National Institute of Technology Karnataka, Surathkal

Department of Electronics and Communication Engineering

EC383 Mini Project in VLSI Design

Design and Implementation of 64-point Fast Fourier Transform Chip for OFDM

Final Evaluation Report

Submitted to- Dr. Kalpana G. Bhat

Submitted By-
1) Rahul Gaikwad (201EC221)
2) Aditya Prasad (201EC205)

Contents

Design and Implementation of 64-point Fast Fourier Transform Chip for OFDM

1.1 Introduction and motivation

Fifth generation wireless and mobile systems are currently the focus of research and development. Broadband wireless systems based on orthogonal frequency division multiplexing (OFDM) will allow packet- based high-data-rate communication suitable for video transmission and mobile Internet applications. The IEEE 802.11a standard defines the principal functions and architecture of such a high-data-rate communication system. Apart from the high speed of operation, the system demands low power consumption since it is primarily aimed at portable and mobile applications. A general purpose DSP with associated software is not beneficial for this application since on average, the power consumption of a software solution is an order of magnitude higher compared to a functionally equivalent dedicated hardware solution. Considering this fact we proposed a data path architecture using dedicated hardware for the base band processor of the above- mentioned standard. We also showed through extensive simulation that the most computationally intensive parts of such a high-data-rate system are the 64-point inverse fast Fourier transform (IFFT) in the transmit direction and the Viterbi decoder in the receive direction. Accordingly, an appropriate design methodology for constructing them has to be chosen. For a given functional specification, the main design concerns are: 1) how much silicon area is needed; 2) how easily the particular architecture can be made flat for implementation in VLSI (routability); 3) in actual implementation how many wire crossings and how many long wires carrying signals to remote parts of the design are necessary (interconnect delay).4) how small the power consumption can be.

Extensive simulation of different algorithms and algorithm to-architecture mapping quality exploration is necessary to choose the best algorithm for a given specification.

1.2 Objectives

The objective of the project is to design a 64-point Fast Fourier Transform chip. This chip has been successfully simulated and tested. It performs a forward and inverse 64-point FFT on a complex two-s complement data set in 23 clock cycles, making it suitable for high-speed data communication systems like computer networking and radio/satellite communication.

The 64-point FFT is realised by decomposing a 64-point FFT into a two-dimensional structure of 8-point FFTs. The main benefit of using this approach for the FFT processor is that it requires less complex multiplication as compared to the conventional radix-2 64-point FFT algorithm. The same unit can be used for both FFT and IFFT by just swapping its real and imaginary parts , and we all know that the multiplier is the main component in FFT that consumes a lot of space. Since multipliers are typically very power-hungry elements in a VLSI design, this type of arrangement results in significant power consumption.

1.3 Methods

1.3.1 Introduction to Fast Fourier Transform-

Fast furrier transform (FFT) is one of optimal algorithms for DFT calculation and in most occasions its results are the same as DFT's (with the exception of rounding error). The operation number needed for DFT is N 2 and it could be greatly reduced using FFT methods. DFT is calculated by

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \; k = 0,1,\ldots,N-1$$

$$W_N = e^{-i2\pi/N}$$

Where, N is a power of 2 such that N=2m , and m is a natural number. This relationship can be divided to two relationships with length N/2 in a way that one of them involves odd members of x and another one utterly involves even members of x.

$$X(k)=\sum_{n_{even}=0}^{N-2} x(n)W_N^{nk}+\sum_{n_{odd}=0}^{N-1} x(n)W_N^{nk}$$

Substituting n with 2m yields:

$$X(k)=\sum_{m=0}^{N/2-2} x(2m)W_N^{2mk}+\sum_{m=0}^{N/2-1} x(2m+1)W_N^{(2m+1)k}$$
$$=\sum_{m=0}^{N/2-2} x(2m)\left(W_N^{2}\right)^{mk}+\sum_{m=0}^{N/2-1} x(2m+1)\left(W_N^{2}\right)^{mk}W_N^{k}$$

WN 2 can be simplified to

$$W_N^{2}=\left(e^{-i2\pi/N}\right)^2$$
$$=e^{-i2\pi/(N/2)}=W_{N/2}$$

So for DFT we have

$$X(k)=\sum_{m=0}^{N/2-1} x_{even}(m)W_{N/2}^{mk}+W_N^{k}\sum_{m=0}^{N/2-1} x_{odd}(m)W_{N/2}^{mk},$$

K= 0,1,...,N-1

Hence the DFT with N point changes into the DFT with N/2.

$$X(k)=DFT_{N/2}\left\{x_{even}(m),k\right\}+W_N^{k}.DFT_{N/2}\left\{x_{odd}(m),k\right\}$$

Fig. 2 shows how the FFT is calculated. Until now there was no simplification in operations (each element of X twice encounters N/2 operations e.g. for all X, the order o operations is N2). The periodic form of W is of significance such that it can be shown that:

$$W_N^{x+N/2}=W_N^{x}W_N^{N/2}$$
$$=W_N^{x}e^{-i\pi}=-W_N^{x}$$

In this way, one just need the half of W multiply operation. Therefore, fig. 2 will change into fig.3. It can be noticed in calculation that it is possible to decrease the total constants W such that all Ws convert to the equivalent WN. For example in fig. it can be placed W4 0 = W2 0 = W8 0 . Fig. is gained with exchanging the equivalent coefficients. Herein each resolution is decomposed to two smaller DFT; therefore, the FFT corresponds to the FFT group known as binary FFT. Also, because of time samples are recurrently divided into odd and even parts, it is known as decimation in time (DIT). If X(k) in each resolution becomes decomposed, an up duality called decimation in frequency (DIF) is gained.

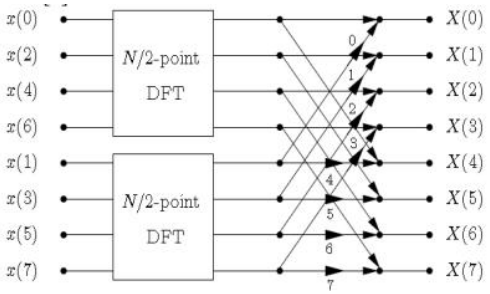

Fig.2. 8-point DFT graph for calculation of two DFT with N/2 points. Arrows indicate the multiply operation in W8 k and numbers on the arrows denote k[2].

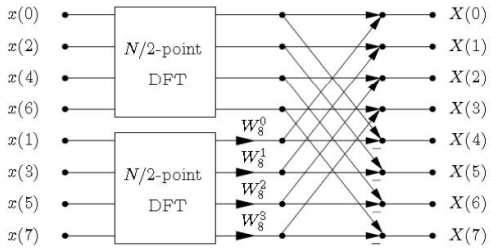

Fig.3. The modified 8-point DFT with periodic W for calculation of DFT with N/2 points. Arrows indicate multiply in W.

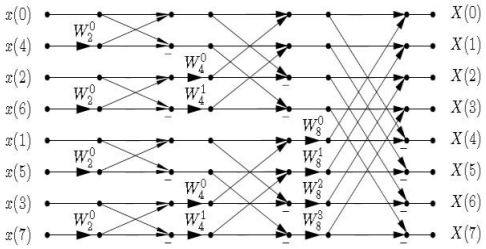

Fig.4.The completely decomposed 8-point DFT graph.

As mentioned before, our desired FFT is a 64-point one in base of 8 so considering the explanatory algorithm it can be written.

$$X(s+8t)=\sum_{l=0}^{7}\left[W_{64}{}^{sl}\sum_{l=0}^{7}x(l+8m)W_8{}^{sm}\right]W_8{}^{lt}$$

The graph of 64-point FFT in the base of 8 is depicted in terms of DIF in fig.

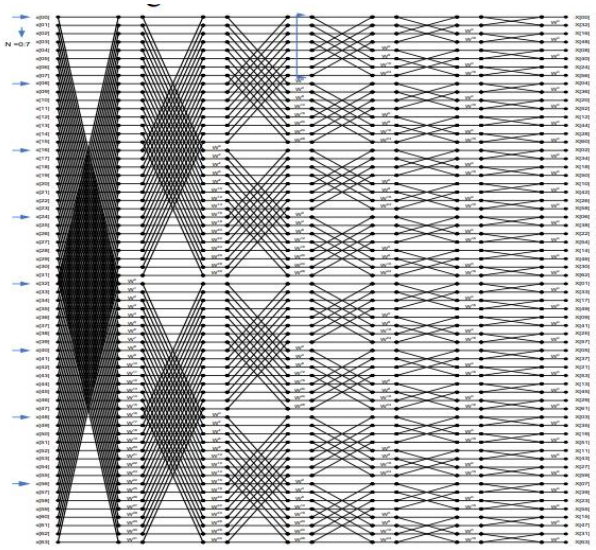

Fig.5.The completely decomposed 64-point DFT graph

1.3.2 Circuit Architecture-

The block diagram of the 64-point FFT/IFFT processor-

It consists of an input unit (I/P unit), two 8-point FFT units, a multiplier unit, an internal storage register bank (CB unit), an output unit (O/P unit), and a 5-bit binary counter that acts as the master controller for the entire architecture.

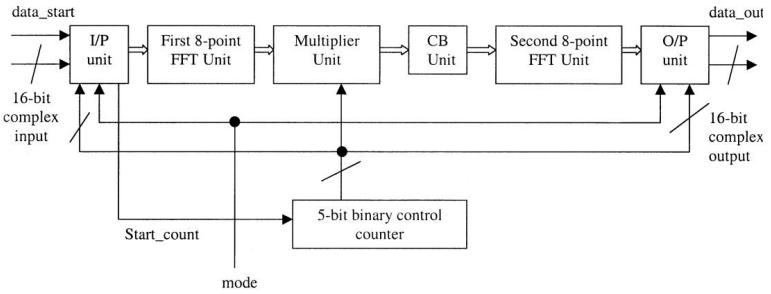

Fig.6 Block diagram of the proposed 64-point FFT/IFFT processor.

Architecture of chip-

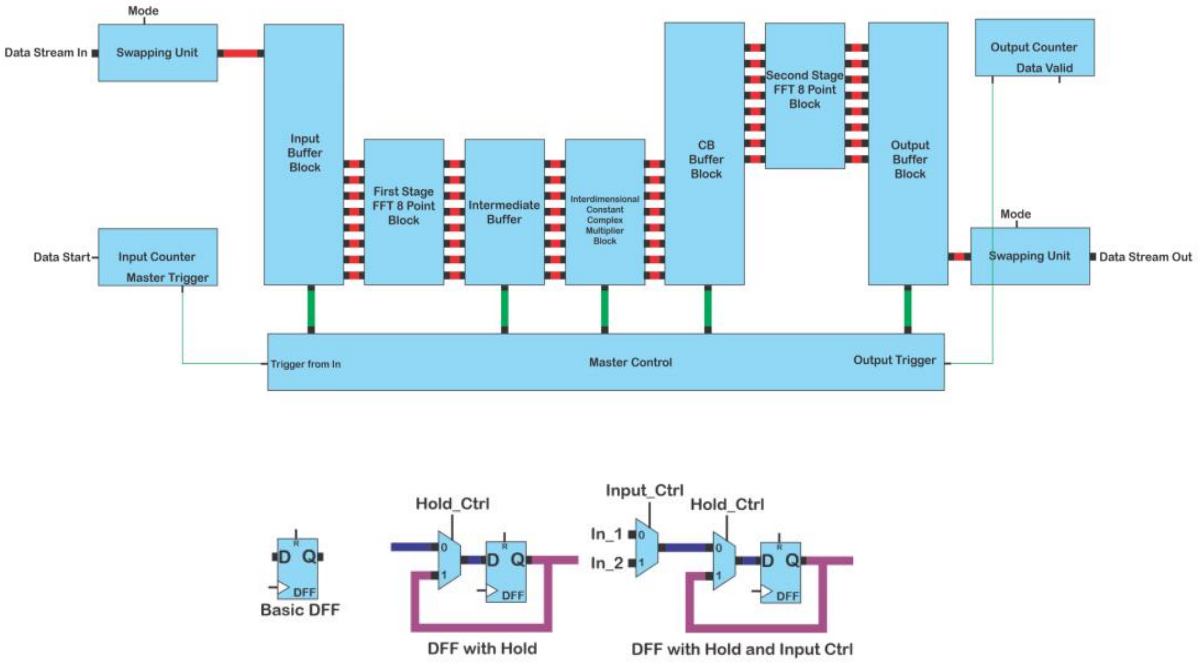

Fig.7 Architecture of chip

Input Circuit-

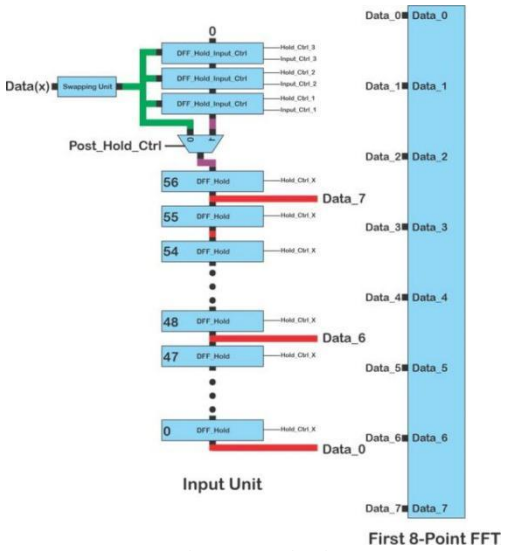

Fig.8 Input Circuit

Component and working of input circuit-

- Perform real-imaginary swapping, data buffering, and serial-to-parallel conversion
- Serial data is converted into 8 parallel data.
- When the input register bank is full, every 8th data is an input to the 8-point FFT unit
- Data shifted every cycle - avoids a parallel multiplexing scheme
- Because the multiplier may take more than one cycle to process the data, three additional buffers used.
- Counter - controls the serial input to the data. Provides the trigger to the master control circuit

Real-Imaginary swapping
Choose whether FFT or IFFT operation is performed. FFT (mode 0) operation does not require the real and imaginary part to be swapped at the beginning and ending, but IFFT (mode 1) does.

Data Buffering
The input data is serial from B(0) into B(63), but 8-point FFT block requires eight parallel data. The buffer will buffer 64 serial data and group them into 8 sets each with 8 data. Each set of data will become a set of input to the first stage of 8-point FFT block.

Data set arrangement
One Input data will be latched into Input Circuit per clock cycle, from B(0) to B(63) with a total of 64 data.Then, the data is grouped into 8 sets as shown below. • Each set enters the 8-point FFT block per clock cycle.

8-point FFT unit
Two 8-point FFT units are instantiated: first stage and second stage. Blue: Complex Adder, implemented as two 16-bit kogge-stone adder. Green: Complex Constant Multiplier, implemented as shift-and-add using 32-bit kogge-stone adder and bit truncation unit.

Interdimensional multiplier-

The output from the first 8-point FFT unit is multiplied with 64-point interdimensional twiddle factor. Out of 64 values, there are nine unique values that can be used to construct other values.

- (1,0) [bypass circuit]
- (0.995178, 0.097961)
- (0.980773, 0.195068)
- (0.956909, 0.290283)
- (0.923828, 0.382629)
- (0.881896, 0.471374)
- (0.831420, 0.555541)
- (0.773010, 0.634338)
- (0.707092, 0.707092)

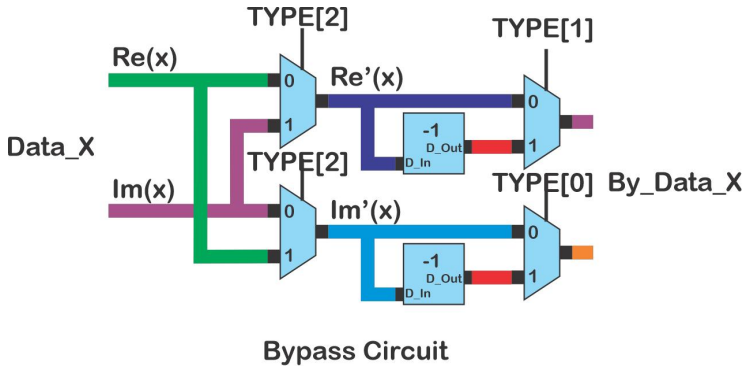

Fig. 9 Bypass Circuit

2.1 Matlab Code and Simulation-

Github Repo- Click here

2.1.1 Matlab analysis-

We give an 8 point sequence as an input (xt0, xt1,.., xt7) as a real and imaginary part and then convert the input to a fixed point numeric object using the fi function in Matlab and store the value in xt0_real.
Example-
xt0_real= fi(real(xt0), 1, 16, 12)
xt0_real = real part of xt0
DataTypeMode: Fixed-point: binary point scaling
Signedness: Signed
WordLength: 16
FractionLength: 12

Now convert xt0_real into xt0_real.bin and xt0_imag into xt0_imag.bin. This operation converts fixed point to binary form. Now concatenate the real and imag parts using strcat syntax in Matlab. The binary form is translated into decimal and then decimal to hexadecimal form, respectively, and stored in xf0. We have so far converted the input into hexadecimal form. Now get an 8-point FFT sequence output by writing a code for FFT calculation.
After getting FFT output now, we convert the output into hexadecimal as done above for the input sequence.

W_N^0	1	W_N^0
W_N^1	$\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}$	W_N^1
W_N^2	$-j1$	W_N^2
W_N^3	$-\frac{\sqrt{2}}{2} - j\frac{\sqrt{2}}{2}$	W_N^3
W_N^4	-1	$-W_N^0$
W_N^5	$-\frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$	$-W_N^1$
W_N^6	$j1$	$-W_N^2$
W_N^7	$\frac{\sqrt{2}}{2} + j\frac{\sqrt{2}}{2}$	$-W_N^3$

Fig. 9 Twiddle factor for 8 point FFT

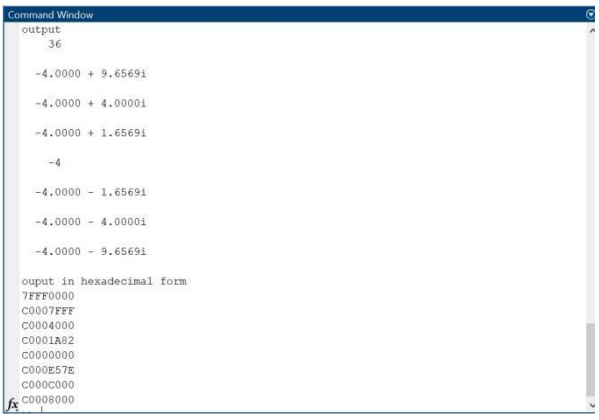

Fig. 10- Matlab Simulation: 8 point FFT output

MATLAB Variable: A64_Fixed
30 Sep, 2022

Page 1
6:23:13 PM

	1	2	3	4
1	'20612061'	'FFCD0031'	'FF4BFFAF'	'FEC7FF2B'
2	'09200A8B'	'FF1F00D5'	'FEA2005A'	'FE16FFD4'
3	'042C063C'	'FF7A01A5'	'FF02012E'	'FE7200A2'
4	'03550491'	'005C018F'	'FFEC011D'	'FF560082'
5	'02D102E0'	'00A700A5'	'00380035'	'FF9DFF8D'
6	'01A801B7'	'FFEFFFFB'	'FF7CFF89'	'FEC7FECD'
7	'007401B4'	'FF0A0055'	'FE9DFFEA'	'FDCDFF1F'
8	'003C0263'	'FF0D013E'	'FEA200D3'	'FDBDFFF8'
9	'00E50287'	'FFE9018C'	'FF81011F'	'FE810022'
10	'016301B7'	'008100D0'	'001A0063'	'FEF6FF34'
11	'00E000BE'	'0016FFF0'	'FFAAFF82'	'FE55FE20'
12	'FFDA00AB'	'FF20FFF3'	'FEADFF84'	'FCF2FDC9'
13	'FF710174'	'FEC500CE'	'FE550061'	'FC19FE36'
14	'000A0202'	'FF70016B'	'FF0000F8'	'FC01FE05'
15	'00D2018F'	'00460101'	'FFD60089'	'FB39FBD4'
16	'00BD0094'	'00320007'	'FFBAFF88'	'F603F588'
17	[]	[]	[]	[]
18	[]	[]	[]	[]
19	[]	[]	[]	[]
20	[]	[]	[]	[]
21	[]	[]	[]	[]
22	[]	[]	[]	[]

Fig. 11- Matlab Simulation: 64 point FFT output

3.1 Verilog Code and Simulation-

Github Repo- Click here

3.1.1 RTL Design using Verilog simulation-

The architecture was first modeled in Verilog and functionally verified using Quartus prime Modelsim simulator. The outputs from the Verilog coded architecture are validated against a standard C-coded FFT routine. According to the specification, a clock frequency of 20 MHz is used. The input data are delivered to the processor in a serial manner. The latency of the processor is 3.85 s, i.e., 77 clock cycles. The parallel to parallel (i.e., the basic 64-point FFT computation, assuming parallel input data and parallel output data) FFT/IFFT computation requires 1.15 s, i.e., 23 clock cycles. The throughput rate of the architecture is one complete set of 64-point FFT/IFFT (provided as serial output) in 3.2 s, i.e., 64 cycles. The minimum time interval between two successive sets of input data (i.e., last data of the previous set and the first data of a new set) for which the architecture shows correct functional behavior is equal to three clock cycles.

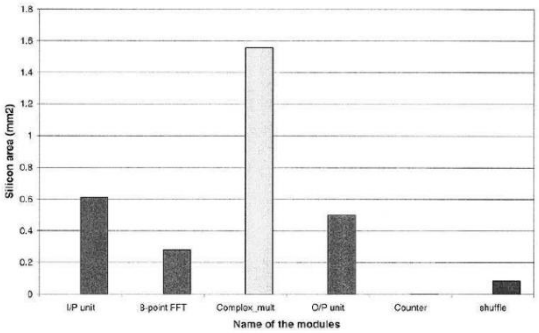

Fig.12 Synthesized cell area of different constituent blocks of the 64-point FFT/IFFT processor.

After functional validation, the architecture was synthesized for IHP 0.25- m three-metal layer BiCMOS technology using the Quartus Prime Analyzer. The synthesized circuit was then back-annotated using the Modelsim simulator, showing the correct functional behavior. The synthesized cell area of the complete processor is 3.6 mm . The cell area requirement for the different blocks of the

design is shown in Fig. 12 where the combined area of the hard-wired multiplier unit, its input/output shuffle network and the internal storage register bank CB is represented in one block termed as complex_mult unit. The combined cell area of all the other data shuffling networks are denoted as shuffle and the area requirement for the master control counter is denoted as counter in Fig. 12. As expected, the complex_mult unit consumes the largest cell area. After synthesis, floor planning and layout were carried out using Quartus Prime. The core area after layout is 6.8 mm and the complete area including power rings and I/O pads is 13.5 mm . The processor has 85 I/O pins of which 36 are input pins and 33 are outputs. The rest of the pins are used for power supply. The die photograph of the fabricated chip is shown in Fig. 13

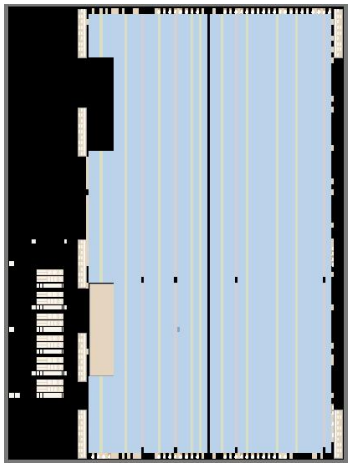

Fig. 13. Die photograph of the fabricated 64-point FFT/IFFT processor

3.2 Implementation

After the functional verification, our design in which the internal word length is 16-bit has been synthesized with Modelsim simulator in CMOS technology. After the back-annotation from Start-RC extractor, the post-simulation has been issued by simulator to verify the functionality. Finally, the power analysis can be done by Intel Quartus Prime. The average power dissipation of the proposed 64-point FFT/IFFT design is 22.36 mW at20 MHz at 1.2V supply voltage. The active chip layout area of the proposed design as shown in Fig. 14 is 1606 um × 1606 um, which has 77 I/O pins where 8 pins are power supply pins. The proposed 64-point FFT/IFFT design not only meets 3.2 μs timing specification for IEEE 802.11a standard, but also achieves the low power and cost-effective feature.

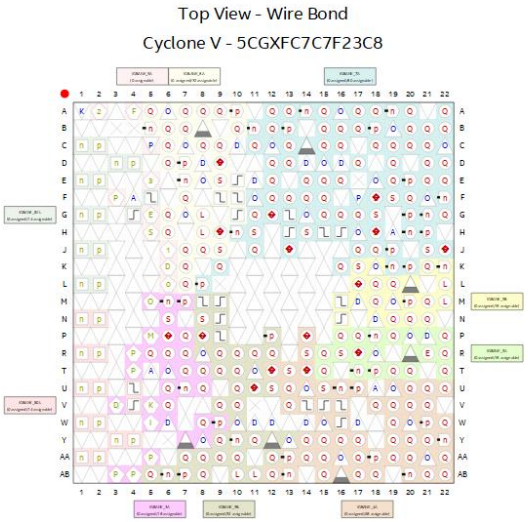

Fig.14 Pin Diagram

3.3 Test Strategy

The inherent linear structure of the architecture greatly simplifies the testing of the chip. We adopted the conventional full scan test for the chip. Existing flip-flops were replaced by scan flip-flops during the synthesis phase. The complete design contains 7134 scan flip-flops. The output scan_out pin is merged with the most significant bit of the real component of the output register out.

3.4 Measurement Results

Testing was carried out using the Quartus prime analyzer. For functional tests, precalculated data vectors were fed to the chip in a continuous manner and the output was checked while operating the chip at 20 MHz frequency. This test validates the functional behavior of the chip. After the functional test, a test for maximum allowable frequency of operation was carried out. The measurement result shows that at 2.5 V supply voltage the chip exhibits correct functional behavior up to 38 MHz. The average power dissipation of the processor measured over 55 fabricated chips is 84 mW at 2.5 V and 20 MHz clock frequency. In the second phase, the function of the chip was tested at lower supply voltage. It was found that the chips operate correctly down to 1.8 V supply voltage. However, the maximum allowable frequency in this case is 26 MHz, which still satisfies our target frequency. In the third phase, the scan test was carried out using the vector set generated by the Quartus prime analyzer. This is done by operating the processor in the scan mode. Vector data generated by Quartus Prime analyzer are directly translated to the tester format. In this mode of operation, parallel vectors are inputted to the processor for three successive cycles (capture cycles) and in the next cycle, a scan operation is carried out.

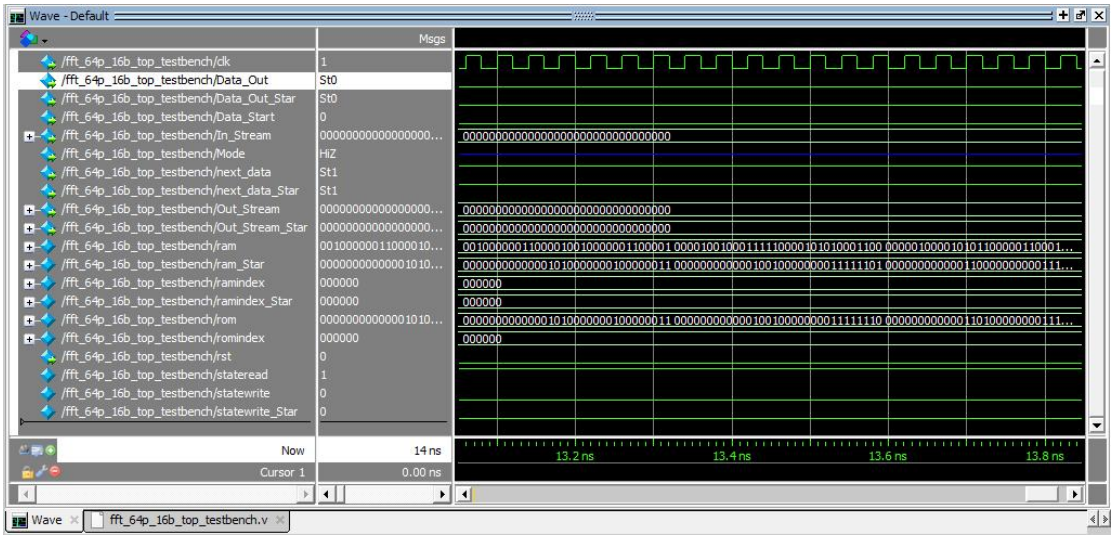

Fig.15 RTL Synthesis Simulation

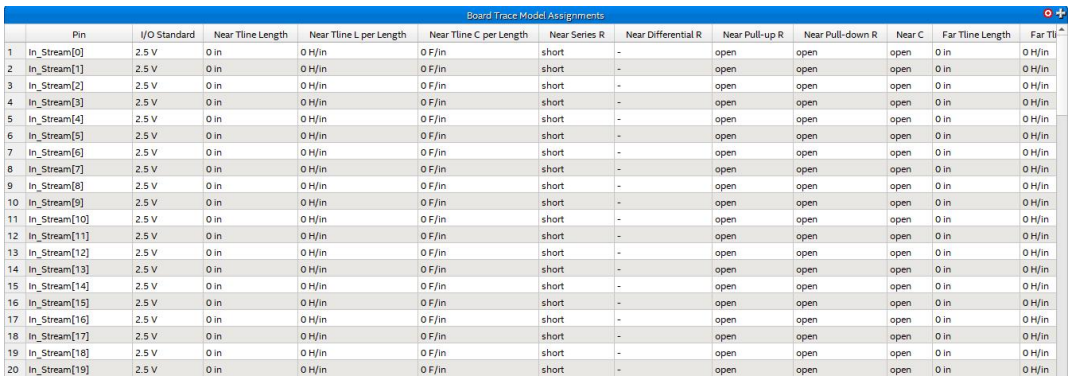

Fig.16 Time analysis

Timing Analyzer Summary	
Quartus Prime Version	Version 21.1.0 Build 842 10/21/2021 SJ Lite Edition
Timing Analyzer	Legacy Timing Analyzer
Revision Name	fft_64p_16b_top_testbench
Device Family	Cyclone V
Device Name	5CGXFC7C7F23C8
Timing Models	Final
Delay Model	Slow 1100mV 85C Model
Rise/Fall Delays	Enabled

Fig.17 Time Analyzer summary

Flow Summary

<<Filter>>

Flow Status	Successful - Tue Nov 22 00:08:14 2022
Quartus Prime Version	21.1.0 Build 842 10/21/2021 SJ Lite Edition
Revision Name	fft_64p_16b_top_testbench
Top-level Entity Name	fft_64p_16b_top_testbench
Family	Cyclone V
Device	5CGXFC7C7F23C8
Timing Models	Final
Logic utilization (in ALMs)	1 / 56,480 (< 1 %)
Total registers	0
Total pins	104 / 268 (39 %)
Total virtual pins	0
Total block memory bits	0 / 7,024,640 (0 %)
Total DSP Blocks	0 / 156 (0 %)
Total HSSI RX PCSs	0 / 6 (0 %)
Total HSSI PMA RX Deserializers	0 / 6 (0 %)
Total HSSI TX PCSs	0 / 6 (0 %)
Total HSSI PMA TX Serializers	0 / 6 (0 %)
Total PLLs	0 / 13 (0 %)
Total DLLs	0 / 4 (0 %)

Fig.18 Power analyzer report summary

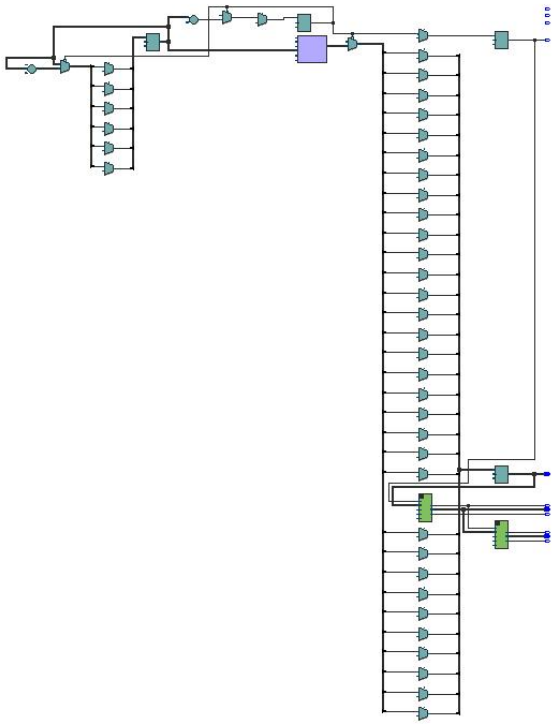

Fig19 RTL Diagram

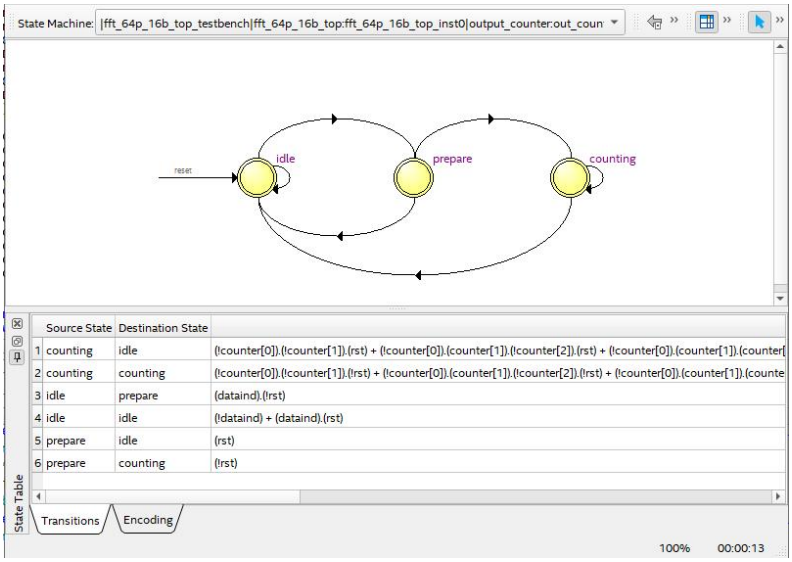

Fig.20 State machine

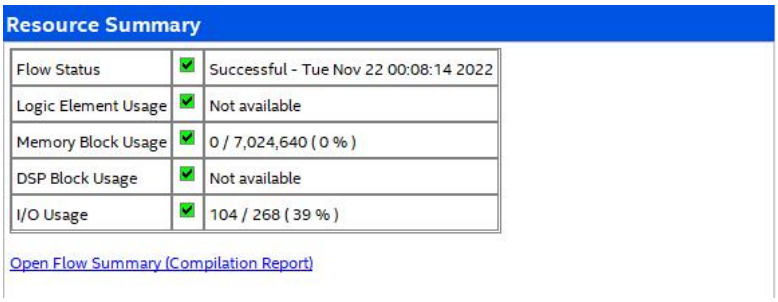

Fig.21 resource Optimizer

4. Conclusion

Proposed method describes a novel 64-point IFFT/FFT architecture to be used in high speed WLAN system based on OFDM transmission. The architecture is based on the technique of decomposing a 64-point FFT into two 8-point FFT. Proposed technique exhibits attractive features like modularity, regularity, simple wiring and high throughput. This architecture consumes less silicon area and results in considerable reduction in cost. Both IFFT/FFT can be realized by same architecture we just have to swap real and imaginary part. Number of non-trivial complex multiplication are 49 that is 26% less than required by radix-2 64 point FFT. For computation of IFFT/FFT full parallel scheme is adopted there by making it very fast. Complex multiplication constant are calculated using shift and add operation there by reducing area and power consumption as compared to multiplier. With proposed design full parallel 64 point FFT/IFFT can be computed in 23 clock cycles.

References

1) K. Maharatna, E. Grass, and U. Jagdhold, "A 64-Point Fourier Transform Chip for High-Speed Wireless LAN Application Using OFDM," IEEE J. Solid-St. Circ., Vol. 39, No. 3, Mar. 2004, pp. 484-493
2) Implementation Of Efficient 64-Point FFT/IFFT Block For OFDM Transreciever Of IEEE 802.11a
3) A Low-Power 64-Point FFT/IFFT Design for IEEE 802.11a WLAN Application
4) International Journal of Scientific & Engineering Research, Volume 3, Implementation of FFT Algorithm for OFDM Wireless LANs
5) Variable length mixed radix MDC FFT/IFFT processor for MIMO-OFDM application