FPGA IMPLEMENTATION OF 3/6 SRFFT ALGORITHM FOR LENGTH 6*m DFTs

M. Sathya[1], M.B.Annadurai[2]

PG Scholar of EEE Department
V.R.S College of Engineering & Technology, Arasur, villupuram(D.T), Tamil nadu, India.
[1]sathyamanickam85@gmail.com
Assistant Professor of EEE Department
V.R.S College of Engineering & Technology, Arasur, villupuram(D.T), Tamil nadu, India.
[2]bagath1986@gmail.com

***Abstract*— The Fast Fourier Transform (FFT) requires high Computational power, ability to choose the algorithm and architecture to implement it. This project explains the realization of a 3/6 FFT processor based on a pipeline architecture. The implementation has been made on a Field Programmable Gate Array (FPGA) as a way of obtaining high performance at economical price and a short time of realization. FPGA can be used with segmented arithmetic of any level of pipeline in order to speed up the operating frequency. The processor has been simulated up to 200 MHz, with an Xilinx Spartan 3E as a target device, for a transform length of 6 complex points. To combine the higher parallelism of the 6-FFTs and the possibility of processing sequences having length of any power of 6.The simultaneous operation of multipliers and adder-subtracters implicit in the 3/6 FFT, leads to faster operation at the same degree of pipeline. The 3/6 FFT algorithm is implemented in Xilinx FPGA Spartan 3E.**

***Keywords*—** Fast Fourier transform (FFT), Field Programmable Gate Array (FPGA), 3/6 FFT.

I. Introduction

Discrete Fourier transform (DFT) is one of the most important tools used in almost all fields of science and engineering, DFT can be implemented with efficient algorithms generally classified as fast Fourier transforms (FFT). The most widely used approaches are so-called the algorithms for 6^m , such as radix-3, radix-6 and split radix FFT (SRFFT). Considerable researches have carried out and resulted in the rapid development on this class of algorithms. The algorithm decompose the DFT into one length-N/3 and four length-N/6 sub-DFT . The flexibility of the decomposition enables the algorithm be competent at the implementation of a non-power-of-six DFT, while its length can exactly divided by 6. Appropriate permutations are used for sub-DFTs input sequences to reduce the computational intension.The rest of the brief is organized as follows. Section II proposed system introduction III compute the DFT using proposed algorithm and compares, analyzes the performance of the proposed algorithm with existing algorithm. Section IV System implementation Section V Execution output and Section VI conclusions and the future works.

II. The Proposed Radix 3/6 Algorithm

Let us recall the definition of DFT:

$$X_k = \sum_{n=0}^{N-1} x_n w_N^{nk} \qquad (1)$$

Where $w_N=e^{-j2\pi/N}$,j=-1 , the length N of sequence x(n) is assumed as an integer, which is divisibly by six. For lengths N of DFT, powers-of-six would be best for the proposed algorithm. Obviously, the DFT can be divided into three length N/3 sub-DFTs. In order to derive a best possible algorithm, we continue to decompose the three sub-DFTs. Due to no scaling factor in front of it, the first sub-DFT should be let as it is and directly go into the recursive decomposition of the next stage. The other two sub DFTs are divided into four sub-DFTs of length-N/6 .

Actually, if the length of a DFT can be divided by 6, the DFT can be decomposed by the algorithm. The generalized length- N can be assumed as $N=2^r x3^m$, where r≥m-1. The decomposition of a DFT of size $N=2^r x3^m$ is denoted by,

$$X_k = \sum_{n=0}^{N/3-1} x_{3^n} w_{N/3}^{nk} + w_{2^r}^k w_{3^m}^k \sum_{n=0}^{N/6-1} x_{6^n+2^r+3^m} w_{N/6}^{nk} + w_{3^m}^k \sum_{n=0}^{N/6-1} x_{6^n+2^r} w_{N/6}^{nk}$$
$$+ w_{3^m}^k \sum_{n=0}^{N/6-1} x_{6^n-2^r} w_{N/6}^{nk} + w_{2^r}^{-k} w_{3^m}^{-k} \sum_{n=0}^{N/6-1} x_{6^n-2^r-3^m} w_{N/6}^{nk} \qquad (2)$$

Where the four length- N/6 sub DFTs are reordered. To simplify the description, (1) can be expressed by,

$$X_k = A_k + w_{2^r}^k w_{3^m}^k B_k + w_{3^m}^k C_k + w_{3^m}^{-k} E_k + w_{2^r}^{-k} w_{3^m}^{-k} F_k$$

Where,

$$A_k = \sum_{n=0}^{N/3-1} x_{3n} w_{N/3}^{nk}$$

$$B_k = \sum_{n=0}^{N/6-1} x_{6n+2r+3m} w_{N/6}^{nk}$$

$$C_k = \sum_{n=0}^{N/6-1} x_{6n+2r} w_{N/6}^{nk}$$

$$E_k = \sum_{n=0}^{N/6-1} x_{6n-2r} w_{N/6}^{nk}$$

$$F_k = \sum_{n=0}^{N/6-1} x_{6n-2r-3m} w_{N/6}^{nk} \qquad (3)$$

In (3), $w_{2^r}^{k} w_{3^m}^{k} B_k$ and $w_{2^r}^{-k} w_{3^m}^{-k} F_k$ can be treated in pairs, since $w_{2^r}^{k} w_{3^m}^{k}$ and $w_{2^r}^{-k} w_{3^m}^{-k}$ is a conjugate-pair. In the similar way, $w_{3^m}^{k}$ can be handled with in pairs. The direct implementation performs many unnecessary operations, computations of X_k, $X_{\frac{N}{6}+k}$, $X_{\frac{2N}{6}+k}$, $X_{\frac{3N}{6}+k}$, $X_{\frac{4N}{6}+k}$, $X_{\frac{5N}{6}+k}$ turn out to share many calculations each other. In particular, if we add to, the size-N/6 to k DFT are not changed (because they are periodic in k), while the size-N/3 DFT is unchanged if we add to 2N/6 to k. So, the only things that changes are the, $w_{2^r}^{-k} w_{3^m}^{-k}$ *and* $w_{3^m}^{-k}$ terms. In order to reduce the number of the operations, the following six identities are necessary,

$$X_k = A_k + (w_{2^r}^{k} w_{3^m}^{k} B_k + w_{2^r}^{k} w_{3^m}^{k} F_k) + (w_{3^m}^{k} C_k + w_{3^m}^{-k} E_k) \qquad (13)$$

$$X_{2N/6+k} = A_k + (w_3^{2^r} w_{2^r}^{k} w_{3^m}^{k} B_k + w_3^{-2^r} w_{2^r}^{-k} w_{3^m}^{-k} F_k) + (w_3^{2^r} w_{3^m}^{k} C_k + w_3^{-2^r} w_{3^m}^{-k} E_k) \qquad (14)$$

$$X_{4N/6+k} = A_k + (w_3^{2^{r+1}} w_{2^r}^{k} w_{3^m}^{k} B_k + w_3^{-2^{r+1}} w_{2^r}^{-k} w_{3^m}^{-k} F_k) + (w_3^{2^{r+1}} w_{3^m}^{k} C_k + w_3^{-2^{r+1}} w_{3^m}^{-k} E_k) \qquad (15)$$

$$X_{N/6+k} = A_{N/6+k} - (w_3^{2^{r-1}} w_{2^r}^{k} w_{3^m}^{k} B_k + w_3^{-2^{r-1}} w_{2^r}^{-k} w_{3^m}^{-k} F_k) + (w_3^{2^{r-1}} w_{3^m}^{k} C_k + w_3^{-2^{r-1}} w_{3^m}^{-k} E_k) \qquad (16)$$

$$X_{3N/6+k} = A_{N/6+k} - (w_{2^r}^{k} w_{3^m}^{k} B_k + w_{2^r}^{-k} w_{3^m}^{-k} F_k) + (w_{3^m}^{k} C_k + w_{3^m}^{-k} E_k) \qquad (17)$$

$$X_{5N/6+k} = A_{N/6+k} - (w_3^{2^r} w_{2^r}^{k} w_{3^m}^{k} B_k + w_3^{-2^r} w_{2^r}^{-k} w_{3^m}^{-k} F_k) + (w_3^{2^r} w_{3^m}^{k} C_k + w_3^{-2^r} w_{3^m}^{-k} E_k) \qquad (18)$$

A complete output set $\{X_k\}$ can be obtained if we let range from 0 to N/6 -1in the above six equations. We now summarize the scheme of the proposed radix-3/6 FFT algorithm. The initial input sequence of length- is decomposed into five sub-sequences. This process is repeated successively for each of new sub-sequences, until the sizes of all subDFTs are indivisible by 6. Figs. 1,2 illustrate the flow graph of 3 and 6point radix 3/6 algorithm (2-points and 4-points FFT can be performed with SRFFT).

In this section, we consider the performance of the proposed algorithm by analyzing the computational complexity and comparing it with existing algorithms. Let M_N and A_N be, respectively the number of multiplications and additions. We assume that a 3-point DFT requires 4 real multiplications and 12 real additions (some algorithm assumes that a 3-point DFT is calculated with 2 real multiplication and 12 real additions, since one need not multiply ½ and the multiplication by 1/2 can be evaluated with bit shift).

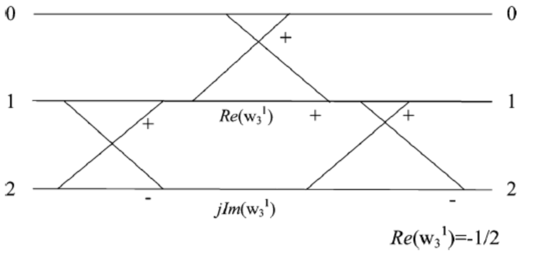


Fig. 1 Flow graph of 3-point FFT

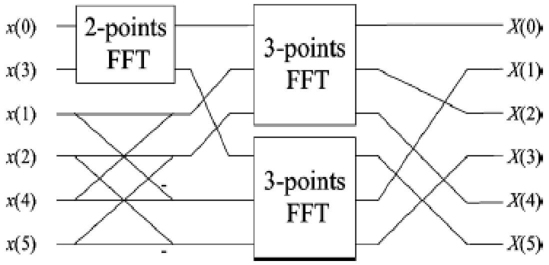


Fig. 2 Flow graph of 6-point 3/6 FFT

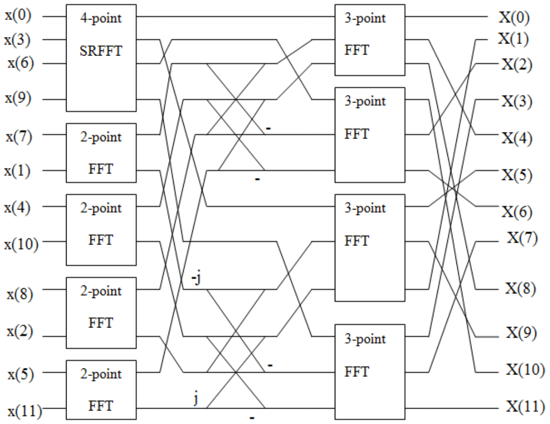


Fig. 3 Flow graph of 12-point 3/6 FFT

III. Performance Analysis

The general butterfly of the proposed algorithm requires 16 real multiplications and 40 real additions. In general butterfly we evaluate (4) with 8 real multiplication and 16 real additions. Because $w_{2^m}^{k} w_{3^m}^{k} = w_{2^m}^{-k*} w_{3^m}^{-k*}$ and $w_{3^m}^{2k} = w_{3^m}^{-2k*}$ .We calculate (14) with 8 real multiplications and 8 real additions because we share real additions with which have been undertaken in evaluating (4). We evaluate (6) with only 4 real additions, because 1+u+u*=0. Furthermore, we perform (7)–(8) at cost of 12 real additions, because all multiplications and some additions have been calculated in (4)–(6).

There are six special cases. The first special case, when k=0 requires 8 real multiplications and 32 real additions. In this case,(4) is evaluated with 8 real additions (one need not multiply 1), (5) is implemented with 4 real multiplications and 6 real additions because we use real additions which have been undertaken in evaluating above calculation, (6) can be calculated with only 2 real additions, because we need not add the duplicate portion between u and u*. In the same way, (4)–(5) can be performed by only 4 real multiplications and 16 real additions. This special butterfly is illustrated in Fig. 2. The second special case,when k=2^{r-2} x 3 $^{m-1}$, requires the number of operations equals that of the first case. In this case, all rotator factors of subDFTs in (1) can be omitted, so it can be evaluated with 8 real additions, (4) can be implemented with 4 real multiplications and 6 real additions,(4) can be calculated with only 2 real additions. Similarly, (6)–(7) can be performed by only 4 real multiplications and 16 real additions.

The decomposition in the proposed algorithm is conducted recursively until the lengths of all sub DFTs cannot be exactly divided by 6. In general, there are only 1 the first special butterfly (if r≥1and m≥1), 1 the second special case butterfly (if r≥2and m≥1), 1 the third special case butterfly and 1 the fourth special case butterfly (if r≥3and m≥1). The total number of the fifth and sixth type of butterflies is 2^{r-1} -4. In additions, there are 2^{r-1} (3^{m-1}- 1)general butterfly. Thus, the arithmetic complexity of the proposed algorithm can be given as follows,

$$M_N = \begin{cases} \dfrac{M_{N/3} + 4M_{N/6} + 8N}{3-8}, r = 1, m \geq 1 \\ \dfrac{M_{N/3} + 4M_{N/6} + 8N}{3-16}, r = 2m \geq 1 \\ \dfrac{M_{N/3} + 4M_{N/6} + 8N}{3-24}, r \geq 3, m \geq 1 \end{cases} \quad (8)$$

$$A_N = \begin{cases} \dfrac{A_{N/3} + 4A_{N/6} + 20N}{3-8}, r = 1, m \geq 1 \\ \dfrac{A_{N/3} + 4A_{N/6} + 20N}{3-16}, r = 2m \geq 1 \\ \dfrac{A_{N/3} + 4A_{N/6} + 20N}{3-2^{r+1}-8}, r \geq 3, m \geq 1 \end{cases} \quad (9)$$

Equation (8) and (9) gives the number of multiplications and additions required by the proposed radix 3/6 algorithm. It is seen that the complexity of this algorithm is less than that required by other algorithms.

The third special case is when k=2^{r-3}x3^m . This butterfly requires 12 real multiplications and 36 real additions. In this case, (13) requires extra 4 real multiplication and 4 real additions over the first case. The computations of the rest equations are similar with that of the second case.The fifth special case is when k mod 3^m and k mod 2^{r-3}≠0. This butterfly requires 16 real multiplications and 36 real additions. In this case, (13) requires extra 8 real multiplication and 4 real additions over the first case. The sixth special case is when ,k mod 3^{m-1}=0,k mod 3^m≠0 and k mod 2^{r-3} . This butterfly requires 16 real multiplications and 36 real additions. In this case, (7) requires extra 8 real multiplication and 4 real additions over the second case.

IV. System Implementation

Implementation Of 3/6 SRFFT Algorithm in VLSI Technology use the Spartan 3E.It provides a powerful and highly advanced self-contained development platform for designs targeting the Spartan 3E FPGA from Xilinx. It features a 500K gate Spartan 3E FPGA with a 32 bit RISC processor and DDR interfaces. The board also features a Xilinx Platform Flash, USB and JTAG parallel programming interfaces with numerous FPGA configuration options via the onboard Intel StrataFlash and ST Microelectronics Serial Flash. The board is fully compatible with all versions of the Xilinx ISE tools including the free Web Pack. The board ships with a power supply and USB cable for programming so designs can be implemented immediately with no hidden costs. The Spartan 3E Starter board is also compatible with the Micro Blaze Embedded Development Kit (EDK) and Pico Blaze from Xilinx. Implementing the 12 point 3/6 SRFFT algorithm in Xilinx FPGA and check the corresponding output in the 2 No's of 20x4 LCD display and checking the corresponding algorithm over area consumption in FPGA. Implementation of 3/6 SRFFT algorithm reduce the area consumption and use less number of LUT.

Xilink Spartan -3E FPGA Board

The Xilinx Spartan 3E FPGA board is a robust board containing many features. A list of key features and their location on the board is listed below, and all of these features are explained in great detail in the manual provided with the FPGA.

- 9-pin RS-232 Serial Port
- JTAG port for low-cost download cable
- JTAG download/debug port compatible with the Xilinx Parallel Cable IV and
- AC power adapter input for included international unregulated 9Vac power supply
- 1Mbit Xilinx XCF01S Platform Flash, in-system programmable configuration PROM, 9-pin RS-232 Serial Port, JTAG port for low-cost download cable

- 100,000-gate Xilinx Spartan-3E XC3S100E FPGA in a 100-very thin quad flat package (XC3S100VQG100C

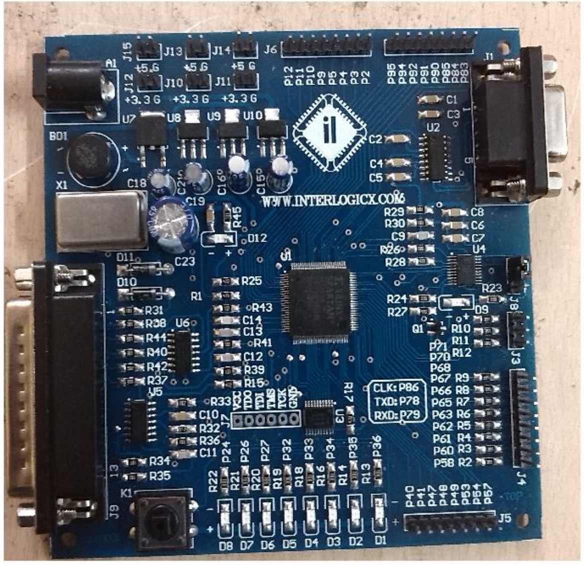


Fig. 4 Xilinx Spartan-3E FPGA Kit Block Diagram

- AC power adapter input for included international unregulated 9Vac power supply
- Power-on indicator LED, On-board 3.3V , 2.5V , and 1.2V regulators

V. HARDWARE RESULT

The output of the project is shown with FPGA kit and with the help of 20*4 the output is displayed in the binary where in real and imaginary terms the execution can be done in the performance of the input data's given in the programs.

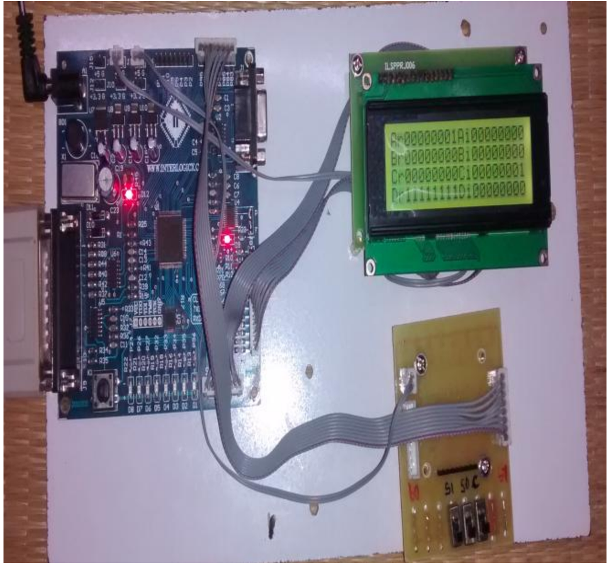
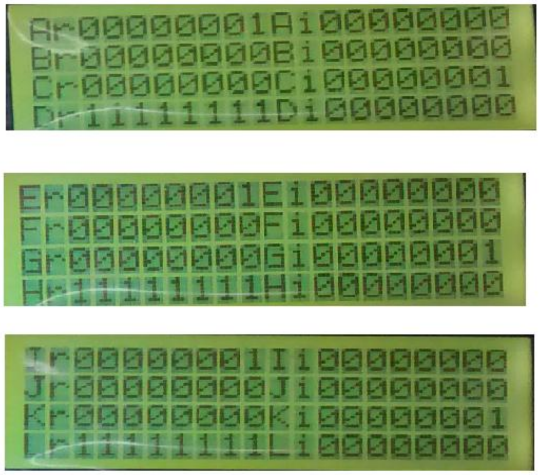


Fig. 5 Output FPGA kit display

LCD display output

The display is the LCD display of 20*4, where in the binary data. 12 point output is implemented with three conditions.

1. When both input are low(S0=S1=0), LCD display the first four point output(A,B,C,D)
2. When S0 = 0 and S1=1, LCD display the next four point output(E,F,G,H).
3. When S0 = 1 and S1=1, LCD display the next four point output(I,J,K,L).



```
Ar00000001Ai00000000
Br00000000Bi00000000
Cr00000000Ci00000001
Dr11111111Di00000000
```

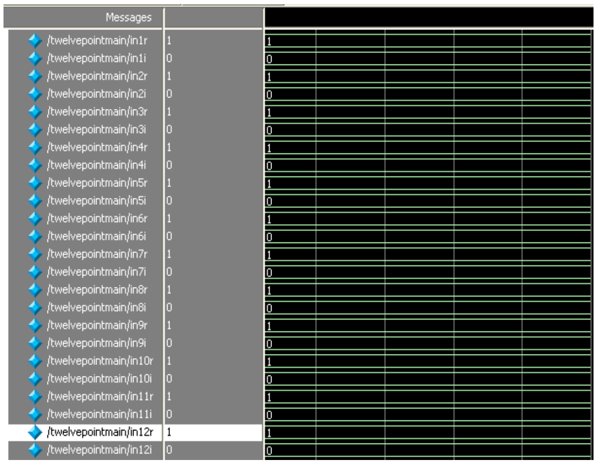
```
Er00000001Ei00000000
Fr00000000Fi00000000
Gr00000000Gi00000001
Hr11111111Hi00000000
```

```
Ir00000001Ii00000000
Jr00000000Ji00000000
Kr00000000Ki00000001
Lr11111111Li00000000
```

Fig. 6 LCD Display Output

VI. SIMULATION OUTPUT

The 12 point DFT sequence has been implemented in VLSI and simulated using modelsim based on radix 3/6 FFT algorithm.



```
Messages
/twelvepointmain/in1r   1   1
/twelvepointmain/in1i   0   0
/twelvepointmain/in2r   1   1
/twelvepointmain/in2i   0   0
/twelvepointmain/in3r   1   1
/twelvepointmain/in3i   0   0
/twelvepointmain/in4r   1   1
/twelvepointmain/in4i   0   0
/twelvepointmain/in5r   1   1
/twelvepointmain/in5i   0   0
/twelvepointmain/in6r   1   1
/twelvepointmain/in6i   0   0
/twelvepointmain/in7r   1   1
/twelvepointmain/in7i   0   0
/twelvepointmain/in8r   1   1
/twelvepointmain/in8i   0   0
/twelvepointmain/in9r   1   1
/twelvepointmain/in9i   0   0
/twelvepointmain/in10r  1   1
/twelvepointmain/in10i  0   0
/twelvepointmain/in11r  1   1
/twelvepointmain/in11i  0   0
/twelvepointmain/in12r  1   1
/twelvepointmain/in12i  0   0
```

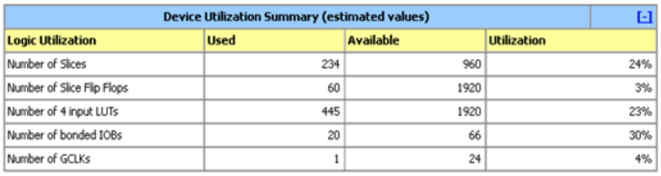
Fig 7 Simulation screenshot 1

The output is checked using the 12 point radix 3/6 flow graph theoretically and it matches with the simulated results. Fig. 7 shows the simulation results of 12 point DFT sequence. Table I and Table II shows the device utilization summary of 12 point DFT sequence in Xilinx XSE. Simulating the 12 point SRFFT with twelve point input(include both real and imaginary value).

Area Consumptions

The Area consumed in the radix 3/6 FFT Algorithm are reduced in number of slices and the number of flip flops and number of 4 input LUTs and number of bonded IOBs and number of GCLKs are performed in the area consumed are very less and the area are configured target device are performed in this algorithm.

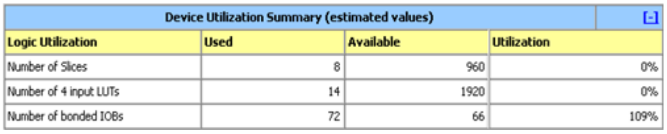
TABLE I
DEVICE UTILIZATION OF NORMAL FFT



Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	234	960	24%
Number of Slice Flip Flops	60	1920	3%
Number of 4 input LUTs	445	1920	23%
Number of bonded IOBs	20	66	30%
Number of GCLKs	1	24	4%

TABLE III
DEVICE UTILIZATION OF SRFFT



Device Utilization Summary (estimated values)

Logic Utilization	Used	Available	Utilization
Number of Slices	8	960	0%
Number of 4 input LUTs	14	1920	0%
Number of bonded IOBs	72	66	109%

VII. CONCLUSION AND FUTURE WORK

The implementation of radix 3/6 FFT algorithm is new type of algorithm where as the cooley tukey algorithm is the oldest algorithm and it takes more delay in transferring the data. Here we can able to transfer the data in very high speed while compared to the performance of the Digital signal processing method. The area consumed in this FPGA is very less compared to the cooley turkey algorithm method. The splices are very less consumed and the data are stored in the PROM memory and therefore the data cannot reduce the memory storage and it saves the output till the next data are stored in the processor data functions. There are two modes of testing the data in binary form one in the form of modelsim software and other with the FPGA Xilinx Spartan 3E kit. The inputs are given in the program and the outputs are checked in the LCD displays. Thus the output is performed for reducing time delay and area consumption and logical verification.

Scope For Future Work

The implementation of radix 3/6 FFT algorithm is new type of algorithm where as the cooley tukey algorithm is the oldest algorithm and it takes more delay in transferring the data. For further representation for this algorithm we can able to implement in ASIC as a hardware chip for direct data.

REFERENCES

[1] S. Bouguezel, M. Ahmad, and M. Swamy, **"A new radix-2/8 FFT algorithm for length- DFTs,"** IEEE Trans. Circuits Syst. I, vol.51, no. 9, pp. 1723–1732, Sep. 2004.

[2] M. Frigo and S. Johnson, **"The design and implementation of fftw3,"**Proc. IEEE, vol. 93, no. 2, pp. 216–231, 2005.

[3] D. Sepiashvili, "**PerformanceModels and SearchMethods for Optimal FFT Implementations,**" M.Sc. thesis, Carnegie Mellon Univ., Pittsburgh, PA, 2000.

[4] J. Cooley and J. Tukey, "**An algorithm for the machine calculation of complex Fourier series,**" *Math. Comput*, vol. 19, no. 90, pp. 297–301, 1965.

[5] P. Duhamel and H. Hollmann, "**Split-radix FFT algorithm,**" *Electron. Lett.*, vol. 20, pp. 14–16, Jan. 1984. [6] I. Kamar and Y. Elcherif, "Conjugate pair fast Fourier transform," *Electron. Lett.*, vol. 25, no. 5, pp. 324–325, Apr. 1989.

[7] S. Bouguezel, M. Ahmad, and M. Swamy, "**A new radix-2/8 FFT algorithm for length- DFTs**," *IEEE Trans. Circuits Syst. I*, vol.51, no. 9, pp. 1723–1732, Sep. 2004.

[8] E. Dubois and A. Venetsanopoulos, "**A new algorithm for the radix-3 FFT**," *IEEE Trans. Acoust., Speech, Signal Process,*, vol. ASSP-26, pp. 222–225, Jun. 1978.

[9] S. Prakash and V. Rao, "**A new radix-6 FFT algorithm,**" *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-29, no. 4, pp. 939–941, Aug. 1981.

[10] Y. Suzuki, T. Sone, and K. Kido, "**A new FFT algorithm of radix 3, 6, and 12**," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-34, no. 2, pp. 380–383, Apr. 1986.