

# DESIGN OF KARATSUBA ALGORITHM IN FIR FILTER

S.Abinaya<sup>1</sup>, R.Jenifa<sup>2</sup>

<sup>1</sup>*Electronics and communication engineering dept. , Prince shri venkateshwara padmavathy engineering college, Tamilnadu, India.*

<sup>2</sup>*Electronics and communication engineering dept. , Prince shri venkateshwara padmavathy engineering college, Tamilnadu, India.*

## ABSTRACT

*FIR Filter has widespread applications in signal processing such as image processing, biomedical signal processing, high speed communication systems, noise elimination and many more. The speed of FIR filter can be improved with high speed multipliers and with low propagation delay adders. In this paper we will discuss in detail about different reconfigurable FIR filter architectures and propose a karatsuba multiplier. Although all those three FIR filter architectures are reconfigurable but each of them have different complexity based on the way processing elements are implemented. Each of them has its own merits and demerits compared with the other methods. Xilinx 12.1 result shows better time and area reduction for our proposed fir filter implementation*

**Keyword:** *FIR filter, Karatsuba*

## 1. INTRODUCTION

FIR Filter has across the board applications in signal processing, for example, picture handling, biomedical signal processing, rapid correspondence frameworks, noise elimination and many more. The speed of FIR filter can be enhanced with fast multipliers and with low engendering postpone adders. In this paper we will talk about in insight about various reconfigurable FIR filter models. Albeit every one of those three FIR channel structures are reconfigurable yet each of them have distinctive unpredictability in view of the way preparing components are actualized. Each of them has its own benefits and faults contrasted and alternate strategies.

Filtering is a class of signal processing, the defining feature of filters being the complete or partial suppression of some aspect of the signal. Most often, this means removing some frequencies and not others in order to suppress interfering signals and reduce background noise. Filters can be classified in several different groups, depending on what criteria are used for classification. The two major types of digital filters are finite impulse response digital filters (FIR filters) and infinite impulse response digital filters (IIR).

Finite Impulse Response (FIR) filter are of great importance in digital signal processing (DSP) systems since their characteristics in linear phase and feed forward implementation make them very useful for building stable high performance filters. The direct and transposed form FIR filters implementations are mostly use in the context of digital filters. Although both architectures have similar complexity in hardware, the transposed form is generally preferred because of its higher performance and power efficiency. The multiplier block of the digital FIR filter in input stage is very compact structure because of the complexity and performance of the design

Techniques for reducing power consumption have become important due to the growing demand for portable multimedia devices. Since digital signal processing is pervasive in such applications, it is useful to consider how algorithmic approaches may be exploited in constructing lowpower solutions. A significant number of DSP functions involve frequency selective digital filtering in which the goal is to reject one or more frequency bands

while keeping the remaining portions of the input spectrum largely unaltered. Examples include lowpass filtering for signal upsampling and downsampling, bandpass filtering for subband coding, and lowpass filtering for frequency-division multiplexing and demultiplexing. The exploration of low-power solutions in these areas is therefore of significant interest. Reconfigurable hardware architectures are emerging as a suitable approach to combine high performance with flexibility and programmability. Recently, with the advent of software defined radio (SDR) technology, finite impulse response (FIR) filter research has been focused on reconfigurable realizations. The fundamental idea of an SDR is to replace most of the analog signal processing in the transceivers with digital signal processing in order to provide the advantage of flexibility through reconfiguration. This will enable different air-interfaces to be implemented on a single generic hardware platform to support multi-standard wireless communications. Wideband receivers in SDR must be realized to meet the stringent specifications of low power consumption and high speed

power consumption and high speed

## 2. FIR FILTER THEORY

The most common digital filter is the linear time-invariant (LTI) filter. An LTI interacts with its input signal through a process called linear convolution, denoted by  $y = f * x$  where  $f$  is the filter's impulse response,  $x$  is the input signal, and  $y$  is the convolved output. The linear convolution process is formally defined by:

$$y[n] = x[n] * f[n] = \sum_{k=0}^{\infty} x[n]f[n-k] = \sum_{k=0}^{\infty} f[k]x[n-k].$$

LTI digital filters are generally classified as being finite impulse response (i.e., FIR), or infinite impulse response (i.e., IIR). As the name implies, an FIR filter consists of a finite number of sample values, reducing the above convolution sum to a finite sum per output sample instant. An FIR with constant coefficients is an LTI digital filter. The output of an FIR of order or length  $L$ , to an input timeseries  $x[n]$ , is given by a finite version of the convolution sum given in (1), namely:

$$y[n] = x[n] * f[n] = \sum_{k=0}^{L-1} f[k]x[n-k],$$

where  $f[0] \neq 0$  through  $f[L-1] \neq 0$  are the filter's  $L$  coefficients. They also correspond to the FIR's impulse response. For LTI systems it is sometimes more convenient to express in the  $z$ -domain with

where  $F(z)$  is the FIR's transfer function defined in the  $z$ -domain by

$$F(z) = \sum_{k=0}^{L-1} f[k]z^{-k}$$

The  $L$ th-order LTI FIR filter is graphically interpreted in Fig.1. It can be seen to consist of a collection of a "tapped delay line," adders, and multipliers. One of the operands presented to each multiplier is an FIR coefficient, often referred to as a "tap weight" for obvious reasons. Historically, the FIR filter is also known by the name "transversal filter," suggesting its "tapped delay line" structure[7]

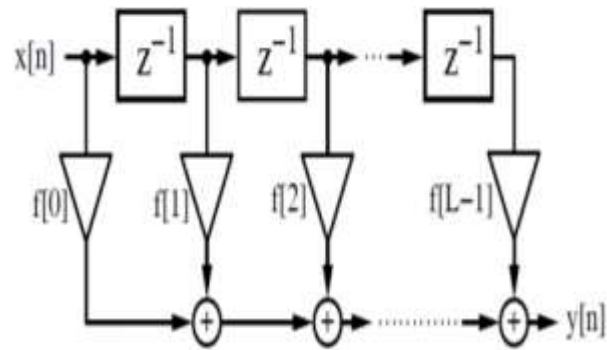


Fig -1 Structure of FIR filter

### 3. RELATED WORK

In [1] author proposes, presented low power and high-speed realisation of differential-coefficients-based finite impulse response filters. The conventional differential coefficients method (DCM) uses the difference between adjacent coefficients whereas we identify the coefficients that have the least difference between their magnitude values and use these minimal difference values to encode the differential coefficients. Our minimal-difference differential coefficients can be coded using fewer bits, which in turn reduces the number of full additions required for coefficient multiplication. By employing a differential-coefficient partitioning algorithm and a pseudofloating-point representation

In [2] author presents a low power programmable FIR filter based on partitioned multipliers. Architecture chosen for implementation is conventional direct form. Power efficient techniques like unsigned multiplication and reduction of switching activity are used. Paper presents power, area and speed analysis of the proposed design. FIR Filter is fully parameterized, dynamically programmable and technology independent

In [3] author proposed method is based on the masking method and the techniques for rounding and sharpening. The coefficient values of the model and masking filters are represented as integers using the rounding technique. The sharpening technique is applied to improve the overall magnitude characteristic and to satisfy the given specification Modified Booth Algorithm

### 4. PROPOSED SYSTEM

#### Karatsuba multiplier

The Karatsuba formula is used to speed-up the multiplication of large numbers by splitting the operands in two parts of equal length.

The standard procedure for multiplication of two  $n$ -digit numbers requires a number of elementary operations proportional to  $n^2$ , or  $O(n^2)$  in big-O notation. Andrey Kolmogorov conjectured that the classical algorithm was asymptotically optimal, meaning that any algorithm for that task would require  $n^2$  elementary operations.

In 1960, Kolmogorov organized a seminar on mathematical problems in cybernetics at the Moscow State University, where he stated the  $\Omega(n^2)$  conjecture and other problems in the complexity of computation. Within a week, Karatsuba, then a 23-year-old student, found an algorithm (later it was called "divide and conquer") that multiplies two  $n$ -digit numbers in  $O(\log^2 n)$  elementary steps, thus disproving the conjecture. Kolmogorov was very agitated about the discovery; he communicated it at the next meeting of the seminar, which was then terminated. Kolmogorov did some lectures on the Karatsuba result at the conferences all over the world and published the method in 1962, in the Proceedings of the USSR Academy of Sciences. The article had been written by Kolmogorov and contained two results on multiplication, Karatsuba's algorithm and a separate result by Yuri Ofman it listed "A.

Karatsuba and Yu. Ofman" as the authors. Karatsuba only became aware of the paper when he received the reprints from the publisher.

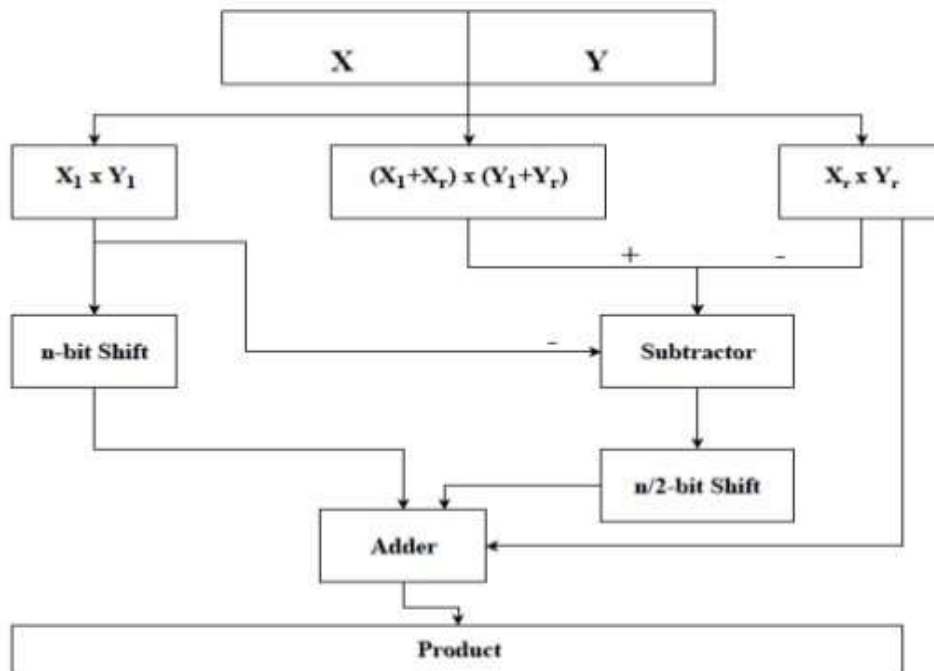
The basic step of Karatsuba's algorithm is a formula that allows one to compute the product of two large numbers  $x$  and  $y$  using three multiplications of smaller numbers, each with about half as many digits as  $x$  or  $y$ , plus some additions and digit shifts. This basic step is, in fact, a generalization of Gauss's complex multiplication algorithm, where the imaginary unit  $i$  is replaced by a power of the base

**Recursive application**

If  $n$  is four or more, the three multiplications in Karatsuba's basic step involve operands with fewer than  $n$  digits. Therefore, those products can be computed by recursive calls of the Karatsuba algorithm. The recursion can be applied until the numbers are so small that they can (or must) be computed directly.

In a computer with a full 32-bit by 32-bit multiplier, for example, one could choose  $B = 231 = 2147483648$ , and store each digit as a separate 32-bit binary word. Then the sums  $x_1 + x_0$  and  $y_1 + y_0$  will not need an extra binary word for storing the carry-over digit (as in carry-save adder), and the Karatsuba recursion can be applied until the numbers to multiply are only one-digit long.

**Block diagram of Karatsuba multiplication**



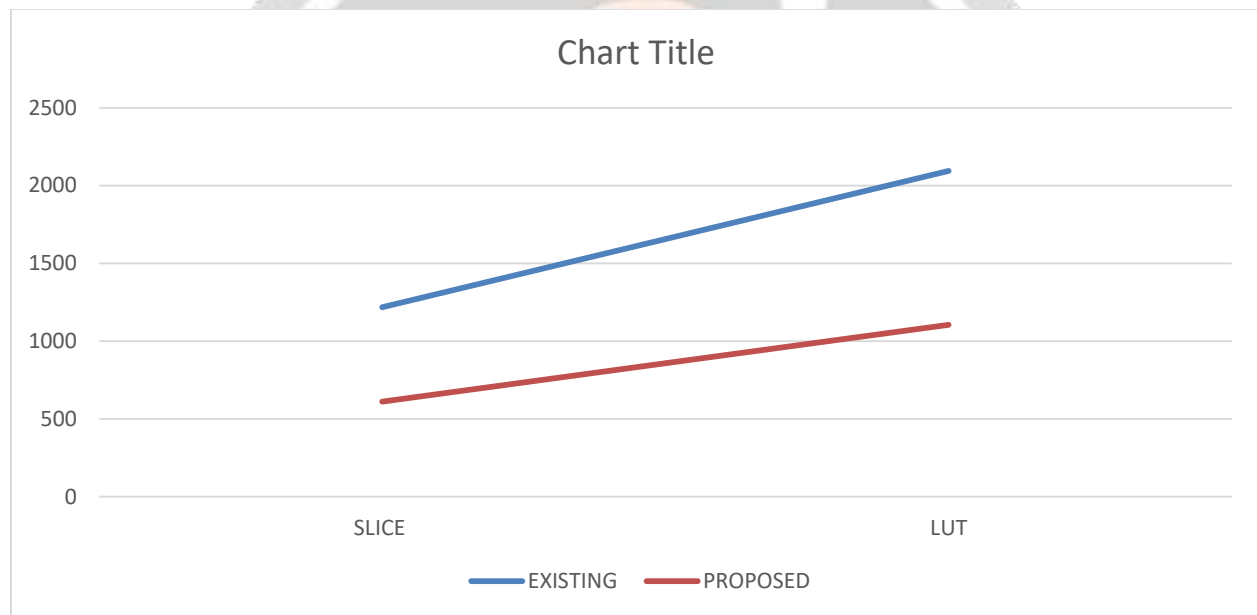
**Fig - 2** Block diagram of Karatsuba multiplier

### 5. PERFORMANCE ANALYSES

The table 1 given below is shown that there is a considerable reduction in time and area based on the implementation results which have been done by using Spartan-3 processor. The proposed algorithm significantly reduces area consumption when compared to the existing system.

**Table -1** Comparison of areas

METHOD	SLICE	LUT
1	1219	612
2	2095	1106



**Fig-3** Analysis

The above graph shows the comparison of area between the existing system and the proposed system. It shows that the area used for proposed system is very less on comparing with the area of the existing system.

### 6. CONCLUSION

In the Karatsuba multiplier, the authors analyzed a new technique for FIR filter implementation based on the use of the Karatsuba formula. The analysis shows that the Karatsuba filter architecture is composed by three sub-filters of reduced dynamic range, three adders and two hardwired shifters for the implementation of the input and output conversions. Evaluations of area, power and speed have been obtained by implementing a set of experiments showing the advantages of the Karatsuba implementation over the traditional transposed implementation. In

particular the earnings are relevant both in power and delay especially for high dynamic ranges. In the future work plan to address the Karatsuba algorithm to fault-tolerant applications that require FIR filtering

## 7. REFERENCES

- [1] Alen V Oppenheim, Ronald W Schafer and John R Buck, "Discrete Time Signal Processing", Prentice Hall Publications, New Jersey, 1999, pp. 340-438.
- [2] Asma Hemdani, M.Wissem Naouar, Ilhem Slama Belkhdja , and Eric Monmasson, "Design of a Digital IIR Filter for Active Filtering Applications", IEEE international Mediterranean Electrotechnical Conference, pp. 1107-1112, Mar. 2012.
- [3] Chong Xu, Ching-Yi Wang, and Parhi K.K, "Order-Configurable Programmable Power-Efficient FIR Filters", IEEE International Conference on High Performance Computing, pp. 357-361, Dec. 1996.
- [4] Fei Xiao, "Fast Design of IIR Digital Filters With a General Chebyshev Characteristic", IEEE Transactions on Circuits and Systems-II: Express Briefs, vol. 61, no. 12, pp. 962-966, Dec. 2014.
- [5] McGovern B.P, R.F. Woods, and C. McAllister, "Optimised Multiply/Accumulate Architecture for Very High Throughput Rate Digital Filters", IET Electronic Letters, vol. 31, no. 14, pp. 1135-1136, Jul. 1995.

