

HIGH PERFORMANCE SPLIT RADIX FFT

Karishma A. Deshmukh Prof. P. R. Indurkar Prof. Mrs. D. M. Khatri
M.TECH IV SEM(VLSI),R.T.M.N.U E&TC Department & university E&TC Department & university

Abstract— A high performance hardware FFT have various application in instrumentation and communication systems. It describes new parallel FFT architecture which come together the split-radix algorithm with a constant geometry interconnect structure. The split-radix algorithm knows to have lower multiplicative difficulty than both radix-2 as well as radix-4 algorithm. The split-radix algorithm maps onto a constant geometry interconnect structure in which the wiring in each FFT stage is the same, resulting in low multiplexing overhead. We are exploiting the lower arithmetic obscurity of split-radix to lower dynamic energy, by gating the multipliers during trivial multiplication. The proposed FFT accomplishes less power than a parallel radix-4 design when calculate at few point, the real-valued transform

Keywords— FFT (Fast Fourier Transform), orthogonal frequency division multiplexing (OFDM), ultra-wideband (UWB)

I. INTRODUCTION

Discrete Fourier Transform generates a predefined duration discrete frequency sequence that is obtained by sampling one period of Fourier Transform. Fast algorithms for determine the DFT which are called as fast Fourier transform (FFT). FFT determine the DFT and produces accurately the same result as calculating the DFT definition directly. The difference is that an FFT is much faster. FFT is applicable in communications. The split-radix FFT has minor complication than the radix-4 or any higher-radix power-of-two FFT.

Y. Chen presented a new dynamic voltage and frequency scaling (DVFS) FFT processor for MIMO OFDM applications. MIMO OFDM systems had achieved better reliability and superior capacity, the power consumption also increases because of the unlimited difficulty for the multi-stream processing. Therefore, less power becomes a major target in design of MIMO OFDM devices, especially for portable applications. To save the power of the FFT processor cultivate proportionally to the stream number this occupies a large percentage of system power. The dynamic voltage and frequency scaling (DVFS) is an effective technique to achieve less power. Parallelized radix-2 FFT structure has been adopted to store the power consumption [8]. In 2005, Y. W. Lin shows that the pipelined FFT architecture which is called as mixed-radix multipath delay feedback (MRMDF). The power consumption and hardware cost was also collect in this processor by using the higher radix FFT algorithm with less memory and complex multipliers. A novel 128-point FFT/IFFT processor for OFDM-based UWB systems was proposed. In addition, the number of complex multiplications is less effectively reduced by using a higher radix algorithm [7]. In 2010 Shen-Jui Huang used the eight-data-path 2048-point FFT processor that has been proposed with a T.R. of 2.4 GS/s for OFDM-based gigabit WPAN application as well as consumed less power. Proposed work was based on split radix FFT architecture. FFT Computation can be done using different algorithm based on Radix 2, Radix 4, Radix 8, Split Radix etc. The split-radix algorithm has lesser multiplicative complexity than both radix-2 and radix-4 algorithms. The split-radix FFT mixes radix-2 and radix-4 decompositions. The split-radix FFT has lower complication than the radix-4 or any higher-radix power-of-two FFT.

II. PARALLEL FFT ARCHITECTURE

In the previous work author observe on the pipelined FFT architecture using more pipelines, consisting of $\log_r N$ stages of butterfly data-paths.

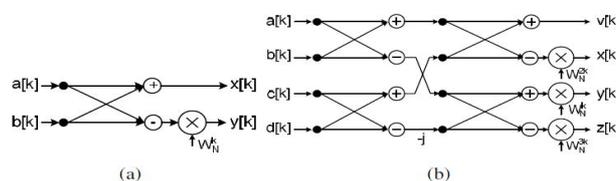


Fig. 1. Block diagram of (a) radix-2 and (b) radix-4 butterfly.

This paper describes parallel architecture with N/r butterfly data-paths. Fig.1 shows the structure of radix-2 and radix-4 butterfly. The N -point, radix- r complex valued FFT contains $N/r \log_r N$ butterfly operations, therefore the throughput of an FFT architecture is limited by the number of butterfly operations computed in parallel.

The row wise parallelization is used in pipelined FFTs, but it is doesn't impossible to parallelize along a column of the SFG. The N/r butterflies are instantiated to determine one stage means one column of the FFT. One column of registers is used to register the output.

III. SPLIT-RADIX CONSTANT GEOMETRY FFT

Split-radix algorithm is a constant geometry consistent structure. The profits of this structure are lower power, shorter critical path than radix-4 constant geometry architecture. The split-radix algorithm provides the advantage of smaller amount of non-trivial complex multiplication and addition than radix-2, radix-4, and radix-8 algorithm. The split-radix algorithm is based on [3] succession decomposing in an N -point DFT into an $N/2$ -point DFT and two $N/4$ -point DFTs. $W_N^k = e^{-j2\pi k/N}$, this is a twiddle factor used in FFT calculation. Using a pair of radix-2-like butterflies as the basic processing unit. The split-radix SFG thought of as a Cooley-Tukey radix-2 SFG, except with dissimilar twiddle factors and details in the butterflies. The split-radix SFG can be reschedule in the same way that a radix-2 Cooley-Tukey algorithm is changed into a constant geometry algorithm. The odd butterflies do not compulsory any multipliers, while the even butterflies necessitate two due to the presence of two twiddle factors. Thus referring to these butterflies as type 1 and type 2 butterflies vary from radix-2 butterflies in two main ways. First, when the butterfly is used to determine the second half of the 4-point DFT a subtraction occurs in the top branch rather than the bottom branch. Second, butterflies are modified to calculate multiplication by $\sqrt{-1}$ (j) without having to use the complex multiplier. This is particularly important in the split-radix design because the algorithm decreases the number of non-trivial multiplications at a cost of additional multiplications by j . These numbers of multiplications obtained only by non-trivial twiddle factors and by j .

IV. MULTIPLIER GATING FOR SWITCHING POWER REDUCTION

The lesser multiplicative complexity of split-radix can be leveraged to reduce power. Specifically, trivial multiplications (i.e. multiplying by 1 or $\sqrt{-1}$). More importantly, during bypass, the multiplier inputs should be assumed that values from the previous cycle in order to suppress switching. At single multiplier input, this is done by putting in a latch. When the multiplier is bypassed, the latch holds data from the previous cycle, otherwise the latch is clear. Like the other control signals, the latch enable signals can be controlled via minute, hard-coded lookup tables. The latches can be build-up during timing verification by forcing the enable signal to "1", in which case they add a small delay to the logic path. The proposed architecture can be extended to efficiently determine FFTs on real-valued inputs using the well-known approach described in [4].

The other multiplier input is controlled by tough-coded twiddle factor lookup tables. Therefore, we can modify the tables to replace "1" or " $\sqrt{-1}$ " values with twiddle factor values from the previous cycle. The net result is that during trivial multiplications, the multiplier output remains not changed from the previous cycle, but this incorrect output is by passed. In this approach, the even and odd samples of an N -point real sequence are used to form the real and imaginary parts of an $N/2$ -point complex sequence. Then, an $N/2$ -point complex-valued FFT is determined using the split-radix algorithm as described above. This is tracked by real-valued post-processing [6].

V. PROPOSED METHODOLOGY

The proposed methodology are design n bit Adder for addition purpose, design n bit Multiplier for multiplication, design n bit complex multiplier is used to perform complex multiplication in FFT algorithm and also design n bit split radix butterfly for generate the butterfly structure, design reorder and the storing purpose design the register.

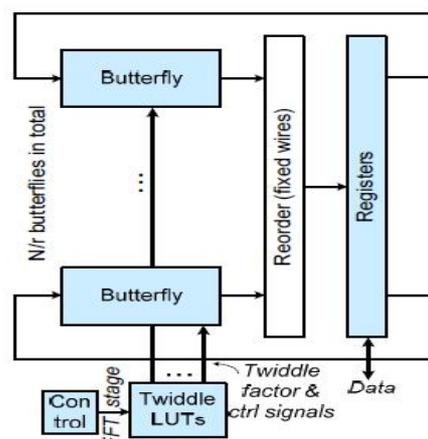


Fig:2 Parallel constant geometry FFT architecture.

In above show in the FFT architecture diagram .By the help of this diagram we design the code for floating point adder, floating point multiplier as well as One stage FFT. Also design the two stage FFT and Three stage FFT. And finally design code the Butterfly using split radix FFT. In the proposed methodology, at that time we work on the Adder and collect the output.

VI. RESULT

Split-radix algorithm is known to have lower multiplicative complexity than both radix-2 and radix-4 algorithms. It is used to generate the butterfly structure, like the upper and lower node computation. For the reordering, the output purpose reorder block is used where reorder is register base. For output storing purpose register is used and Control block is used to control the overall system. N bit Adder is used to perform addition in FFT algorithm. Multiplier is used to perform real multiplication in FFT algorithm. Sequential type of multiplier has been used in these works. Complex multiplier is used to perform complex multiplication in FFT algorithm. Using the multiplier designed in the previous literatures split-radix butterfly algorithm is realised.

ADDER:-

In the Design adder we perform the some Arithmetical addition and we get the below output.

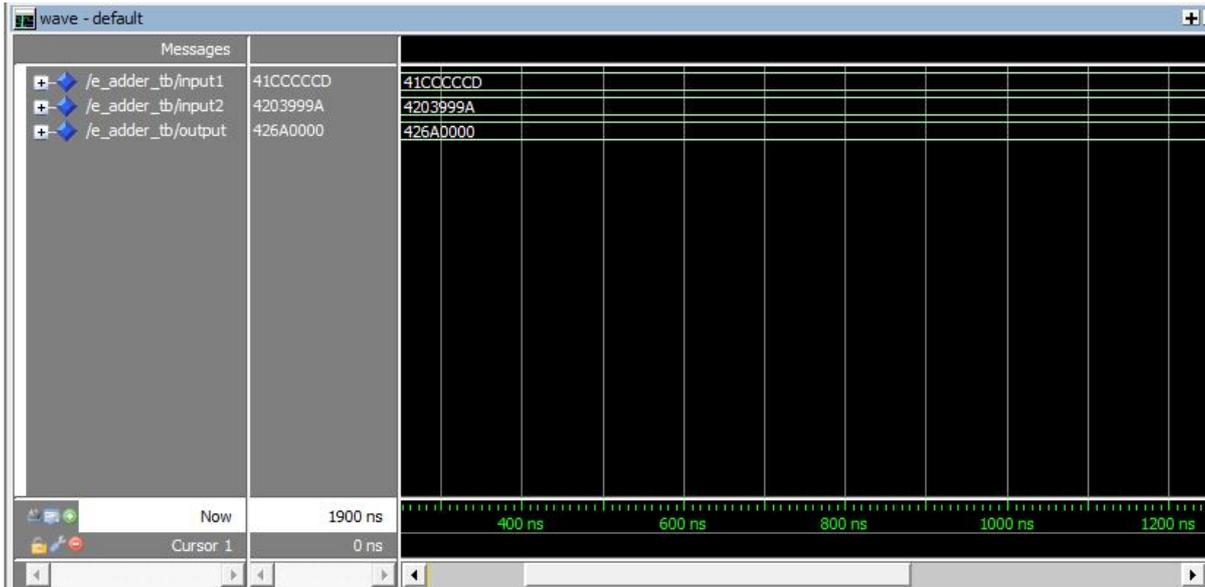
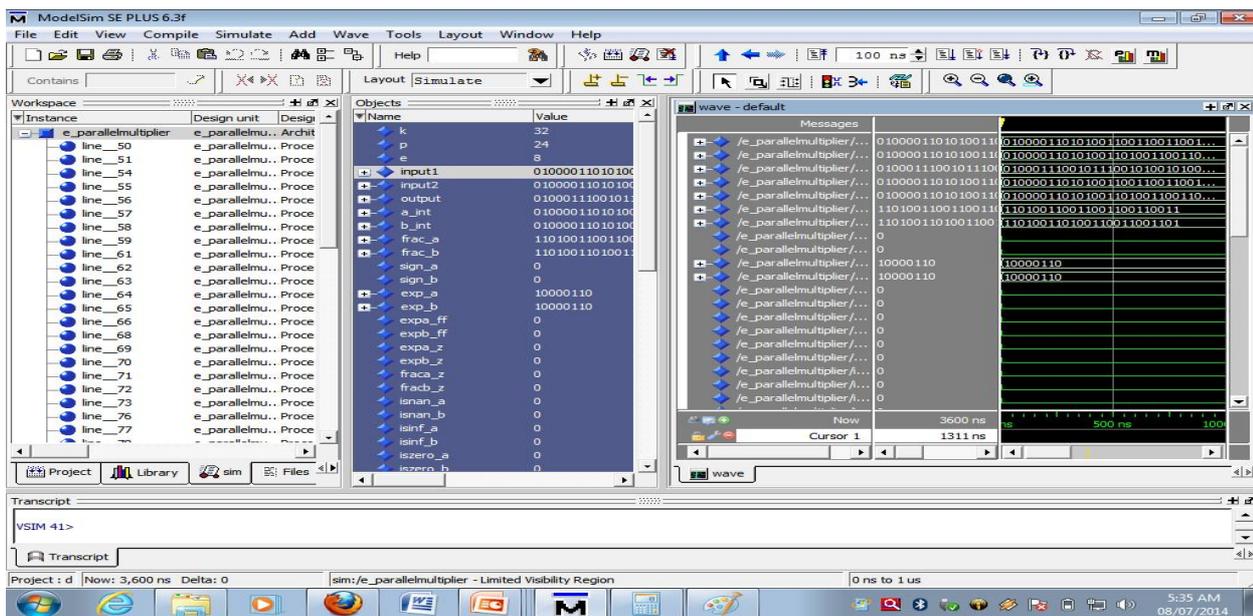


Fig:3 Output waveform of adder.

In above waveform is a output of the adder. We add the 25.6 and 32.9 means in the hexadecimal form is the 41CCCCCD And 4203999A and we get a output is the 58.3 means in hexadecimal is 426A0000. Adder verified and simulated by using VHDL in Model-Sim simulator. Above figures shows the simulation waveforms of Adder

MULTIPLIER:-

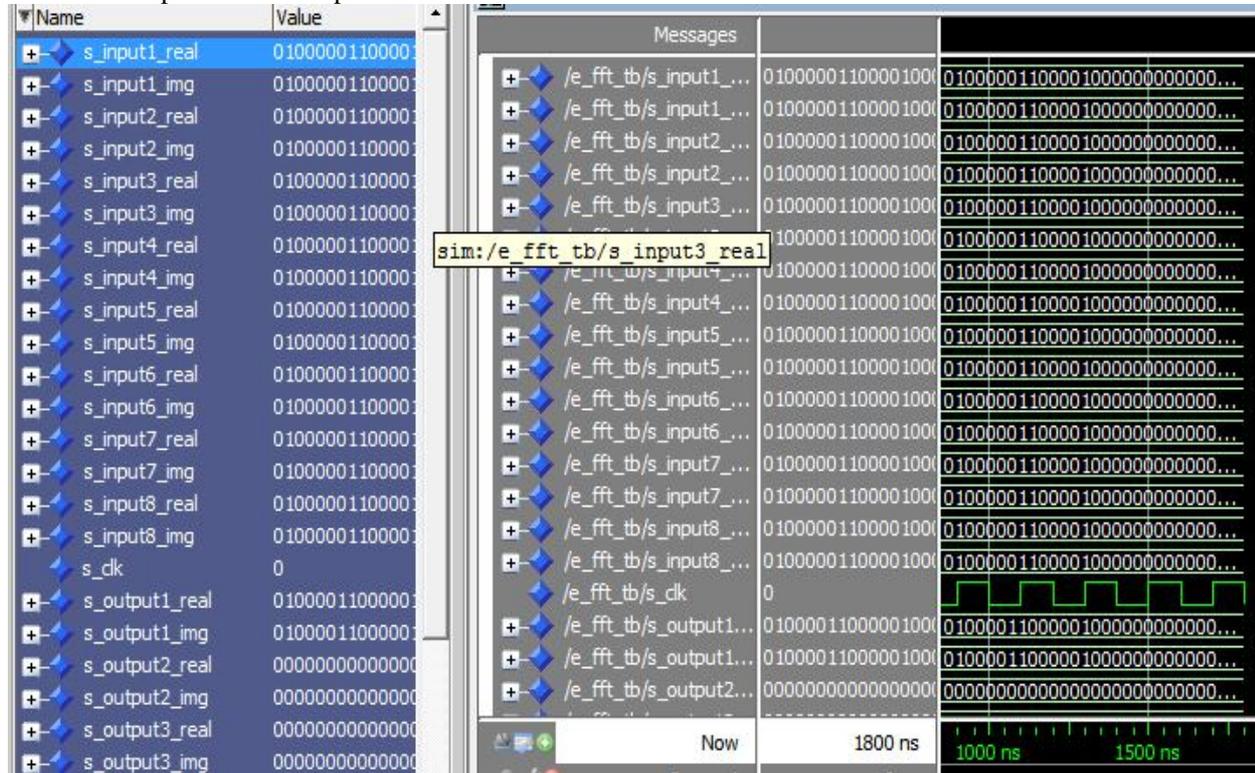


In the Design adder we perform the some Arithmetical multiplication and we get the below output.

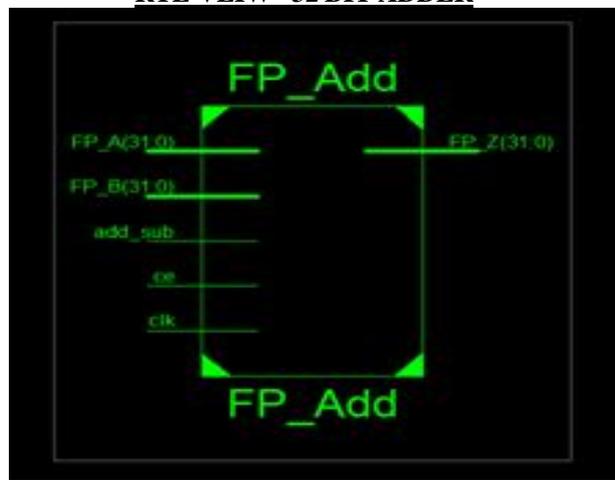
In above waveform is a output of the multiplier. The inputs are 211.2 and 211. we get a output is the 44626.56. multiplier verified and simulated by using VHDL in Model-Sim simulator. Above figures shows the simulation waveforms of multiplier.

FFT:-

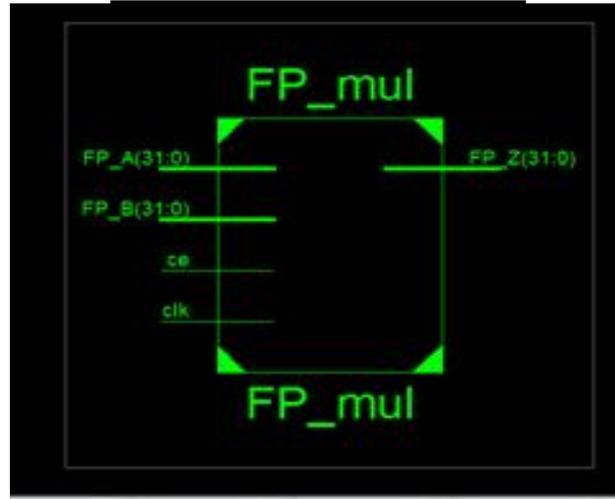
This is the output of the multiplier.



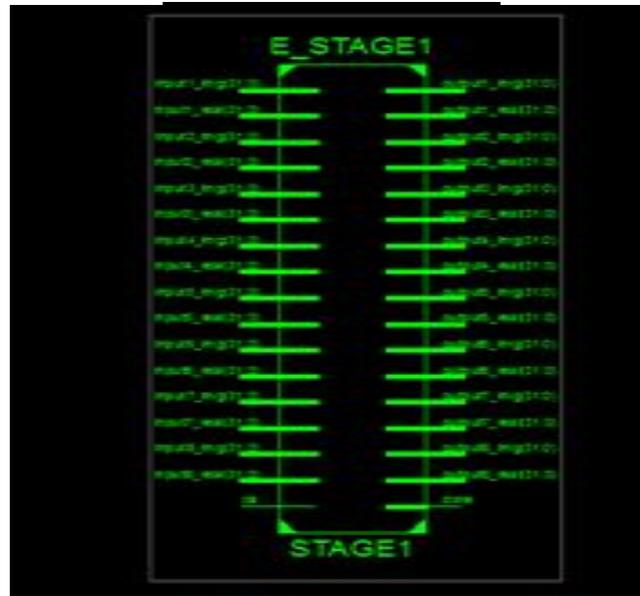
RTL VIEW 32 BIT ADDER



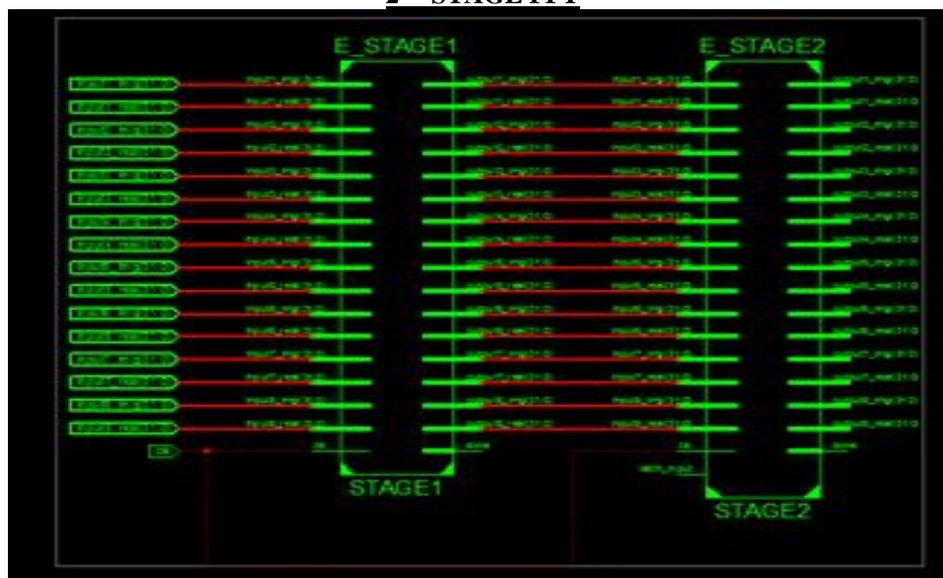
RTL VIEW 32-BIT MULTIPLIER



RTL VIEW 1ST STAGE FFT



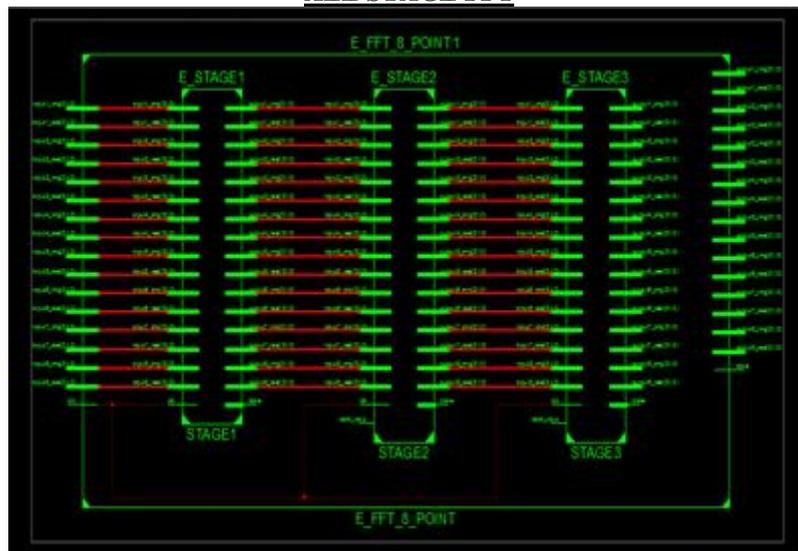
2ND STAGE FFT



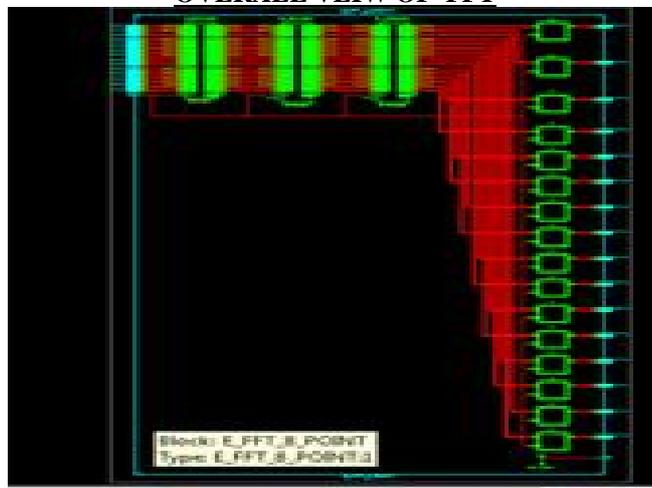
3RD STAGE FFT



ALL STAGE FFT



OVERALL VIEW OF FFT



The Area of the overall Split Radix FFT is given below. As well as the delay of split radix FFT is = 64.259ns .

Device Utilization Summary (esti	
Logic Utilization	Used
Number of Slices	54356
Number of Slice Flip Flops	1586
Number of 4 input LUTs	101550
Number of bonded IOBs	1026
Number of GCLKs	1

Above fig. shows that the area of the FFT. We get the synthesis output of area is the number of the slices is 54356.and number of the slice flip flops 1586.

COMPARISION TABLE FOR FFT:

PARAMETER	PREVIOUS	PROPOSED
Data path Width(bits)	12	32
FFT Size	128	256
Architecture	Column wise parallel	Column wise parallel
Power(mV)	60.9	46
Speed.	More than 15.56Mbps	15.56Mbps
Type of numbers	Real number	Floating Point number

VII. CONCLUSIONS

In the proposed work, the probable outcome is low power consumption in FFT design and the proposed is reduction calculation time in FFT design. The Floating point FFT is increase the speed of calculation. Reduced FFT calculation time. Power consumption of the system is less. Floating point FFT having 15.56Mbps speed as well as power is 46% .

ACKNOWLEDGMENT

I wish to thank Prof. P. R. Indurkar and Prof .Mrs. D. M. Khatri for their guidance and support.

REFERENCES

- [1] M. C. Pease, "An adaptation of the Fast Fourier Transform for parallel processing," *Journal of the ACM*, vol. 15, pp. 252–264, April 1968.
- [2] M. Corinthis, "The design of a class of Fast Fourier Transform computers," *IEEE Transactions on Computers*, vol. C-20, no. 6, pp.617–623, June 1971.
- [3] P. Duhamel and H. Hollmann, "'Split radix' FFT algorithm," *Electronics Letters*, vol. 20, no. 1, pp. 14 –16, May 1984.
- [4] H. Sorensen, D. Jones, M. Heideman, and C. Burrus, "Real-valued Fast Fourier Transform algorithms," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 6, pp. 849–863, June 1987.
- [5] F. Argu'ello and E. Zapata, "Constant geometry split-radix algorithms," *Journal of VLSI Signal Processing*, 1995.
- [6] R. Matusiak. (2001, Aug.) Implementing Fast Fourier Transform algorithms of real-valued sequences with the TMS320 DSP platform. [Online]. Available: <http://focus.ti.com/lit/an/spra291/spra291.pdf>
- [7] Y.-W. Lin, H.-Y. Liu, and C.-Y. Lee, "A 1-GS/s FFT/IFFT processor for UWB applications," *IEEE Journal of Solid-State Circuits*, vol. 40, no. 8, pp. 1726 – 1735, Aug. 2005.
- [8] Y. Chen, Y.-W. Lin, Y.-C. Tsao, and C.-Y. Lee, "A 2.4-Gsample/s DVFS FFT processor for MIMO OFDM communication systems," *IEEE Journal of Solid-State Circuits*, vol. 43, no. 5, pp. 1260–1273, May 2008.
- [9] S.-N. Tang, J.-W. Tsai, and T.-Y. Chang, "A 2.4-GS/s FFT Processor for OFDM-Based WPAN Applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 57, no. 6, pp. 451–455, June 2010
- [10] S.-J. Huang and S.-G. Chen, "A green FFT processor with 2.5-GS/s for IEEE 802.15.3c (WPANs)," in *International Conference on Green Circuits and Systems*, June 2010, pp. 9 –13.
- [11] T. Cho, H. Lee, J. Park, and C. Park, "A high-speed low-complexity modified radix-25 FFT processor for gigabit WPAN applications,"