

Low Power and Area efficient Linear Convolution by using Cook-Toom Multiplier

K.MOGILEESWARA¹, Ms.BHARATHI² and Mr.KRAMESH³

¹ Student, M.Tech-VLSI,

Department of ECE, SVEC, Tirupati.

Mobile No: 7893456094

kmmogileeswar@gmail.com

² Assistant Professor,

Department of ECE, SVEC, Tirupati.

Mobile No: 8919162867

bharathi891@gmail.com

³ Assistant Professor,

Department of ECE, SVEC, Tirupati.

Mobile No: 9652582053

kurakularamesh562@gmail.com

ABSTRACT:

Digital signal processing systems, Digital control systems, Telecommunication systems, Speech and audio processing for audiology are widely used VLSI applications. In modern days, VLSI is the design and implementation of DSP systems and these are very important for above applications. In most of the DSP systems, adders and multipliers are the basic building blocks. If we design an efficient adders and multipliers then we can achieve efficient design for Digital signal processing (DSP) systems. Convolution and Linear time invariant (LTI) systems are one of the fundamental applications of Digital signal processing (DSP) systems. The behaviour of an LTI systems in continuous time is described by Convolution integral whereas the behaviour in discrete-time is described by linear convolution [8][9]. In this paper an efficient multiplier Cook-Toom multiplier is used to obtain linear convolution. The Cook-Toom multiplier is working based on the principle divide and conquers which reduces the complexity and improves the speed [16]. The coding is done using VHDL (VHSIC Hardware Description Language). Simulation and Synthesis is performed by using the Xilinx Tools.

Keywords: DSP, LTI, Convolution, Cook-Toom multiplier

I. INTRODUCTION

In microelectronics, a multiplier is a factor of proportionality that measures how much an endogenous variable changes in response to a change in some exogenous variable [1][2]. Multiplication is an essential arithmetic operation for Digital Signal Processing (DSP) applications. Convolution, correlation, Fast

Fourier Transform (FFT), filtering, and etc are the widely used DSP applications. For any VLSI design we must concentrate on the following parameters such as speed, Power and Area. To achieve high execution speed, Parallel Array [2] multipliers are widely used for DSP applications. But these array multipliers consume more power because to design an array multiplier we need large Area. So power consumption is the critical concern in today's VLSI system designs. So the designers are must look onto the power efficient multipliers for the design of Low-power Digital Signal Processing (DSP) systems [9]. In general a basic multiplier can be divided into three parts to achieve efficient multiplier design such as (i).Partial product generation. (ii).Partial product addition and (iii).final addition. If we can reduce the number of partial products in the multiplication then we can reduce the delay for the design i.e high speed can be achieved for such multipliers.

II. RELATED WORKS

A. ARRAY MULTIPLIER

The array multiplier is an efficient layout of combinational multiplier. The array multiplier can be chosen to achieve high speed for the design. In array multiplier, multiplication of two binary numbers can be obtained with one micro operation by using combinational circuit that forms the product bits, all at once thus making it a fast way of multiplying two numbers. Since only the delay is the time for the signals to propagate through the gates that forms the multiplication array. For an $N \times N$ array multiplier requires $N(N-2)$ full adders, N half adders and N^2 AND gates in his internal design. In this paper, we present a Low power array multiplier by using modified full adder. For any VLSI design delay is also one of the important parameter, in worst case the delay of an array multiplier would be $(2N+1)td$ [1][2][3][4].

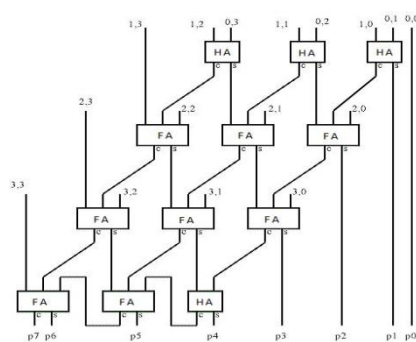


Figure 1: 4×4 Array multiplier

Hence by seen an array multiplier design we can easily said that, it requires large chip area and also it requires high power for the design, so we need to concentrate on Low power for the design. As we said earlier full adder is the basic element to design an array multiplier, hence if we decrease the power in design of full adder then automatically power consumption of the array multiplier will also decreased [1]. Now we can see some of the modified full adder designs to reduce the power consumption.

B.CONVENTIONAL FULL ADDER

As shown in below figure 2 the conventional full adder consists of two XOR gates, two AND gates and one OR gate. Due to the presence of XOR gates in

conventional full adder, it requires high power because XOR gate consumes more power. The delay for this conventional full adder for the path of sum is

$$\text{Delay} = 2 * \text{XOR gate delay}$$

And one XOR gate, one AND gate and one OR gate are included in the critical path of the carry.

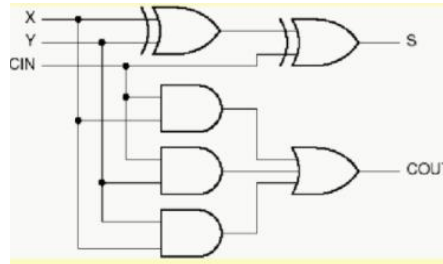


Figure 2: conventional full adder

C.MUX BASED FULL ADDER

The full adder using 4:1 MUX based is shown in below figure 3 and it consists of one NOT gate and two 4:1 MUXs. This modified full adder using 4:1 multiplexers are used to reduce the power. By implementing the MUX based full adder the reduction phase of the array multiplier power reduction has been achieved.

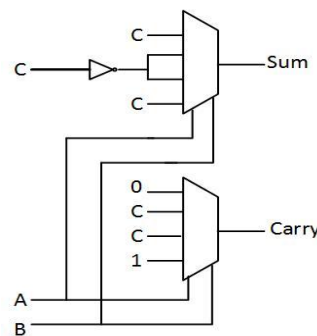


Figure 3: 4:1 MUX based Full adder

But more area is required while using this 4:1 MUX based full adder to design an array multiplier because to design full adder we need two 4:1 multiplexers. Hence we cannot reduce the power and delay for this type of circuits. So we need to go for another design which is 2:1 MUX based full adder. For the design of 4:1 MUX based full adder the critical path delay for sum output will be

$$\text{Delay} = \text{NOT} + \text{MUX}$$

D.2:1 MUX BASED FULL ADDER

2:1 MUX based full adder is one of the best designs to achieve Low power full adder. To design 2:1 MUX based Full adder we required one XOR gate and two 2:1 multiplexers to generate sum and carry. The design of 2:1 MUX based full adder is shown in the below figure4.

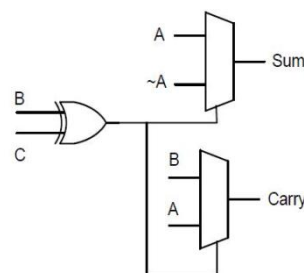


Figure 4: 2:1MUX based full adder

By seeing the above structure of full adder, one XOR block in the conventional full adder is replaced with multiplexer block so that the critical path delay would be minimized. The critical path delay is given by

$$\text{Delay} = \text{XOR} + \text{MUX}$$

This full adder can be implemented by using XOR output as selection line. Since XOR employs most of the power consumption in the adder circuit, if we reduce the number of XOR gates in the design then power consumption of the full adder can be reduced. But even though by using these designs for the array multiplier we cannot reduce the much power because area required to design an array multiplier is more. Suppose if we considered 16 bit for the array multiplier then we required 256 AND gates, 16 half adders and 224 full adders. By seeing this example we can easily understand how much area is required and how much power is required for all components. And as the bit size is increased then array multiplier is very complex and hard to design.

III. COOK-TOOM MULTIPLIER

Cook-Toom, also known as Toom-3, named after Andrei Toom, who introduced the new algorithm with its low complexity, and Stephen Cook, who cleaned the description of it, is a multiplication algorithm, method of multiplying two large integers [16]. The Cook-Toom algorithms follow the divide and conquer method for multiplying large integers. Just like Karastuba, it splits the given integer into 'k' limbs of some fixed size. The Cook-Toom multiplier works on the principle of polynomial multiplication [16]. The operands in the multiplication are divided into limbs of given size, and each in the form of polynomial, and the limb size is used as radix. In Cook-Toom multiplication, instead of multiplying the obtained polynomials directly, we evaluated at asset of points, and values multiplied together at those points. Based on the products obtained at those points the product polynomial is obtained. At last the final result is then obtained by substituting the radix [16].

For example let us consider two large integers 'A' and 'B', Cook-Toom splits up 'A' and 'B' into 'k' smaller parts each of length 'l', and performs operations on the parts. As limb size 'k' grows one may combine many of multiplication sub operations, thus reducing the overall complexity of algorithm. The multiplication sub operations can be computed recursively by using Cook-Toom multiplication again, and so on. Although the terms "Toom-

3” and “Cook-Toom” are sometimes incorrectly used interchangeably [16], Toom-3 is only single instance of Cook-Toom algorithm, where $k=3$.

Toom-k reduces $k*k$ multiplications to $(2k-1)$, for example if $k=2$ then it reduces 9 multiplications to 5 and runs $\Theta(n^{\frac{\log(5)}{\log(3)}})$, about $\Theta(n^{1.465})$. In general Toom-k multiplier runs in $\Theta(c(k)n^e)$, where $e=\log(2k-1)/\log(k)$, n^e is the spent time on sub multiplications and ‘C’ is the time spent on additions and multiplication by small constants.

Let us considered an example for Cook-Toom multiplier, where a, b are operands and ‘w’ is the multiplication result.

Let $a=129$, $b=111$

Now we split a, b in the polynomial form as $X(t)$ and $Y(t)$ respectively, and they are written as $X(t)=x_2t^2+x_1t+x_0$, $Y(t)=y_2t^2+y_1t+y_0$

Where $t=10$, $x_0=9$, $x_1=2$, $x_2=1$ and $y_2=1$, $y_1=1$, $y_0=1$

Then the final result of the multiplication will be obtained from the following polynomial equation [16].

$$W(t) = w_4t^4 + w_3t^3 + w_2t^2 + w_1t + w_0$$

In the above polynomial the constant terms w_0, w_1, w_2, w_3, w_4 are obtained from the following equations directly as shown below.

In Cook-Toom -3 way method ‘t’ has five values which are ‘0’, ‘1’, ‘-1’, ‘2’ and ‘ ∞ ’ which are helpful to get the values of the above constants.

If we substitute the values of ‘t’ in the above polynomial then we get

$$W(0) = w_0$$

$$W(1) = w_4 + w_3 + w_2 + w_1 + w_0$$

$$W(-1) = w_4 - w_3 + w_2 - w_1 + w_0$$

$$W(2) = 16w_4 + 8w_3 + 4w_2 + 2w_1 + w_0$$

$$W(\infty) = w_4$$

By solving the above equations we get the values for w_0, w_1, w_2, w_3, w_4 or in other way we get the values directly by substituting the values of the constants obtained from $X(t)$ and $Y(t)$ then

$$w_0 = x_0y_0 = 9, w_1 = x_0y_1 + x_1y_0 = 11, w_2 = x_0y_2 + x_1y_1 + x_2y_0 = 12, w_3 = x_1y_2 + x_2y_1 = 3$$

$$\text{And } w_4 = x_2y_2 = 1$$

Now at final we substituting the values in the $W(t)$ then we gets

$$W(t) = w_4t^4 + w_3t^3 + w_2t^2 + w_1t + w_0$$

$$X(t)*Y(t) = t^4 + 3t^3 + 12t^2 + 11t + 9$$

$$a * b = 10000 + 3000 + 1200 + 110 + 9$$

$$129 * 111 = 14319$$

IV.CONVOLUTION

Convolution is defined as the integral of the product of two functions after one is reversed and shifted. The Linear and time invariant (LTI) systems are one

of the important class of systems and in general they have many digital signal processing applications. And these Linear and Time Invariant (LTI) systems are satisfied both linearity and time invariance properties [9][10].

An LTI system is purely characterized by its impulse signal. The impulse response is a response of system to the impulse signal or sequence [11].

In continuous time, the LTI system with input signal $x(t)$, output signal $y(t)$ and our impulse response $h(t)$ are related to convolution integral as,

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(t - \tau)h(\tau)d\tau = \int_{-\infty}^{\infty} x(\tau)h(t - \tau)d\tau$$

Where $h(t)$ is the impulse response.

In other words linear convolution for discrete time sequences, the output sequence $y[n]$ is the linear and time invariant system, with input $x[n]$ and due to the impulse response $h[n]$. And Convolution is written in the form of equation for an LTI system is

$$Y[n] = x[n] * h[n] = \sum_{k=-\infty}^{\infty} x[k]h[n - k]$$

The response of an output sequence can be expressed in terms of all input and impulse response sequences [12][17].

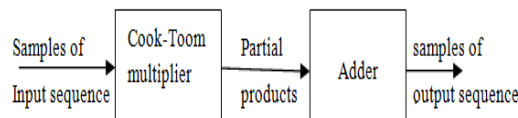


Figure 5: Block diagram of linear convolution

Therefore the length of the output sequence N , can be given for $y[n]$ is

$$N = L + M - 1$$

Where L , M are lengths of input sequence $x[n]$ and impulse sequence $h[n]$ respectively.

Example: Let the input and impulse sequences are $x[n] = \{1, 2, 3, 4\}$ and $h[n] = \{2, 3, 4, 5\}$, respectively. And in this example the lengths of input and impulse sequences are $L=4$, $M=4$. Then the length of the output Sequence $y[n]$ is $N=7$ and $n_1 = 0$, $n_2 = 0$, the starting time of the output sequence is $n=0$ then the 1st sample with starting time $n=0$ is given by

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

$$y[0] = \sum_{k=-\infty}^{\infty} x[k]h[-k] = 2$$

$$y[1] = \sum_{k=-\infty}^{\infty} x[k]h[1 - k] = 7$$

Similarly $y[2]=16$; $y[3]=30$; $y[4]=34$; $y[5]=31$; and $y[6]=20$.

V.SIMULATION RESULTS

The simulation results for linear convolution is plotted in the below figures.



Figure 6: Output waveform of linear convolution

DISCUSSIONS:

In our simulation we consider $x[n]=\{2,3\}$ and $h[n]=\{4,5\}$ then

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

the length of an output sequence $y[n]$ is $N=2+2-1$ $N=3$

Now from the above equation, for $n=0$, $y[0]=x[0]*h[0]=2*4=8$

for the second term $n=1$, then $y[1]=x[1]*h[0]+x[0]*h[1]=3*4+2*5=22$

for the third term $n=2$, then $y[2]=x[1]*h[1]=3*5=15$

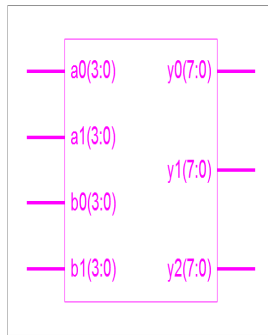


Figure 7: Convolution block diagram

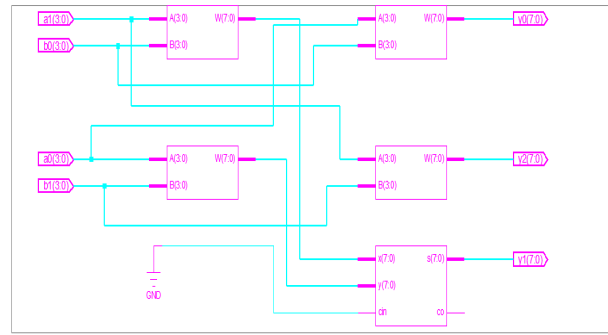


Figure 8: RTL schematic of linear Convolution

Table: comparison table for linear convolution

Parameter		Array multiplier	Cook-Toom Multiplier
Delay		13.931nsec	15.2nsec
Power consumption		0.56mwatts	0.48mwatts
Area	No. of slices	77 out of 4656	70 out of 4656
	No.of4i/p LUT's	127 out of 9312	123outof 9312

DISCUSSIONS:

By observing the above table the Cook-Toom multiplier has an advantage of low power consumption. Power was reduced by 17% as compared with the power of an array multiplier. And array multiplier has less delay but as the bit size increased then power consumption and area of the array multiplier is increased drastically. But we are dealing with large number systems in modern DSP applications of VLSI circuits.

VI.CONCLUSION

In this paper, linear convolution is performed for discrete time finite length sequences by using Cook-Toom multiplier. Here the proposed multiplier is Cook-Toom multiplier which has less path delay, high speed and low power consumption is obtained while perform in the Linear convolution. The path delay and power consumptions for linear convolution is 15.2nsec and 0.48mwatts respectively.

REFERENCES

- [1]. I.Thahira bhanu, Vishnu Priya and G.Usha “Low power array multiplier using modified full adder” 2nd IEEE International conference on Engineering and Technology (ICETECH), 17th &18th march 2016.
- [2]. C. Wallace, “A suggestion for a fast multiplier,” IEEE Transaction on Electronic Computers, Vol. EC-13, pp.14-17.
- [3]. R. S. Waters, E. E. Swartzlander, “A reduced complexity Wallace multiplier reduction,” IEEE Transactions on Computers, vol. 59, no.8, pp.1134-1137.
- [4]. Shahzad Asif and Yinan Kong, “Low Area Wallace Multiplier,” VLSI Design, vol. 2014.
- [5]. S.T. Oskuii, P. G. Kjeldsberg, “Power optimized partial product reduction interconnect ordering in parallel multipliers,” NORCHIP, Aalborg, 19-20 Nov2007.
- [6]. Karthick, S. Karthika and S. Valannathy, “Design and Analysis of Low Power Compressors,” International Journal of Advanced Research in Electrical, Electronics and instrumentation Engineering, You, Issue 6, Dec. 2012.
- [7]. M. Rabaey, Anantha Chandrasasan Borivoje Nikdic, 2003, Digital Integrated Circuits by Prentice-Hall.
- [8]. Purushotam D. Chidgupkar and Mangesh T.Karad, 2004, “The Implementation of Vedic Algorithms in Digital Signal Processing”, Global J. Of Engineering. Educ., Vol.8, **No.2**, UICEE Published in Australia.
- [9]. J.G. Proakis and D.G. Monalkies, 1988, Digital Signal Processing by Macmillan.
- [10]. A.V. Oppenheim and R. Schafer, 1975, Discrete Time Signal Processing Englewood Cliffs, N.Prentice-Hall.
- [11]. Asmita, Kamlesh Singh, 2011, “A Novel Approach For High Speed Block Convolution Algorithm”, prof. Of the International Conference on Advanced Computing & Communication Technologies (ACCT).
- [12]. Jagadguru Swami Sri Bharati Krishna Tirthji Maharaja, 1986, “Vedic Mathematics”, Motilal Banarsidas, Varanasi, India.
- [13]. R. Crandall and C. Pomerance. Prime Numbers –A Computational Perspective. Second Edition, Springer, 2005. Section 9.5.1: Karastuba and Toom–Cook methods, pg.473.
- [14]. M.Bharathi and D. Leela Rani “High Speed Carry Save Multiplier Based Linear Convolution Using Vedic Mathematics” International Journal of Computers & Technology Volume 4 No. 2, March-April, 2013, ISSN 2277-3061.

BIO-DATA:

	K.MOGILEESWARA is currently pursuing M.Tech 2 nd year in field of VLSI Design in the department of ECE at Sree Vidyanikethan Engineering College, Tirupati.
	Ms.M.Bharathi M.Tech., is currently working as an Assistant Professor in ECE department of Sree Vidyanikethan Engineering College, Tirupati. She has completed M.Tech in VLSI Design, in Satyabhama University. Her research areas are Digital System Design, VLSI Signal Processing.
	Mr K.RAMESH M.Tech., is currently working as an Assistant Professor in ECE department of Sree Vidyanikethan Engineering College, Tirupati. He is completed M.Tech in VLSI Design, in National Institute of Technology, Warangal.

