International Journal of Engineering & Technology, 7 (2.7) (2018) 733-738

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research Paper

Low power and high speed GDI based convolution using Vedic multiplier

C. Priyanka[1], N. Manoj Kumar [2] L.Sai Priya [2] B.Vaishnavi [2] M. Rama Krishna

[1]*Assistant Professor,* [2]*Students KLEF*
Department of ECE, Koneru Lakshmaiah Education Foundation, Vaddeswaram ,Guntur,Andhra Pradesh,India.
Corresponding author E-mail:chintapallipriyanka@kluniversity.in

Abstract

Convolution is having extensive area of application in Digital Signal Processing. Convolution supports to evaluate the output of a system with arbitrary input, with information of impulse response of the system. Linear systems features are totally stated by the systems impulse response, as ruled by the mathematics of convolution. Primary necessity of any application to work fast is that rise in the speed of their basic building block. Multiplier, adder is said to be the important building blocks in the process of convolution. As these blocks consumes plentiful time to obtain the response of the system. Several methods are designed to progress the speed of the Multiplier and adder, among all GDI (Gate Diffusion Input) is under emphasis because of faster working and low power consumption. In this paper GDI based convolution is implemented using Vedic multiplier and adder in T-SPICE Software which increases the speed and consumes less power compared to CMOS technology.

Keywords: *Linear convolution, CMOS, GDI, Adders, Vedic multiplier.*

1. Introduction

Now a day there is a rapid growth of portable electronic devices and has a high demand of speed and need low power consumption has become a challenge to researchers. In many of the building blocks of dsp applications convolution is the important block. If we reduce the size, power consumption and increase the speed of the convolution block, it improves the characteristics of the whole circuit. So, in this paper we are going to implement the convolution block using Vedic multiplier and GDI technique which satisfies the above criteria. Convolution block is used in many applications like in probability, computer vision, Fourier transforms, signal and image processing, statistics, natural language processing etc.

Because of using in these many applications, it improves the characteristics of many electronic devices. However, it is Some tough for the new candidate to accomplish convolution. As convolution technique is so long and consumes more time. So numerous procedures are projected for execute Discrete Convolution, one of a hard method is a Graphical method, it is quite systematic and sophisticated but, it is very extensive and time consuming. The foremost module for execute Convolution is Multiplier and. For performing linear convolution Pierre and John have executed the fast method. This technique is very Simple and Easy, it is like to accomplish simple multiplication of Decimal numbers [2]. And because of this technique is actual little time. it is possible to calculate Convolution of long sequences is very easily. Also, a GDI technique is used for execution of convolution. As Adder is also an important block for the proposed method, so all the probable adders are deliberated and synthesized using GDI technique. The Area of all adders and Delay is associated. Among all Adders which having less area occupy and highest speed is used for execution convolution. For the conventional multiplication, multipliers with Traditional shifts and add technique is used. This technique is difficult for VLSI operation and its Delay is huge. Vedic mathematics provides the unique solution for Multiplication. Vedic Multiplier based on Urdhava Tiryagbhyam sutra (Vertically and Crosswise) is used to implement Convolution.

Here the Vedic multiplier is designed using GDI technique for the implementation of convolution which reduces transistor count, consumes less power, and increases the speed of the convolution process. This technique is chosen because it is more advantage compared to other logic styles implementation like CMOS logic, Transmission gates and pass transistors logic. Pass transistor logic is a most used logic to overcome the designing issues of the circuits. GDI technique also improves the problem of CMOS and PTL logics that is voltage drop across N-channel. Thus, reducing in voltage drop is very important for low power devices. So here we are going to design convolution using Vedic multiplier and GDI.

2. Convolution

Convolution is one of the main part in Digital Signal Processing which is used to a large extent. One of the most efficient way to perform convolution is by performing multiplication in frequency domain. Generally, convolution is a process used to calculate the output response of a LTI (Linear Time Invariant) systems as these systems are independent from time we can calculate the output directly. Basically, there are two types of convolution. They are

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Linear convolution and Circular convolution. Linear convolution is a basic operation performed to calculate the output of any LTI system with predefined input and impulse response. Circular convolution is similar, but the signal will be periodic. In this paper we are going to implement linear convolution using Vedic Multiplier and GDI. Mathematically, we can acquire third signal based on two input signals. It helps to determine the systems output based on given arbitrary input. Features of the linear system are fully specified by the impulse response of the system, as constrained by mathematics of convolution. In Convolution operation we have two input functions which generates an output function (similar to multiplication and summation of two functions). The Convolution of two Discrete input sequences is given by [1]

$$s(n) = f(n) * g(n) \quad (1)$$

$$s(n) = \sum_{k=-\infty}^{\infty} f(k)(n-k) \quad (2)$$

2.1. Properties of a convolution

Convolution can be performed for any two functions with same variable, also for other purposes apart from knowing the output of the system for a given input sequence.

2.1.1. Convolution theorem

This is the main property. The Fourier transform of the convolution sequence will be same as the product of Fourier transforms of individual function.

$$F\{f*g\} = F\{f\}F\{g\}$$

2.1.2. Commutative

Exchange of functions is possible $f*g=g*f$

2.1.3. Associative

Convolution can be done in any order if there are multiple sequences $f*(g*h)=(f*g)*h$

2.1.4. Distributive

Sum of Convolution = Convolution of a sum

$$f*(g+h)=f*g+g*h$$

2.1.5. Scaling

Though, we multiply a constant at any stage of convolution, we will get the same result. $\alpha(f*g)=(\alpha f)*g=g*(\alpha f)$

2.1.6. Identity

The impulse function(delta) is multiplying the sequence with 1which returns the original function itself as output. $f*\delta=f$

2.1.7. Integration

Integral of convolution sequence = product of integrals of individual function $\int(f*g)(t)dt=(\int f(t)dt)(\int g(t)dt)$

2.1.8. Differentiation

Derivative of Convolution Sequence Product of derivative of any function with other $df*gdt=dfdt*g=f*ghdt$

EXAMPLE

Let us assume two sequences f(n)={2,3,4,5} and g(n)={1,2,3,4}.The process of convolution takes place in the following method.

f(n)	2	3	4	5			
g(n)	1	2	3	4			
	2	3	4	5			
		4	6	8	10		
			6	9	12	15	
				8	12	16	20
	2	7	16	30	34	31	20

Fig 2.1: Convolution Process

Suppose if the binary input sequence is considered the convolution can be performed similar to above methodology. In this paper convolution for two 4-bit binary sequences 1111 and 1111 respectively is performed using GDI based Vedic multiplier. After performing multiplication the outputs will be given to GDI based 4 bit adder for addition and the final output sequence is 11100001 (MSB to LSB).

3. GDI (Gate Diffusion Input)

To improve the performance of the circuits many CMOS technologies has been implemented. Pass transistor logic is the most popular logic among them. This implementation uses nmos transistors. In this technique a group of control signals has given to the nmos transistor's gate input and another group of data signals has been given to the source of the nmos transistor. This model has a advantage of low delay due to node capacitance and less interconnections, but has a disadvantage of threshold voltage drop across single channel pass transistor results in reducing supply voltage which is very important for low power designs.

Another technique is transmission gate technique which reduces the large circuits complexity by using a less number of complementary transistors. By using these technique, we can overcome a problem of low level swing of pmos and nmos. But it has a problem of consuming static power for low swing at the gates of the input transistors.

To overcome from the above difficulties DPL, LCPL, SRPL circuit models are designed but these techniques are also having a drawback of taking more area, top down logic design complexity etc. So, to overcome all these disadvantages we go for GDI technique which overcomes all the above defined problems.

The basic GDI cell looks like a traditional cMOS inverter as shown in Fig 3.1. But there are two main differences in this technique that makes this technique more efficient. They are,

the cell consists of three inputs to the three terminals namely G, P, N.

G-Input to the common Gate of nMOS or pMOS

P-pMOS source/drain input

N- nMOS source/drain input

Therefore, the main difference between the CMOS and GDI based design is in GDI cell VDD is not connected to source of PMOS and GND is not connected to source of NMOS. This gives two extra input pins for GDI which makes the design more flexible than CMOS design.

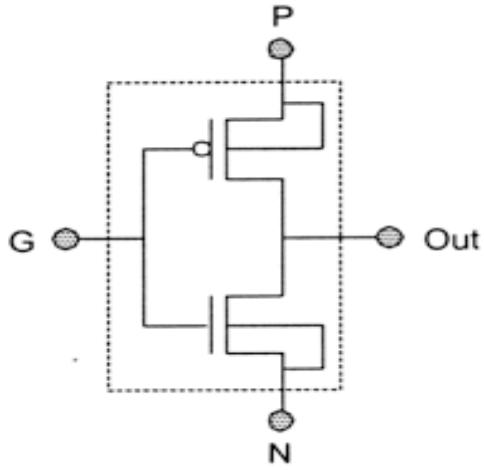


Fig 3.1: Basic GDI cell

3.1. Operational Analysis

As we know that normal PTL design methods have threshold drop across channel in pass transistors [2] because of this low swing at outputs will occur. GDI method will overcome this problem. To understand this method, we will take one example function F1 because it can be used in any gdi functions. The below table shows the logical functionality modes of F1.
From the table 1, the state of the output that low swing occurs is when A=0, B=0 only. Instead of 0V here the voltage level is Vtp. It is because of less high to low transition effect of a transistor [3]. Most of the GDI cells with input as B=1, the circuit operates as normal cmos inverter. When Vdd=1 instead of drop in the swing from the before stages the GDI cell acts as a inverter and brings the voltage swing. These cells contain self-swing restoration .

Table 3.1: Output functionality of F1 function

A	B	Functionality	F1
0	0	pMOSTrans Gate	V_{Tp}
0	1	CMOS Inverter	1
1	0	nMOSTrans Gate	0
1	1	CMOS Inverter	0

Thus this approach leads to design complex logical circuits with two transistors . These circuits are used in designing low power and fast circuits with less number of transistors . Because of 3 inputs it is a arbitary biased at contrast cmos inverter.
Here Vedic multiplier and adder is analyzed using GDI circuits and transient behavior of the circuit is done.

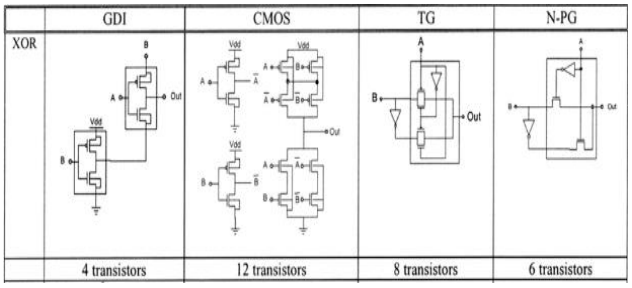
3.2. Full adder using GDI

The basic buliding blocks for the implementation of full adder are XOR gates and AND gates. Implementation of GDI based XOR gates and AND gates is shown below.

3.2.1. XOR Gate using GDI Cell

Table 3.2 shows the implementation of XOR Gate using GDI, CMOS,Transmission gates and pass transistors. From the table it is easily said that the tansistor count for GDI is less when compared to other techniques.Therefore it saves the area of the chip and reduces consuming more power.

Table 3.2: XOR Gate Implementation using GDI.CMOS,TG and N-PG



3.2.2. AND Gate using GDI Cell

Table 3.3 shows the AND Gate implementation using GDI and its comparision with other logic styles.From the table it is said that the tansistor count for GDI is less compared to other techniques.Therefore it saves the area of the chip and reduces consuming more power.

Table 3.3: XOR Gate Implementation using GDI.CMOS,TG and N-PG

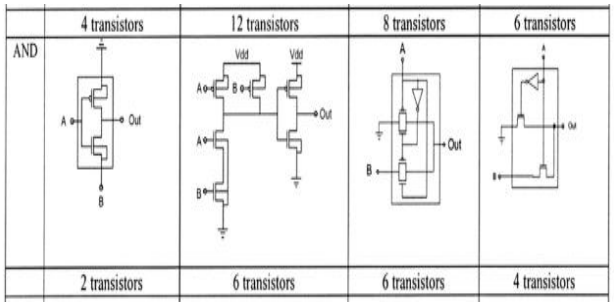


Fig 3.2 shows the GDI implemenation of the full adder, where A,B,C are the inputs of the full adder which produces the ouput sum and cout.

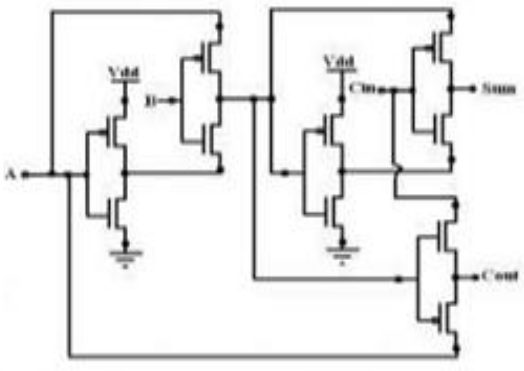


Fig 3.2: Full adder using GDI

From using this technique the transistor count is reduced to 10(GDI) from 36(CMOS). So we use GDI technique for implementation of convolution block.as it is very important in many digital applications.

4. Vedic Multiplier

From the ancient Indian Vedas, a new ancient system of mathematics was rediscovered and named as Vedic mathematics in the early twentieth century. The word 'Vedic' is derived from the word 'veda' which means knowledge. Age of Vedic Texts from 300BC.Vedic System reconstructed between 1911 and 1958 Bharati Krsna wrote one introductory value in 1958 “Vedic Mathematics” published in 1965.

Multiplier is one of the key hardware block used in most of the applications. Multiplier block itself is having huge delay with more amount of power dissipation. Therefore, we need a high-speed multiplier for the better performance and would meet the needs of high speed processors. Multipliers are used mostly in Microprocessors, DSP and Communication applications.
In many DSP applications multiplication is very important operation. By using Vedic multiplier, we can increase the speed of operation order in the basic building block of multiplier for arithmetic computation most of the DSP applications Demands for faster adder. Multipliers can be designed by using different adders as carry save adders, carry select adders and Manchester adders. In this paper for multiplication a systematic Vedic multiplier is using Urdhava Tiryagbhyam. this Vedic multiplier occupies less area and performs faster multiplication among the all multipliers[13-21]. By using conventional multiplier, it reduces the typical calculation which is difficult to compute the formula Urdhava Tiryagbhyam is applicable for all types of multiplications. The parallelism in generation of partial product improves the speed of multiplication. For implementing lengthy multiplication, the number is divided into small blocks and utilize for design. Some modification is required for higher number of bit. in this number is divided into two equal parts let's analyze 4 x 4 multiplication these are $X_3X_2X_1X_0$ and $Y_1Y_2Y_3Y_0$.The result of to two multiplication numbers is $M_7M_6M_5M_4M_3M_2M_1M_0$.Let's divide X and Y into two part For X ,X_3X_2 and X_1X_0,For Y,Y_3Y_2 and Y_1Y_0.By using Vedic multiplier technique take two bits at a time and perform multiplication using 2 bit multiplier the below structure shows the procedure or structure of 4 x 4 multiplication using Vedic multiplier.

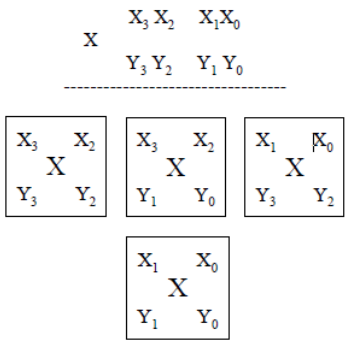


Fig 4.1: Block Diagram of 4X4 Vedic Multiplier

Each individual block shown in above diagram is a 2X2 multiplier. X_3X_2 represents the two-bit number and Y_3Y_2 represents the two bits of another number. Similarly, X_1X_0 represents another two bits and Y_1Y_0 represents with which X_1X_0 is to be multiplied. Let the final result of multiplication of 8-bit sequence be $M_7\,M_6\,M_5\,M_4\,M_3\,M_2\,M_1\,M_0$ as shown below.

$$\begin{array}{cccc} X_3\ X_2 & X_3\ X_2 & X_1\ X_0 & X_1\ X_0 \\ Y_3\ Y_2 & Y_1\ Y_0 & Y_3\ Y_2 & Y_1\ Y_0 \\ \hline M_{33}\,M_{32}\,M_{31}\,M_{30} & M_{23}\,M_{22}\,M_{21}\,M_{20} & M_{13}\,M_{12}\,M_{11}\,M_{10} & M_{03}\,M_{02}\,M_{01}\,M_{00} \end{array}$$

Fig 4.2: Decomposing 4-bit into each 2-bit multipliers to obtain intermediate partial products using vedic multiplier

Let the output of each block will be stored as shown above. For getting final result, the output of each block will be rearranged as shown in below diagram. Here, we will add the middle products as shown below.

$$\begin{array}{cccccccc} M_{33} & M_{32} & M_{31} & M_{30} & 0 & 0 & M_{01} & M_{00} \\ & & M_{23} & M_{22} & M_{21} & M_{20} & & \\ & & M_{13} & M_{12} & M_{11} & M_{10} & & \\ & & 0 & 0 & M_{03} & M_{02} & & \\ \hline M_7 & M_6 & M_5 & M_4 & M_3 & M_2 & M_1 & M_0 \end{array}$$

Fig 4.3: Final Partial Product of a 4-bit multiplier

M_1 and M_0 the direct outputs taken from M_{01} and M_{00} respectively. A 4-bit full adder is used for adding the (M_{02} M_{03} 0 0) with (M_{13} $M1_2$ M_{11} M_{10}) The result of this will be given as input to another full adder where as the other input is (M_{23} M_{22} M_{21} M_{20}). This result will be given as input to another full adder and added with (M_{31} M_{32} 0 0). The carry propagated will be added with M_6 . M_6 and M_7 are direct outputs from M32 and M33 respectively.

5. Implementation of Convolution Using GDI

Convolution using Vedic multiplier and adder is implemented by making use of GDI technique as it requires less transistor count and consumes less power. Fig 5.1 shows the block diagram for convolution.
Initially two 4-bit sequence is considered as input to a system. In order to obtain response of the system i.e., convolution, multiplication is the initial step that has to be performed. Hence multiplication is done using Vedic multiplier where the 4-bits is decomposed into four 2-bits each as shown in fig 4.1. The obtained intermediate partial products(fig 4.2) of each two bit multiplier is arranged as shown in fig 4.3.
Thus, in order to obtain final product a 4-bit adder is required for performing addition between intermediate partial products.
Fig 5.1 shows the block diagram for convolution using GDI. This is implemented using Tanner Tools T-spice v13.0 as shown below

.

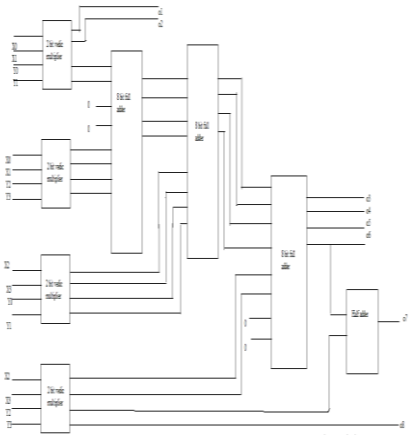


Fig 5.1: Block diagram of GDI based Convolution

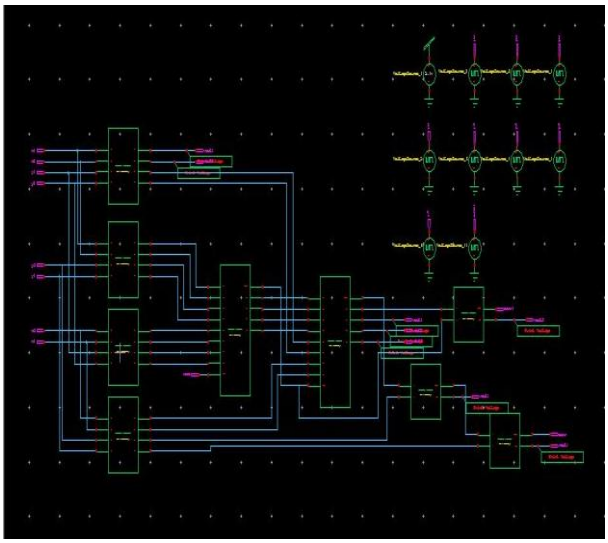


Fig 5.2: Implementation of GDI based convolution using T-SPICE Software

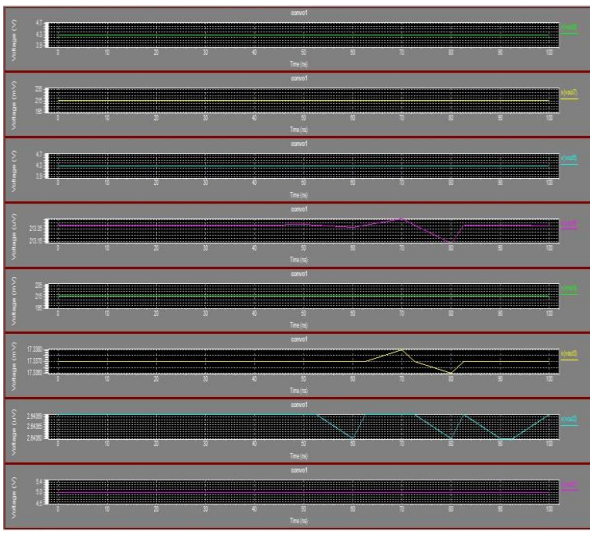


Fig 6.1: Simulation results for GDI based convolution

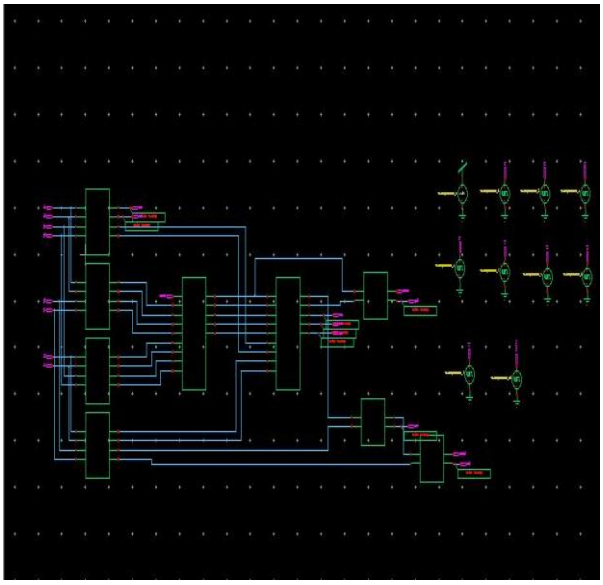


Fig 5.3: Implementation of CMOS based convolution using T-SPICE Software

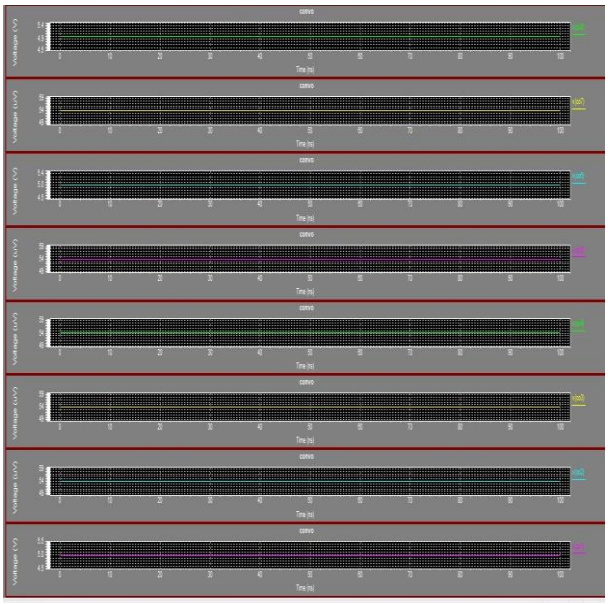


Fig 6.2: Simulation results for CMOS based convolution

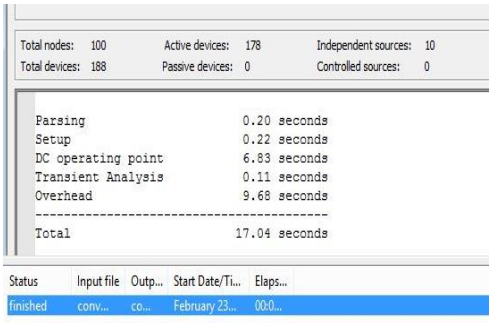
6. Outputs

Fig 6.1 and 6.2 shows the simulation results obtained using T-SPICE Software for GDI based convolution and CMOS based convolution technology for binary input sequence of x(n) = (1,1,1,1) and h(n)=(1,1,1,1). Therefore, the obtained output after performing the convolution between the two sequences is y(n) = (1,1,1,0,0,0,0,1)

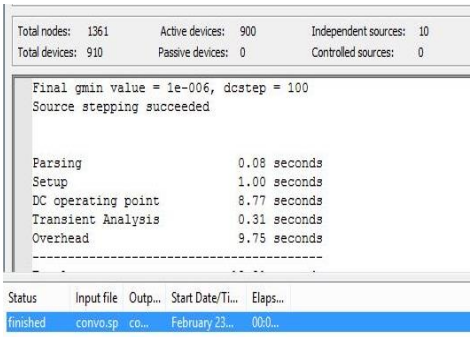
7. Results and Discussion

The time taken for simulation is 17.04 seconds using GDI technique which is less when compared to the CMOS which took 19.91 seconds. This small amount of time can further lead to large delay when we cascade the blocks.

USING GDI



USING CMOS



Density of Transistors

The number of transistors has been reduced to a great extent in GDI technology. This helps us in faster implementation of circuit and less power consumption and less delay.

	USING GDI	USING CMOS
Xor Gate	4	16
Half Adder	6	22
Full Adder	10	42
Convolution	178	900

Calculation of Power

In this paper, by using GDI technology the density of transistors is diminished to a great extent. So the power consumption is also reduced. Practically, the power consumption in GDI is 32.534 Watts and in CMOS it is 36.020 Watts.

References

[1] Arkadiy Morgenshtein , "Gate-Diffusion Input (GDI): A Power-Efficient Method for Digital Combinatorial Circuits Ieee Transactions Vol-10,No-5,Oct-2002.

[2] N. Weste and K. Eshraghian, *Principles of CMOS digital de- sign*. Reading, MA: Addison-Wesley, pp. 304–307.

[3] A. P. Chandrakasan, S. Sheng, and R. W. Brodersen, "Low-powerCMOS digital design," *IEEE J. Solid-State Circuits*, vol. 27, pp.473–484, Apr. 1992.

[4] A. P. Chandrakasan and R. W. Brodersen, "Minimizing power consump- tion in digital CMOS circuits," *Proc. IEEE*, vol. 83, pp. 498–523, Apr.1995.

[5] W. Al-Assadi, A. P. Jayasumana, and Y. K. Malaiya, "Pass-transistor logic design," *Int. J. Electron.*, vol. 70, pp. 739–749, 1991.

[6] Pierre, John W. "A novel method for calculating the convolution sum of two finite length sequences." Education, IEEE Transactions on 39.1(1996): 77-80.

[7] Jain, S. ; Saini S. "High Speed Convolution and Deconvolution algorithm (Based on Ancient Indian Vedic Mathematics) electricalengineering/electronics, computer, telecommunications and information technology (ecti-con), 2014 11th international conference on doi: 10.1109/ecticon.2014.6839756 Publication Year: 2014 , Page(s): 1 – 5.IEEE 2014

[8] Lomte, Rashmi K., and P. C. Bhaskar. "High Speed Convolution and Deconvolution Using Urdhva Triyagbhyam." VLSI (ISVLSI), 2011 IEEE Computer Society Annual Symposium on. IEEE, 2011.

[9] Itawadiya, Akhalesh K., et al. "Design a DSP operations using vedic mathematics." Communications and Signal Processing (ICCSP), 2013 International Conference on. IEEE, 2013.

[10] Bansal, Y. ; Madhu, C. ; Kaur, P." High speed Vedic Multiplier Design A Review" Proceedings of 2014 RAECS UIET Panjab UniversityChandigarh, 06 – 08.IEEE March, 2014

[11] Huddar S., Kalpana M., Mohan S."Novel High SpeedVedic Mathematics Multiplier Using Compressors" Automation, Computing,Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference pp: 465 – 469

[12] Senapati, Ratiranjan, Bandan Kumar Bhoi, and Manoranjan Pradhan" Novel binary divider architecture for high speed VLSI applications." Information & Communication Technologies (ICT), 2013 IEEE Conference on. IEEE, 2013.

[13] BALA DASTAGIRI, N. and HARI KISHORE, K., 2016. Analysis of low power low kickback noise dynamic comparators in pacemakers. Indian Journal of Science and Technology, 9(44),..

[14] BALA DASTAGIRI, N. and HARI KISHORE, K., 2016. Reduction of kickback noise in latched comparators for cardiac IMDs. Indian Journal of Science and Technology, 9(43),.),

[15] HUSSAIN, S.N. and KISHORE, K.H., 2016. Computational Optimization of Placement and Routing using Genetic Algorithm. Indian Journal of Science and Technology, 9(47),.

[16] MUDAVATH, M. and HARIKISHORE, K., 2016. Design of CMOS RF front-end of low noise amplifier for LTE system applications. Asian Journal of Information Technology, 15(20), pp. 4040-4047.

[17] MURALI, A., KAKARLA, H.K. and VENKAT REDDY, D., 2016. Integrating FPGAs with trigger circuitry core system insertions for observability in debugging process. Journal of Engineering and Applied Sciences, 11(12), pp. 2643-2650

[18] BALA GOPAL, P., HARI KISHORE, K., KALYANA VENKATESH, R.R. and HARINATH MANDALAPU, P., 2015. An FPGA implementation of onchip UART testing with BIST techniques. International Journal of Applied Engineering Research, 10(14), pp. 34047-34051

[19] BHARADWAJ, M. and KISHORE, H., 2017. Enhanced launch-off-capture testing using BIST design. Journal of Engineering and Applied Sciences, 12(3), pp. 636-643.

[20] VUNDAVILLI, P.R., PARAPPAGOUDAR, M.B., KODALI, S.P. and BENGULURI, S., 2012. Fuzzy logic-based expert system for prediction of depth of cut in abrasive water jet machining process. Knowledge-Based Systems, 27, pp. 456-464.

[21] KILARU, S., HARIKISHORE, K., SRAVANI, T., ANVESH CHOWDARY, L. and BALAJI, T., 2014. Review and analysis of promising technologies with respect to Fifth generation networks, 1st International Conference on Networks and Soft Computing, ICNSC 2014 - Proceedings 2014, pp. 248-251