International Journal of Computing and Network Technology
ISSN (2210-1519)
Int. J. Com. Net. Tech. 4, No. 1 (Jan.-2015)

Low Power FIR Filter using Karatsuba Multiplier

Shiny P. Wilson[1] and Ramesh P.[2]

[1]*Department of Electronics and Communication, CUSAT, College of Engineering Munnar, Kerala, India*
[2]*Department of Electronics and Communication, CUSAT, College of Engineering Munnar, Kerala, India*

Received: 20 Sept. 2015, Revised: 8 Dec. 2015, Accepted: 12 Dec. 2015, Published: 1 (January) 2016

Abstract: High speed multipliers are the primary requirement of multi-core processors because of key applications like high performance and security requires well designed and reliable hardware implementations. The overall performance of a particular processor is based on the speed of the multiplier unit inside it. This paper presents implementation of low power FIR Filter using karatsuba multiplier. Multiplier is designed using gates. Implementation of FIR Filter using karatsuba multiplier and array multiplier is done .The simulation of various parameters such as delay, area and power are carried out and these parameters are reduced in FIR filter using karatsuba multiplier. Estimated power of filter using array multiplier and karatsuba multiplier is 5.7μW and 3.13 μW respectively. 45% of power and 37% of area can be reduced by using this implementation.

Keywords: Array Multiplication**,** Karatsuba Algorithm, FIR filter,Verilog and CADENCE.

1. INTRODUCTION

Multipliers are the fundamental and necessary building blocks of many high performance systems. Multiplication is commonly used operation which is now implemented in many processors, attaining security objectives of data integrity and confidentiality for high speed network applications. The speed of the multiplier determines the speed of the system; hence the speed of the multiplier has to be increased. The performance of the system depends upon the multiplier's speed which is optimized by the proposed multiplier. Low power consumption is also a critical issue in multiplier design. The main contributors to the power consumption of digital circuits are adders, multipliers, comparators and shifters. Power reduction is now becoming very crucial for implementation of VLSI designs. With increasing importance of power reduction, it is becoming essential to evaluate different data path architectures from the point of view of both delay and power.

Karatsuba Algorithm (KA) is the fast multiplication algorithm which helps to increase the speed of the multiplier, and is one of the fastest methods to multiply long integers. The classical multiplication algorithm multiplies every digit of a multiplicand by every digit of the multiplier and adds the result to the partial product. KA has less complexity than classical schoolbook method and thus it multiplies large numbers faster. The Karatsuba Algorithm is more efficient for multiplication of large numbers. It uses Divide and Conquer strategy, divides a multiplication operation into smaller and smaller products recursively, and then computes the result of the multiplication, benefiting from the results of these smaller products. Thus it saves coefficient multiplications at the cost of extra additions as compared to the ordinary multiplication method. The Karatsuba Algorithm is more efficient for multiplication of large numbers and the number of multiplication is reduced by increasing the number of additions, thus reducing the overall cost of the hardware design.

Multiplication can be done by using some smaller multiplication, addition and some shifting operations.

E-mail address: shiniwilson19@gmail.com, rap@ieee.org

Thus speed of the multiplier can be increased by using high speed adder. This paper presents the design of a fast multiplier using the Karatsuba Algorithm to multiply two numbers using the technique of polynomial multiplication. Here the coding is done on Verilog HDL and synthesis is done by using Xilinx ISE 14.7 and further analysis is done through Cadence Encounter Tool.

2. ARRAY MULTIPLIER

An array multiplier is a parallel multiplier which does shift and add all at once. This multiplier is called an array because it has array adders. An array multiplier also uses shift and add operation as in binary multiplier but it adds the partial products in parallel [1]. Each partial product is generated by the multiplication of the multiplicand with one multiplier bit. The partial products are shifted according to their bits orders and then added. Addition is mainly done by carry save algorithm in which every carry and sum signal is passed to the adder of the next stage. Final product is obtained in a final adder by any fast adder. In array multiplier we need to add as many partial product as many partial product as their in multiplier bit. An array multiplier is well known for its regular structure. Figure 1 shows an array multiplier. Here A=a1a0 and B= b1b0 and product P= P3P2P1P0.

3. KARATSUBA ALGORITHM

The Karatsuba algorithm is a fast multiplication algorithm. It was first implemented by Anatolii Alexeevitch Karatsuba in 1960 and published in 1962[2]. It scales down the multiplication of two n-digit numbers to two single-digit multiplications. Multiplying two polynomials efficiently is a crucial point in different applications such as signal processing, cryptography and coding theory. The Karatsuba Algorithm (KA) reduces coefficient multiplications at the cost of extra additions compared to the schoolbook or ordinary multiplication method [3,4]. We consider the KA to be efficient if the total cost of using it is less than the cost of the ordinary method.

In KA number of multiplication is reduced by increasing the number of additions, thus reducing the overall cost of the hardware design. Thus, the efficient software implementations of the multiplication operation in the finite fields are desired in cryptographic applications, particularly in the elliptic curve cryptography [5]. Several new methods for basic arithmetic operations in the finite fields, suitable for software implementations have been recently developed [2]. Among these algorithms, the Karatsuba-Ofman Algorithm (KOA) [6,7] has a special place since it is the only practical algorithm which is asymptotically faster than the standard methods for the cryptographic applications in which the numbers in the range 160 to 1024 bits are used. However, most of these implementations report speedup only for higher bit lengths, i.e., bit lengths of 1024 or more. Multiplication can be done by using some smaller multiplication, addition and some shifting operations. Thus speed of the multiplier can be increased by using high speed adder. KOA could be used in recursive mode and applied for any degree m, utilizing the scheme will yield more gate savings with longer delay. It is used for polynomial multiplication. A main advantage of KOA approach could be contributed to its recursive possibility.

A. *Karatsuba Algorithm for integer multiplication*

By using Karatsuba algorithm, the product of two numbers can be calculated using three multiplications of smaller numbers along with some additions and digit shifts.

Let a and b represent two n-digit numbers with some radix R, each number can be divided as

$$a=a_1R^m+a_0 \text{ and } b=b_1R^m+b_0 \quad (1)$$

Then the product *ab* can be calculated as

$$ab=(a_1R^m+a_0)(b=b_1R^m+b_0)$$
$$=a_1b_1R^{2m}+(a_1b_0+a_0b_1)R^m+a_0b_0$$
$$ab =u_2 R^{2m}+u_1 R^m+u_0 \quad (2)$$

Here $u_2= a_1b_1$,$u_0= a_0b_0$ can be written as

$$u_1= (a_1+a_0)(b_1+b_0)- a_1b_1- a_0b_0 \quad (3)$$

B. *Karatsuba Algorithm for Polynomial Multiplication*

The Karatsuba method can be used to reduce the execution time for large operand multiplication. This method replaces some multiplications with additions and subtractions.

i. *KA for Degree-1 Polynomials*

Consider two degree-1 polynomials $A(x)$ and $B(x)$. Here

$$A(x) = a_1x + a_0 \text{ and } B(x) = b_1x + b_0$$

Then the polynomial $C(x) = A(x)\,B(x)$ can be calculated in the following way:

$$C(x) = (a_1b_1)\,x^2+(a_0b_1+a_1b_0)x+a_0b_0 \quad (4)$$

Coefficient of x $(a_0b_1+a_1b_0)$ can be written as

$((a_0+a_1)(b_0+b_1)-a_0b_0-a_1b_1)$

Then $C(x)=(a_1b_1)x^2+((a_0+a_1)(b_0+b_1)-a_0b_0-a_1b_1)x+a_0b_0$

Let D_0, D_1, $D_{0,1}$ be auxiliary variables with

$D_0= a_0b_0$, $D_1=a_1b_1$,$D_{0.1}= (a_0+a_1)(b_0+b_1)$

$C(x) = D_1x^2+ (D_{0.1}-D_0-D_1)x+D_0$ (5)

Here four additions and three multiplications needed to compute $C(x)$. Using the schoolbook method need four multiplications and one addition, thus this technique can save one multiplication and need three extra additions.

ii. *KA for Degree-2 Polynomials:*

Consider two degree-2 polynomials:

$A(x) = a_2x^2+a_1x+a_0$ *and* $B(x) = b_2x^2+b_1x+b_0$, with Auxiliary variables $D_0= a_0b_0$, $D1 = a_1b_1$, $D_2 = a_2b_2$, $D_{0.1} = (a_0+a_0)(b_0+b_1)$, $D_{0.2} = (a_0+a_2)(b_0+b_2)$ *and* $D_{1.2} = (a_1+a_2)(b_1+b_2)$.

Then $C(x) = D_2x^4 + (D_{1.2}-D_1-D_2)x^3 + (D_{0.2}-D_0-D_2+D_1)x^2 + (D_{0.1}-D_0-D_1)x+D_0$ (6)

The coefficients of x, $x^2$and x^3 in the above polynomial can be written as:

$(a_0b_1 + a_1b_0) = ((a_0 + a_1)(b_0 + b_1) – a_0b_0– a_1b_1)$

$(a_0b_2 + a_2b_0) = ((a_0 + a_2)(b_0 + b_2) -a_0b_0– a_2b_2)$

$(a_1b_2 + a_2b_1) = ((a_1+ a_2)(b_1 + b_2) – a_1b_1 – a_2b_2)$

We need 13 additions and 6 multiplications. Using the schoolbook method we need 4 additions and 9 multiplications.

iii. *KA for Degree-5 Polynomials:*

Consider the two polynomials $A(x)$ and $B(x)$

$A(x) = a_5x^5+ a_4x^4+ a_3x^3+ a_2x^2+a_1x+a_0$,

$B(x) = b_5x^5+ b_4x^4 + b_3x^3+ b_2x^2+b_1x+b_0$

Then $A(x)$ and $B(x)$ can be written as

$A(x) = A_1(x)x^3+A_0$, $B(x)=B_1(x)x^3+B_0$

With $A_1(x)=a_5x^2+a_4x+a_3$ $A_0(x)=a_2x^2+a_1x+a_0$

$B_1(x)=b_5x^2+b_4x+b_3$ $B_1(x)=b_2x^2+b_1x+b_0$

Now we apply the KA for degree-1 polynomials. Notice that the coefficients of the polynomials A(x) and B(x) are themselves polynomials and the multiplications of these coefficients result in further applications of the KA for degree-2 polynomials. Let D_0, D_1, $D_{0,1}$ be auxiliary variables with

$D_0=A_0B_0$, $D_1=A_1B_1$, $D_{0.1}= (A_0+A_1)(B_0+B_1)$.

Thus we obtain

$C(x)= D_1x^6+(D_{0.1}-D_0-D_1)x^3+D_0$ (7)

The KA for degree-1 polynomials needs four additions and three multiplications of degree-2 polynomials. Each multiplication is solved by the KA for degree-2 polynomials for which we need 13 additions and 6 multiplications. Overall we need 18 multiplications and 59 additions.

4. Architecture of Karatsuba Multiplier

The basic building blocks for a Karatsuba Multiplier for polynomial multiplication are AND gate and XOR gate. Figure 2 shows block diagram for a 2 Bit Karatsuba Multiplier for polynomial multiplication.

5. FIR Filter Design

Finite impulse response (FIR) filters are highly used in most of the DSP applications. Most of the applications, the FIR filter circuit must be able to operate at high sampling rates, while in many applications, the FIR filter circuit[8] must be a circuit operating at moderate sample rates and uses low power. Various techniques can be applied to digital FIR filters to either increase the effective speed or reduce the power consumption of the original filter. Less work has been done that dealing with reducing the hardware complexity or power consumption of FIR filters [9]. Since many years parallel processing applied to an FIR filter involves the hardware unit's replication that exists in the original filter. The choice of the multiplier and adder circuit also affects the resultant power dissipation. If the multiplier is chosen wisely with less number of calculations speed and power can be optimized [10].FIR filter provides variable length taps have been widely used in many application fields. It is memory chip in which an address generation unit & modulo unit to access memory in a circular manner. A simple FIR filter [11] is described by a convolution operation. Figure 3 shows Block Diagram of n- Tap FIR Filter.

6. Implementation

In this project the coding was done using the Verilog HDL and software used is Xilinx ISE 14.7 provided by Xilinx. Performance analysis is done by using CADENCE. Figure 4 and 5 shows the simulation results of 8 bit array multiplier and 8 bit KM using polynomial multiplication. Figure 6 and 7 shows simulation result of 4-tap FIR Filter using array multiplier and 4-tap FIR Filter using karatsuba multiplier and figure 8, 9 shows its RTL view using cadence RTL compiler.

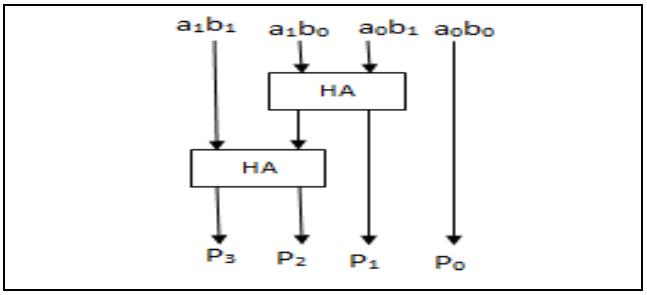

Figure 1: Block Diagram of 2 bit Array Multiplier

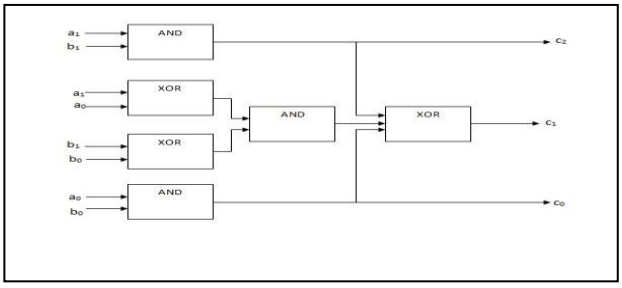

Figure 2: Block Diagram of 2 bit Karatsuba Multiplier for polynomial multiplication

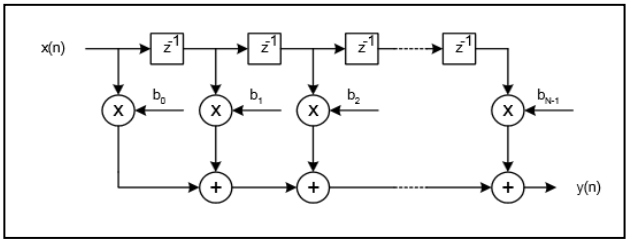

Figure 3: Block Diagram of n-tap FIR Filter

A. The Verilog Language

Many DSP applications demand high throughput and real-time response, performance constraints that often dictate unique architectures with high levels of concurrency. DSP designers need the capability to manipulate and evaluate complex algorithms to extract the necessary level of concurrency. The Verilog language supports the modeling at the algorithm or behavioural level, and at the implementation or structural level. It provides a versatile set of description facilities to model DSP circuits from the system level to the gate level. At the system level we can build behavioural models to describe algorithms and architectures. In many respects Verilog is a very powerful, high-level, concurrent programming language.

B. XILINX

Xilinx ISE (**I**ntegrated **S**ynthesis **E**nvironment) is a software tool produced by Xilinx for synthesis and analysis of HDL designs, enabling the developer to synthesize their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer. It is a design environment for FPGA products from Xilinx, and is tightly-coupled to the architecture of such chips, and cannot be used with FPGA products from other vendors. The Xilinx ISE is primarily used for circuit synthesis and design, while the ModelSim logic simulator is used for system-level testing.

C. CADENCE

Cadence is a powerful software tool which helps in both analog and digital ASIC design. With the help of Cadence Nc Launcher it is possible for the simulation of a HDL code. The simulated code can be synthesized by using the Cadence RTL complier which helps to provide net-list for the ASIC design. After the synthesize it is possible for the design to have different analysis like time, area, power etc. Cadence Encounter the power full tool helps us to provide the complete layout for the ASIC design.

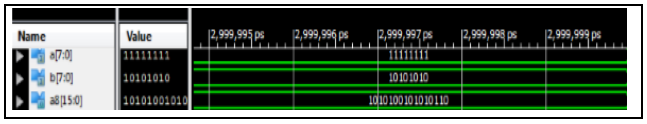

Figure 4: Simulation result of 8 bit Array Multiplier

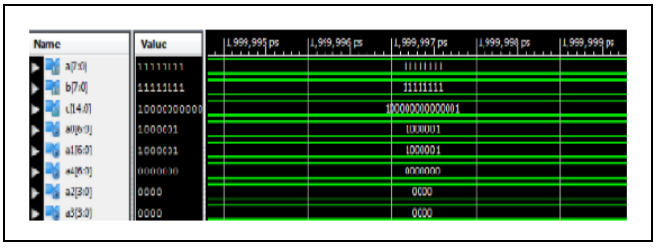

Figure 5: Simulation Result of 8 bit Karatsuba multiplier using polynomial multiplication

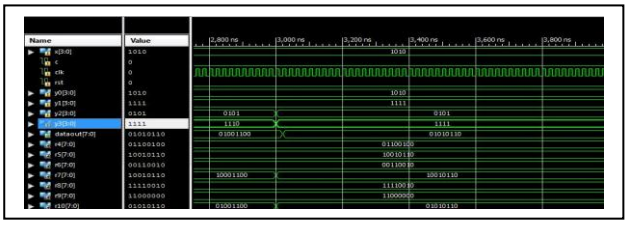

Figure 6: Simulation Result of FIR Filter using array multiplier

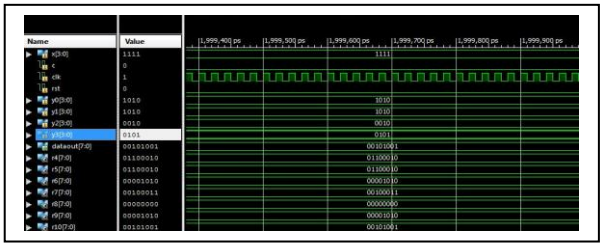

Figure 7: Simulation Result of FIR Filter using Karatsuba multiplier

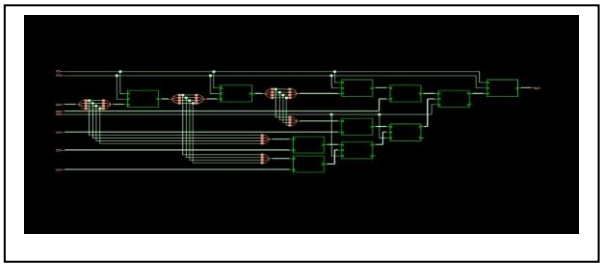

Figure 8: RTL view of 4-tap FIR filter using array multiplier

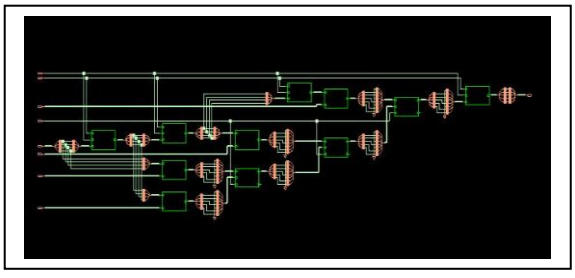

Figure 9: RTL view of 4-tap FIR filter using karatsuba multiplier

Table1: Analysis of Polynomial multiplier with different length

Analysis of Polynomial multiplier with different length			
Bit size	**Number of cells**	**Cell Area**	**Delay(ps)**
4	27	133.6	749
8	91	456.52	1003
16	198	1003.36	1200

Table 2: Analysis of FIR filter

Analysis of FIR Filter				
FIR filter using	**Number of cells**	**Cell Area**	**Delay (ps)**	**Power (nw)**
Array Multiplier	163	1416.84	179	5705.46
KM	147	895.41	180	3132.69

7. Results and Discussion

This work is implemented by the technique Karatsuba Algorithm using gates. FIR Filter using array multiplier and karatsuba multiplier are implemented in Verilog using Xilinx and the constraints area, power and timing are optimized using Cadence.

a. 4 bit ,8 bit and 16 bit karatsuba multiplier using polynomial multiplication is implemented using verilog code.

b. Analyse the speed of different bit multipliers using Cadence. The delay associated with 16 bit karatsuba multiplier is only 1.2 ns.

c. Compared the performance of Karatsuba Multiplier with array multiplier area, power and delay is less than array multiplier.

d. FIR Filter using array multiplier and karatsuba multiplier is implemented.

e. Finally we conclude that performance wise, FIR Filter using Karatsuba multiplier consumes less area and power. Here power and area is reduced by 45% and 37% respectively.

Table 1 shows analysis of KM for polynomial multiplication for different bit lengths. Table 2 shows delay, area and power analysis of FIR filter.

8. Conclusions

In This paper we presented a low power and low area FIR filter. Here low power consumption is achieved by using karatsuba multiplier. This filter was compared for area and power with filter using array multiplier and it demonstrated that this approach is most effective for implementations with the constraints of low area and low power. Among the different multiplication methods available multiplication based on KA for polynomial multiplication have high speed of operation due to its multiplication technique. The computational delay associated with proposed multiplier is very less, which gives better speed than other multipliers. Area and power of FIR filter is reduced by using polynomial multiplication. Besides these facts the performance of multiplier depends on the performance of adder circuits.Thus by implementing fast Karatsuba Multiplier, multiplying units in the current processors can be replaced by the proposed one, the area, power is less and

thus total cost can be reduced. Its computational delay is only 1.2ns, which gives better speed than other multipliers. Also this multiplier has a better performance on large bit data and it finds application in DSP and Cryptography.

ACKNOWLEDGMENT

It gives immense pleasure for me to acknowledge the assistance and contributions of many individuals in making this work a success. First and foremost, I would like to thank Dr. Ramesh P, Associate Professor, Department Of Electronics and Communication Engineering, College Of Engineering Munnar, for his assistance, ideas, and feed backs during the process in doing this work. Secondly, it is a pleasure to express my thanks to all beloved teachers in the Electronics and Communication Department. I deeply appreciate their helpfulness and willingness in providing the useful information for this study. Above all I would like to thank Almighty, my family and all my friends who rendered all possible help in doing this work. I also express heartiest thanks to all those who helped me in one way or other for completing this work successfully.

REFERENCES

[1] A. Somani, D. Jain, S. Jaiswal, K. Verma, and S. Kasht, "Compare vedic multipliers with conventional hierarchical array of array multiplier."

[2] A. Karatsuba and Y. Ofman, "Multiplication of multi digit numbers on automata," in Soviet physics doklady, vol. 7, 1963, p. 595.

[3] E. Triveni.K.S, "High speed efficient karatsuba- ofman pipelined multiplier for low contrast image enhancement," in International Journal of Engineering and Advanced Technology, 2013.

[4] A. Weimerskirch and C. Paar, "Generalizations of the karatsuba algorithm for efficient implementations". IACR Cryptology, e-Print Archive, vol. 2006, p.224, 2006.

[5] A. J. Menezes, Elliptic curve public key cryptosystems.Springer 1993, vol. 234.

[6] D. E. Knuth, "The art of computer programming, vol. 2: Semi-numerical algorithmsaddison-wesley," Reading, MA, pp. 229–279, 1969.

[7] S. Veeramachaneni and M. Srinivas, "New improved 1-bit full adder cells," in Electrical and Computer Engineering, 2008. CCECE 2008 Canadian Conference on IEEE, 2008, pp. 000 735–000 738.

[8] Sarita Chouhan and Yogesh Kumar, "Low power designing of FIR Filters" in International Journal of Advanced Technology & Engineering Research (IJATER), Volume 2, Issue 2, May 2012.

[9] Kadu, Mr Pravin Y and Dhengre, Ku Shubhangi , "Low Power and High Speed Design Of FIR Filter", in IORD Journal of Science & Technology 2014.

[10] Bahram, Rashidi, Farshad Mirzaei, Bahman Rashidi and Majid Pourormazd "Low Power FPGA Implementation of Digital FIR Filter Based on Low Power Multiplexer Base Shift/Add Multiplier" International Journal of Computer Theory and Engineering, Vol. 5, No. 2, April 2013.

[11] H.J. Kang, H. Kim and I.-C. Park, "FIR filter synthesis algorithms for minimizing the delay and the number of adders" presented at Computer Aided Design, 2000.ICCAD-2000. IEEE/ACM International Conference on, 2000.