakumam "A Reconfigurable FIR filter architecture to trade off filter performance for dynamic power consumption ", 2013.

[8] J.Britto Pari and P.Joy Vasanth Rani' "Reconfigurable architecture of RNS based high speed FIR filter", 2014.

[9] K.Ananthan and N.S.Yogaananth "VLSI implementation of reconfigurable low power fir filter architecture", 2014.

[10] Basant Kumar Mohanty and Pramod Kumar Meher' "A High-performance FIR filter architecture for fixed and reconfigurable applications", 2015.