ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

Area Efficient 16 Point Radix 4 Complex Fast Fourier Transform Algorithm for Efficient FPGA Implementation Using NEDA with Modified CSLA

Sangeetha Vijayan

PG Student [VLSI and Embedded Systems], Department of ECE, Saintgits College Engineering, Kottayam, Kerala, India

ABSTRACT: Fast Fourier Transforms (FFT), Discrete Cosine Transforms (DCT) are major blocks in communication systems.FFT is used to compute DFT with reduced number of arithmetic units. The major applications of FFT include signal analysis, image filtering, sound filtering, data compression, partial differential equations etc. The proposed design reports the architecture of 16 point complex FFT core using NEw Distributed Arithmetic (NEDA) algorithm. In order to implement FFT, radix-4 Decimation-In-Time algorithm is used. NEw Distributed Arithmetic is used for complex multiplications. It is one of the techniques used to implement many digital signal processing systems that require multiply and accumulate units. The advantage of the NEDA is that, it is a multiplier-less and ROM- less method and the entire section can be implemented using adders and shifters only, thus minimising the hardware requirement compared to other architectures. In the NEDA section, modified carry select adder with Binary-to-Excess one Converter (BEC) logic is used for addition. The design is simulated by using ModelSim SE(6.2b) and synthesised by Xilinx ISE project navigator(13.2).The synthesis results are taken for different Virtex FPGAs (Virtex 4,Virtex 5,Virtex 6).These results show that the computation for calculating the 16 point FFT is efficient in terms of area and power using the proposed method.

KEYWORDS: Fast Fourier Transform (FFT), FPGA, New Distributed Arithmetic (NEDA), radix-4,modified CSLA

I.INTRODUCTION

Today's electronic systems mostly run on batteries thus making the designs to be hardware efficient and power efficient. Application areas such as digital signal processing, communications, etc. employ digital systems which carryout complex functionalities. Hardware efficient and power efficient architectures for these systems are most required to achieve maximum performance.

Fourier analysis is named after Jean Baptiste Joseph Fourier (1768 to 1830), a French mathematician and physicist. Joseph Fourier, while studying the propagation of heat in the early 1800's, introduced the idea of a harmonic series that can describe any periodic motion regardless of its complexity. Later, the spelling of Fourier analysis gave place to Fourier transform (FT) and many methods derived from FT are proposed by researchers .In general, FT is a mathematical process that relates the measured signal to its frequency content.

The Fourier transform is the one of the several mathematical tools for analyzing the signals. It involves the decomposition of the signals in the frequency-domain in terms of sinusoidal or co sinusoidal components. The mathematical definition of a continuous FT (CFT) is given in the following.

$$X(\omega) = 1/\sqrt{2\pi} \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt \qquad (1)$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

where $x(t)$ is the original signal, $X(\omega)$ is the representation of signal in the frequency-domain, j is the imaginary number, ω is the angular frequency and t is the time index. The inverse of the continuous Fourier transform (ICFT) is defined as;

$$x(t) = 1/\sqrt{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} dt \qquad (2)$$

The continuous-time periodic signals and finite-energy non-periodic signals can be defined with Fourier series representation. In addition, the discrete-time signals can be represented within finite duration in practice. An alternative transformation is called Discrete Fourier Transform (DFT) for a finite-length signal, which is discretized in frequency. The frequency range of discrete-time signals is defined over the interval between $-\pi$ to π. A periodic digital signal consisted of N samples is to separate the frequency components into $2\pi/N$ radians intervals by dividing the frequency-domain. Then, Fourier series representation of the discrete-time signal will consist of N frequency components Kuo et al. (2001). The general Fourier series representation of a periodic signal ($x(n)$) is expressed as:

$$X(n) = \sum_{k=0}^{N-1} C_k e^{jk(2\pi/N)n} \qquad (3)$$

where N is the harmonic index related with the exponentials function ($e\ jk(2n/N)n$) for k=0, 1,…,N-1, ck is the coefficients of the Fourier series.The coefficients ck are calculated as:

$$C_{k=1/N} \sum_{n=0}^{N-1} x(n) e^{-jk(2\pi/N)n} \qquad (4)$$

The coefficients ck of Fourier series are the form of a periodic sequence of fundamental period N. The time domain spectrum of a periodic signal can be represented as periodic sequence with DFT. The frequency analysis of discrete-time periodic signals ($sin(nt)$ and $cos(nt)$) involves Fourier transform of the time-domain signal. DFT is defined as multiplication of N samples $x(n)$ with N discrete frequencies. These samples are taken at discrete frequencies ($\omega k=2\pi k/N$), where k=0,1...N-1, between $0 \le \omega \le 2\pi$. This means that $X(\omega)$ is evaluated at the successive samples by equally spaced frequencies.X(ω) is given in the following.

$$X(\omega_k) = \sum_{n=-\infty}^{N=\infty} x(n) e^{-j(2\pi/N)n} \qquad (5)$$

DFT is a mapping between N samples $x(n)$ of the time domain into N samples $X(\omega)$ of the frequency-domain. This gives the opportunity to compute DFT of the periodic and the finite-length signals. The frequency-domain spectrum of a periodic sequence can be re-obtained as the periodic signal by using inverse discrete-time Fourier transform (IDFT). IDFT can be defined by using the frequency samples of $X(k)$. It is given in the following.

$$x(n)_{=1/N} \sum_{k=0}^{N-1} X(k)\, W_N^{-kn} \quad ,n=0,1,...N-1 \qquad (6)$$

IDFT shows that there is no loss information by transforming the frequency spectrum of $X(k)$ back into the original time sequence of $x(n)$.

The DFT has seen wide usage across a large number of fields; we only sketch a few examples below (see also the references at the end). All applications of the DFT depend crucially on the availability of a fast algorithm to compute discrete Fourier transforms and their inverses, a fast Fourier transform.

Due to increased employability of FFT inv modern electronic systems, higher radix FFTs such as radix –4, radix – 8, radix – 2^K, split radix, etc. are designed for improved timing and reduced hardware. The basic difference of the mentioned methods lies in the structure of their butterfly units.

II.NEDA

In many digital signal processing (DSP) system,Distributed Arithmetic (DA) is used to implement multiply and accumulate (MAC) unit.DA eliminates the need of a multiplier that is used as a part of MAC unit.It implements MAC unit by pre-computing all possible products and by storing them using a read only memory (ROM). If one set of the

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

inputs has a fixed value,ROM can be eliminated. This can be done by distributing the coefficients to the inputs of the unit. This approach is called NEw Distributed Arithmetic (NEDA). Thus, using NEDA, any MAC like unit can be implemented just by using adders and shifters. Architecture designs in use either DA approach or CORDIC unit approach to implement FFT, which require ROM as en essential unit in the design. The proposed approach is based on NEDA, which does not require any ROM thus making the design to have reduced hardware. The distribution of the coefficients is done optimally to further reduce the redundant hardware units. New Distributed Arithmetic (NEDA) technique is being used in many digital signal processing systems that require MAC unit. Transforms such as FFT, DCT, etc. have many multiplications that in turn require a number of multipliers. Implementation of such transforms using NEDA improves performance of the system in terms of speed, power and area. The mathematical derivation of NEDA is discussed as follows. Inner product calculation of two sequences may be represented as

$$Z = \sum_{i=1}^{k} C_i X_i \tag{7}$$

Where Ci are constant coefficients and Xi are varying inputs. Matrix representation of equation (7) may be given as

$$Z = \begin{bmatrix} C_1 & C_2 & \cdots & C_k \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_k \end{bmatrix} \tag{8}$$

Considering both Ci and Xi in 2's complement format, they may be expressed in the form

$$C_i = -C_i^M 2^M + \sum_{k=N}^{M-1} C_i^k 2^k \tag{9}$$

Where Ci= 0 or 1, k=N,N+1, … , M and Ci ^{M} is the sign bit and Ci ^{N} is the least significant bit. Substituting equation (9) in equation (8) results in the following matrix product which is modelled according to the required design.

$$Z = \begin{bmatrix} -2^0 & 2^{-1} & \cdots & 2^{-12} \end{bmatrix} \begin{bmatrix} C_1^0 & \cdots & C_k^0 \\ \vdots & \ddots & \vdots \\ C_1^{12} & \cdots & C_k^{12} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_k \end{bmatrix} \tag{10}$$

The matrix containing Ci^{k} is a sparse matrix, which means the values are either 0 or 1. The number of rows in C matrix defines the precision of fixed coefficients. Equation (10) is rearranged as shown below.

$$Z = \begin{bmatrix} -2^0 & 2^{-1} & \cdots & 2^{-12} \end{bmatrix} \begin{bmatrix} W_0 \\ W_1 \\ \vdots \\ W_{12} \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} W_0 \\ W_1 \\ \vdots \\ W_{12} \end{bmatrix} = \begin{bmatrix} C_1^0 & \cdots & C_k^0 \\ \vdots & \ddots & \vdots \\ C_1^{12} & \cdots & C_k^{12} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_k \end{bmatrix} \tag{12}$$

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

The W matrix consists of sums of the inputs depending on the coefficient values, in each row. An example that shows the NEDA operations is discussed below. Consider to evaluate the value of equation (12).

$$Y = \begin{bmatrix} \cos\frac{\pi}{8} & \cos\frac{\pi}{4} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{13}$$

Equation (13) can be expressed in the form of equation (10) as shown in equation (14).

$$Y = \begin{bmatrix} -2^0 & 2^{-1} & \cdots & 2^{-12} \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{14}$$

Equation (14) may be rewritten as

$$Y = \begin{bmatrix} -2^0 & 2^{-1} & \cdots & 2^{-12} \end{bmatrix} \begin{bmatrix} 0 \\ X_1 + X_2 \\ X_1 \\ X_1 + X_2 \\ X_2 \\ X_1 \\ X_1 + X_2 \\ 0 \\ X_2 \\ X_1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{15}$$

Applying precise shifting, we rewrite equation (15) as

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

$$Y = [2^{-1} \quad 2^{-2} \quad 2^{-3} \quad 2^{-4} \quad 2^{-5} \quad 2^{-6} \quad 2^{-8} \quad 2^{-9}] \begin{bmatrix} X_1 + X_2 \\ X_1 \\ X_1 + X_2 \\ X_2 \\ X_1 \\ X_1 + X_2 \\ X_2 \\ X_1 \end{bmatrix} \tag{16}$$

Thus implementing equation (16) further reduces number of adders compared to implement equation (15). Multiplication with 2^i, i-belongs to Z^+ can be realized with the help of shifters. In equation (16), the first row of X matrix shifts right by 1 bit, second row by 2 bits and so on. More precisely, the shifts carried out are arithmetic right shifts. The output Y can be realized as a column matrix if we need the partial products. Thus NEDA based architecture designs have less critical path compared to traditional MAC units.

III.PROPOSED RADIX-4 ALGORITHM

In many applications of digital signal processing, the Discrete Fourier Transform (DFT) plays an important role.DFT has been applied in a wide range of fields such as linear filtering, spectrum analysis, digital video broadcasting and orthogonal frequency demodulation multiplexing (OFDM). The rapidly increasing demand of OFDM based applications, including modern wireless telecommunication such as LAN, needs real-time high speed computation in Fast Fourier Transform algorithm. This has made the design of FFT processor a critical requirement for the up coming wireless technology.

With the advent of this requirement, the study of high performance VLSI FFT architecture is likewise of increasing importance. Many different hardware architectures have been proposed for the implementation of FFT algorithms. The main concern of the design approach will be power and architectural size. Among various FFT algorithms, radix-2 FFT with Cooley-Turkey algorithm, is very popular because it makes efficient use of symmetry . Several architectures have been proposed based on Cooley- Turkey algorithm to further reduce the computation complexity, including radix-4, radix-2, and split-radix. Basically, this Fast Fourier Transform algorithm use Divide-and-Conquer approach to divide the computation recursively and then extract as many common twiddle factors as possible. The number of required real additions and multiplications is usually used to compare the efficiency of different FFT algorithms. In terms of the multiplicative comparison, the split-radix FFT is computationally better to all the other algorithms because it has most trivial multiplications. Eventually, this algorithm has a drawback because of irregular structure that leads this algorithm not suitable for implementation on digital signal processors. Structural regularity is also important in implementation of FFT algorithms on dedicated chips such as in ASIC (Application Specific Integrated Chip). Hence, radix-2 and radix-4 FFT algorithm are preferable in terms of speed and accuracy.

In this paper, we have proposed the implementation of 16- point complex FFT using radix-4 method. Complex multiplications required during the process have been implemented by using NEDA. According to the radix-4 algorithm, to implement 16-point FFT, eight radix-4butterflies are required. Four radix-4 butterflies are used in the first stage and the other four being used in the second/final stage. The input is taken in normal order and the output in bit-reversal order.

The output of each radix-4 butterfly is multiplied by the respective twiddle factors. In the shown block diagram, the first stage consists of four radix-4 butterflies. The inputs to the butterflies are s(n), s(n+4), s(n+8), s(n+12) where n is 0 for first butterfly, 1 for second butterfly, 2 for the third butterfly, and 3 for the last butterfly, all of first stage. The twiddle factors are given by W_{16}^{0}, W_{16}^{q}, W_{16}^{2q},W_{16}^{3q} where *q* is 0 for first butterfly, 1 for second butterfly, 2 for third butterfly, and 3 for the last butterfly, all of first stage. The outputs of first stage are multiplied with respective twiddle

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

factors and are given as inputs to the second stage.As proposed in our design, the complex twiddle multiplications required at the stage-1 output have been implemented by using NEDA blocks..

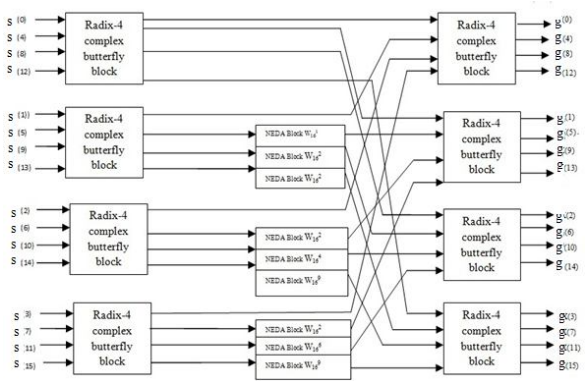

Fig. 1. Block diagram of the proposed architecture

NEDA blocks are required at the output of first stage of the 16 point FFT processor. In the second stage, 4 more radix-4 butterfly blocks are used. The first radix-4 butterfly in the second stage takes the first output of the 4 radix-4 butterfly blocks used in the first stage. The second radix-4 butterfly in the second stage takes the second output of the 4 radix-4 butterfly blocks followed by the NEDA block (if required). This process continues for the rest radix-4 butterfly blocks present in the second stage. There is no need of using any NEDA block after second stage as the twiddle factor W_{16}^{0} that is 1 is multiplied to the outputs of the second stage. The final output comes in a bit-reversal order. The advantage of using radix-4 algorithm is that it retains the simplicity of radix-2 algorithm and gives the output with lesser complexity. The NEDA block shown in the proposed block diagram does the complex multiplication of the output of the first stage and the respective twiddle factor. The twiddle factor values used here are as follows.

$$
\begin{aligned}
W_{16}^{1} &= \cos \pi/8 - j \sin \pi/8 = 0.9238 - j0.3826 \\
W_{16}^{2} &= \cos \pi/4 - j \sin \pi/4 = 0.7071 - j0.7071 \\
W_{16}^{3} &= \cos 3\pi/8 - j \sin 3\pi/8 = 0.3826 - j0.9238 \\
W_{16}^{4} &= \cos 4\pi/8 - j \sin 4\pi/8 = 0 - j \\
W_{16}^{6} &= \cos 3\pi/4 - j \sin 3\pi/4 = -0.7071 - j0.7071 \\
W_{16}^{9} &= \cos 9\pi/8 - j \sin 9\pi/8 = 0.9238 - j0.3826
\end{aligned}
\qquad (18)
$$

The basic structure of the radix 4 butterfly is shown in figure 2.It has four inputs and four outputs.

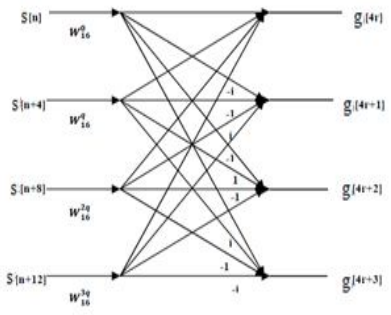

Fig. 2. Radix-4 butterfly structure

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

IV.NEDA USING CARRY SELECT ADDER AND MODIFIED CARRY SELECT ADDER

In the NEDA section carry select adder is used for addition.Design of area- and power-efficient high-speed data path logic systems are one of the most substantial areas of research in VLSI system design. In digital adders, the speed of addition is limited by the time required to propagate a carry through the adder. The sum for each bit position in an elementary adder is generated sequentially only after the previous bit position has been summed and a carry propagated into the next position.

The CSLA is used in many computational systems to alleviate the problem of carry propagation delay by independently generating multiple carries and then select a carry to generate the sum . However, the CSLA is not area efficient because it uses multiple pairs of Ripple Carry Adders (RCA) to generate partial sum and carry by considering carry input Cin=0 and Cin=1, then the final sum and carry are selected by the multiplexers (mux).

The structure of the 16-b regular CSLA is shown in Fig. 3. It has five groups of different size RCA.

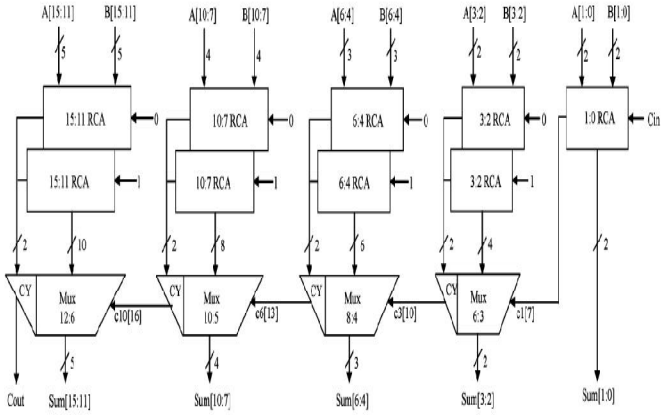

Fig.3 Structure of 16-b CLSA

Binary to Excess-1 Converter(BEC) is used instead of RCA with Cin=1 in the regular CSLA to achieve lower area and power consumption. The main advantage of this BEC logic comes from the lesser number of logic gates than the n-bit Full Adder (FA) structure.

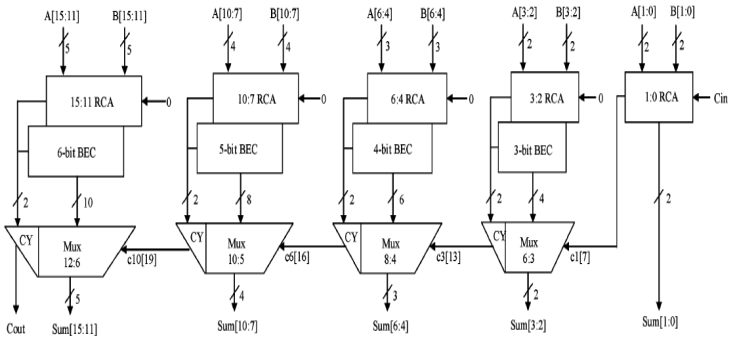

Fig..4 .Modified 16-b CSLA

The structure of the proposed 16-b CSLA using BEC for RCA with Cin=1 to optimize the area and power is shown in Fig. 4

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

V. RESULT AND DISCUSSION

The synthesis report obtained from Xilinx ISE 13.2 for different Virtex FPGAs are shown in following tables.Table.I gives the device utilisation summary obtained for Virtex 6 FPGA for FFT with NEDA using CSLA and FFT with NEDA using Modified CSLA. Device utilization is less for NEDA using modified CSLA.

Table.I Device utilisation summary obtained for Virtex 6

	USED		AVAILABLE	UTILISATION	
	FFT with NEDA using CSLA	FFT with NEDA using Modified CSLA		FFT with NEDA using CSLA	FFT with NEDA using Modified CSLA
No. of slice LUTs	6471	4364	474,240	1%	1%
No. of slice registers	5192	2808	948,480	1%	1%
No. used as logic	6439	4332	474,240	1%	1%
No. of occupied slices	2924	1751	118,560	2%	1%

Table.II gives the power and delay obtained for Virtex 6 FPGA for FFT with NEDA using CSLA and FFT with NEDA using Modified CSLA.NEDA using modified CSLA uses less power as compared to NEDA using CSLA.

Table.II Power and delay obtained for Virtex 6

	FFT with NEDA using normal CSLA	FFT with NEDA using modified CSLA
TOTAL POWER(W)	4.792	4.462
DELAY(ns)	7.022	9.612

Table.III gives the device utilisation summary obtained for Virtex 5 FPGA for FFT with NEDA using CSLA and FFT with NEDA using Modified CSLA.

Table.III Device utilisation summary obtained for Virtex 5

	USED		AVAILABLE	UTILISATION	
	FFT with NEDA using CSLA	FFT with NEDA using Modified CSLA		FFT with NEDA using CSLA	FFT with NEDA using Modified CSLA
No. of slice LUTs	8244	6226	207,360	3%	3%
No. of slice registers	4400	2016	207,360	1%	1%
No. of occupied slices	6439	4332	474,240	1%	1%
No. used as logic	8204	6186	207,360	3%	2%

Table.IV gives the power and delay obtained for Virtex 5 FPGA for FFT with NEDA using CSLA and FFT with NEDA using Modified CSLA.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

Table. IV Power and delay obtained for Virtex 5

	FFT with NEDA using normal CSLA	FFT with NEDA using modified CSLA
TOTAL POWER(W)	4.21	4.195
DELAY(ns)	9.524	16.399

Table.V gives the device utilisation summary obtained for Virtex 4 FPGA for FFT with NEDA using CSLA and FFT with NEDA using Modified CSLA.

Table .V Device utilisation summary obtained for Virtex 4

	USED		AVAILABLE	UTILISATION	
	FFT with NEDA using CSLA	FFT with NEDA using Modified CSLA		FFT with NEDA using CSLA	FFT with NEDA using Modified CSLA
No. of 4 input LUTs	9199	6622	178,176	5%	3%
No. of slice registers	4400	2016	178,176	2%	1%
No. of bonded IOBs	4988	3659	89088	5%	4%
No. of occupied slices	897	897	960	93%	93%

Table.VI gives the power and delay obtained for Virtex 4 FPGA for FFT with NEDA using CSLA and FFT with NEDA using Modified CSLA.

Table.VI Power and delay obtained for Virtex 4

	FFT with NEDA using normal CSLA	FFT with NEDA using modified CSLA
TOTAL POWER(W)	2.911	2.768
DELAY(ns)	13.023	20.693

VI.CONCLUSION

Radix 4 complex 16 point FFT core using NEDA with modified CSLA is designed. It is a ROM-less and multiplier-less method. The proposed design is efficient in terms of hardware and power consumption. Fast Fourier transform (FFT) is an efficient algorithm to compute the discrete Fourier transform and it's inverse. A DFT decomposes a sequence of values to the components of different frequencies. This operation is useful in many fields but computing it directly from the definition is often too slow to be practical.

Using NEDA, any MAC like unit can be implemented just by using adders and shifters. Existent architecture designs use either DA approach or CORDIC unit approach to implement FFT, which require ROM as en essential unit in the design. The proposed approach is based on NEDA, which does not require any ROM, thus making the design to have reduced hardware. The distribution of the coefficients is done optimally to further reduce the redundant hardware units, thus reducing the hardware. The distribution of the coefficients is done optimally to further reduce the redundant hardware units. NEw Distributed Arithmetic (NEDA) technique is being used in many digital signal processing systems that require MAC unit. Transforms such as FFT, DCT, etc. have many multiplications that in turn require a number of multipliers. Implementation of such transforms using NEDA improves performance of the system in terms of speed, power and area.

ISSN (Print) : 2320 – 3765
ISSN (Online): 2278 – 8875

International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 10, October 2014

The proposed design is simulated by using ModelSim and synthesised by Xilinx ISE project navigator. The synthesis results show that the computation for calculating the 16 point FFT is efficient in terms of area and power using the proposed method.

REFERENCES

[1] C. González-Concejero, V. Rodellar,“ A portable hardware design of a FFT algorithm”, Latin American Applied Research,37:78-82, 2007.
[2] Asmitha Haveliya, Amity University Lucknow, India “Design And Simulation Of 32- Point FFT Using Radix-2 Algorithm For FPGA Implementation” Second International Conference on Advanced Computing & Communication Technologies, 2012
[3] B. Ramkumar and Harish M Kittur,” Low-Power and Area-Efficient Carry Select Adder”, IEEE Transactions on Very Large Scale Integration (VLSI) Systems,VOL. 2, NO. 2,PP. 371-375, Feb.2012.
[4] Anumol B. Chennattucherry, Diego James, “ A Novel Approach to Reduce Area and Power for FFT Implementation”, Proc. Intl. Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering,vol.2,pp 130-138.
[5] Alban Ferizi,Bernhard Hoeher, Melanie Jung, Georg Fischer, and Alexander Koelpin, “Design and Implementation of a Fixed-Point Radix- 4 FFT Optimized for Local Positioning in Wireless Sensor Networks,” Intl. Multi-Conf. Syst.Signals and Devices, pp. 1 – 4, Mar. 2012.
[6] Li Wenqi, Wang Xuan, and Sun Xiangran, “Design of Fixed-Point High- Performance FFT Processor,” Intl. Conf. Edu. Tech. and Comput., vol. 5, pp. 139 – 143, Jun. 2010.
[7] M. Hasan, T. Arshan, and J.S. Thomson, “A novel coefficient ordering based low power pipelined radix-4 FFT processor for wireless LAN applications”, IEEE Trans. of Consumer Electronics: 128-134,2003.
[8] Stanley A.White, “Applications of Distributed Arithmetic to Digital Signal processing:A Tutorial Review,” IEEE ASSP Magazine, vol. 6,no. 3, pp. 4 – 19.