IOSR Journal of Engineering (IOSRJEN) **www.iosrjen.org**
ISSN (e): 2250-3021, ISSN (p): 2278-8719
PP 34-36

Prefix Topology Basedhierarchical Bitwidthoptimization For Highperformance Fir Filter

[1]Ms. DELPHINE CHANDRA MARY D. M.E VLSI design, [2]Ms. S. Lakshmi, M.E., (Ph.D.)

[1]Department of Electronics and Communication Engineering. Thirumalai Engineering College,
[2]Department of Electronics and Communication Engineering. Thirumalai Engineering College,

Abstract : *In we present prefixes topology based accumulation units w ith variable latency to link equations via parallel-prefix computation using various methodologies such as such as ripple carry adder, Kogge Stone adder, Brent Kung adder, Ladner Fischer adder and Han Carlson adder. This work is also permitted to design high-speed and unique MAC hardware structure using vedic multiplier, thereby making them suitable for any DSP applications. To prove the hardware efficiency of the Wallace tree multiplier unit it is compared with state-of-the-art methods like high speed vedic and shift and add based DA method. Moreover, this methodology has several attractive features such as simplicity, regularity and modularity of architecture. Also, the DA technique can be designed to meet low complexity demand requirements of FIR filter design, where the bit optimization is processed within the bounded delay. To reduce the complexity of filter, coefficients are represented in canonical signed digit representation as it is more efficient than traditional binary representation. The comparative analyzes is carried out using 20 order FIR filter. Hardware optimization in terms of area, delay and power of different prefix techniques, Vedic multiplier, add and shift method and Wallace tree (WT) multiplier are is analyzed using FPGA hardware synthesis. CUT in a parallel. In this scan design testing; all selected storage elements are replaced by scan cells.*

I. Introduction

Digital filtering has specific characteristics that you need to pay special attention to. The analog input signal must satisfy certain requirements. Furthermore, on converting an output digital signal into analog form, it is necessary to perform additional signal processing in order to obtain the appropriate result. The process of converting an analog signal into digital form is performed by sampling with a finite sampling frequency fs. If an input signal contains frequency components higher than half the sampling frequency (fs/2), it will cause distortion to the original spectrum. This is the reason why it is first necessary to perform filtering of an input signal using a low-pass filter that eliminates highfrequency components from input frequency spectrum. This filter is called antialiasing filter as it prevents aliasing. After the process of filtering and sampling, a digital signal is ready for further processing which, in this case, is filtering using the appropriate digital filter. The output signal is also a digital signal which, in some cases, needs to be converted back into analog form. After digital-to-analog conversion, signal contains some frequency components higher than fs/2 that must be eliminated. Again, it is necessary to use a low-pass filter with the sampling frequency fs/2. The specific characteristics of conversion affecting the signal are beyond the scope of this book.Digital filter attenuation is usually expressed in terms of the logarithmic decibel scale (dB).

The attenuation measured in decibels can be found using the following expression:

$$a = 20 * \log(H(f))$$

Cut-off frequencies are used for filter specification, which will be discussed later. The cut-off frequency of the passband is a frequency at which the transition of the passband to the transition region occurs. The cut-off frequency of the stopband is a frequency at which the transition of the transition region to the stopband occurs. These two frequencies are equivalent only for the ideal filter which is not possible to realize in practice. In other words, they are always different.

II. Existing System

The addition of two binary numbers can be formulated as a prefix problem. The corresponding parallel-prefix algorithms can be used for speeding up binary addition and for illustrating and understanding various addition principles. This section introduces a mathematical and visual formalism for prefix problems and algorithms.

Prefix Problems. In a prefix problem, n outputs (yn-1, yn-2, …., y0) are computed from n inputs (xn-1, xn-2, ….,x0) using an arbitrary associative operator as follows

$$y_0 = x_0$$
$$y_1 = x_1 \bullet x_0$$
$$y_2 = x_2 \bullet x_1 \bullet x_0$$
$$\vdots$$
$$y_{n-1} = x_{n-1} \bullet x_{n-2} \cdots \bullet x_1 \bullet x_0$$

The problem can also be formulated recursively:

$$y_0 = x_0 \qquad i = 1,2,\ldots,n-1$$
$$y_i = x_i \bullet y_{i-1};$$

In other words, in a prefix problem, every output depends on all inputs of equal or lower magnitude, and every input influences all outputs of equal or higher magnitude. Due to the associativity of the prefix operator •, the individual operations can be carried out in any order. In particular, sequences of operations can be grouped in order to solve the prefix problem partially and in parallel groups of input bits (xi, xi-1, ….,xk), resulting in the group variables Yi:k. At higher levels, sequences of group variables can again be evaluated, yielding m levels of intermediate group variable Yl i:k. It denotes the prefix result of bits (xi, xi-1, ….,xk) at level l. The group variables of the last level m must cover all bits from I to 0 (Ym i:k) and therefore represent the results of the prefix problem

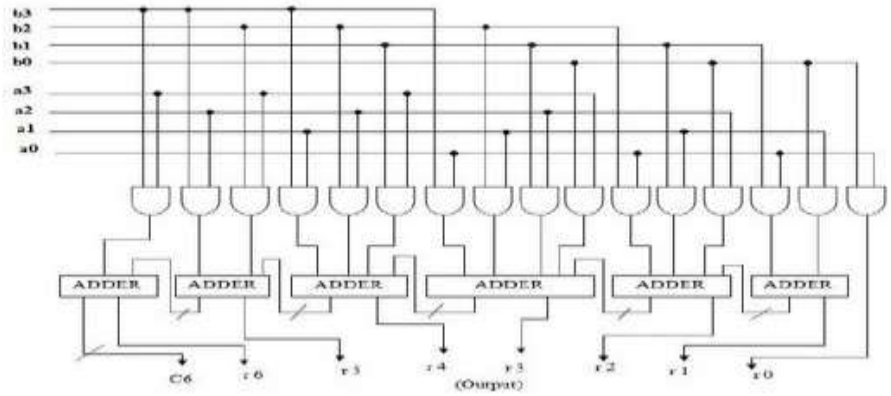
III. Proposed System

Finite Impulse Response (FIR) filters are one of the key building blocks of many signal processing applications in communication systems. Channel equalization, interference cancellation and matched filtering are some variety of FIR filter applications. Recently, software defined radio (SDR) application has increased the demand for reconfigurable communication systems capable of multi standard operations. Hence, the programmable and reconfigurable FIR filter architectures are needed for next generation communication systems with low power consumption, low complexity and high speed operation requirements. The major bottleneck in FIR filter implementation is coefficients multipliers, which are traditionally implemented by add/sub/shift operations. In non-reconfigurable filters, these coefficients are constant and shift operation is done by hardwiring. The long tree of adders in multiplier implementation increases switching activity and physical capacitance and then power consumption. Some techniques have been proposed to minimize the number of required adders for multiplier implementation. Canonical signed digit (CSD) coefficient representation and Common Sub expression Elimination (CSE) methods are well known approaches which produce FIR filter coefficient multipliers with low complexity. In fixed coefficients FIR filter application, the CSE is an efficient approach to share the common sub expressions. This is accomplished by searching the common sub expressions in coefficients terms before filter implementation and sharing them to eliminate extra computational complexity.

In programmable and reconfigurable filter, the coefficients are not fixed and it is not easy to find the common sub expressions for newly applied coefficients. The conventional FIR filter implementation approaches (CSD, CSE…) cannot be usable and the dedicated multipliers are required for each coefficient multiplication. Several methods have been proposed in [10-11] to implement the reconfigurable FIR filters. In a fully

programmable multiply accumulates (MAC) based filter processor has been proposed which is suffering from large delay of long data path, large area and power consumption requirements. An interesting CSD based reconfigurable FIR filter architectures have been proposed in and where its area and power consumption are also large. Two other efficient reconfigurable FIR filter implementation approaches. These works have been focused on the implementing of FIR filter by partitioning the filter coefficient into fixed groups (sub-coefficients) and pre-computing the products of input data with these constant fixed sub coefficients. These partial products are distributed inside the chip instead of input data and properly selected by filter tap multiplexers to compose the desired coefficient multiplication. This is like as the splitting coefficient multiplier to two global and local units reported in, where split CSD (SCSD) representation was involved to reduce adder counts in constant coefficient FIR filter implementation. Signed sub coefficient method is proposed in this paper to compute the partial products. In the next session, the review of partial product method is described.

IV. Architecture Diagram



Hardware architecture of the Urdhva tiryakbhyam multiplier

V. Conclusion

Adder component is a crucial component in automatic ECG processing. Generally speaking, successful front-end features should carry enough discriminative information for enhancements, and it should fit well with the backend modeling, and be robust fetal ECG extraction with respect to the changes of acoustic environments. As a part of the project work the FIR feature extraction were analyzed for better understanding of the work. The performances of proposed system were simulated in detail and their performances were evaluated under different noise conditions.

Future Enhancement

We can introduce the concept of fault enable by pass by using bi directional fault detection circuit as a pre-processer to preserve details of the faults occurs in interior stages of the design.

References

[1]. Avizienis, ―Signed Digit Number Representation for Fast Parallel Arithmetic,‖ IRE Trans. Electron. Computers, vol. EC-10, pp.389 400, 1961.
[2]. Byonghyo Shim, *Member, IEEE, and Naresh R. Shanbhag, Fellow, IEEE*―Energy-
[3]. Efficient Soft Error-Tolerant Digital Signal Processing,‖ *IEEE Trans. Comput., vol. 39, no. 10,* pp. 1304–1308, Oct.2006.
[4]. K. H. Chen and T. D. Chiueh, ―A low-power digit-based reconfigurable FIR filter,‖ IEEE Trans. Circuits Syst. II, vol. 53, no. 8, pp. 617–621, Aug. 2006.
[5]. K. Dabbagh-Sadeghipour, A. Aghagolzadeh, "A New Hardware Efficient, Low Power Fir Digital Filter Implementation Approach", Proc. IEEE ICECS.