Journal of Computer Science 3 (6): 376-382, 2007
ISSN 1549-3636

Low Power Hardware Implementation of High Speed FFT Core

M. Kannan and S.K. Srivatsa
Department of Electronics Engineering, Madras Institute of Technology
Anna University, Chennai-44, India

Abstract: Applications based on Fast Fourier Transform (FFT) such as signal and image processing require high computational power. This paper proposes the implementation of radix-4 based parallel-pipelined Fast Fourier Transform processor which incorporates a low power commutator, butter-fly with multiplier-less architecture. The proposed parallel pipelined architectures have the advantages of high throughput and low power consumption. The multiplier-less architecture uses shift and addition operations to realize complex multiplications.

Key words: Dual port RAM, shift register, finite state machine

INTRODUCTION

Fourier transforms play an important role in many digital signal processing applications including speech, signal and image processing. However, direct computation of Discrete Fourier Transform (DFT) requires on the order of N^2 operations where *N* is the transform size. The FFT algorithm, first explained by Cooley and Tukey[1], opened a new area in digital signal processing by reducing the order of complexity of DFT from N^2 to $N \log_2 N$.

Parallel-pipelined FFTs are preferred for both high throughput and low power consumption. In real-time applications, input data is a sequential stream. For this reason, the commutator is needed to reorder the input data. The proposed architecture in this paper is an improvement to be power efficient compared to previous commutator architectures used in pipelined FFT[2]. It is well known that the switching power is mainly responsible for power consumption in CMOS circuits. This power, Psw, is given by

$$p_{sw} = \frac{1}{2} k c_{load} v_{dd}^2 f \quad (1)$$

Where *k* is the average number of times the gate makes an active transition during one clock cycle, *f* is the clock frequency, *Vdd* is the supply voltage and *Cload* is the load capacitance of the gate. Hence, for achieving low power, one or more of the parameters *Cload*, *Vdd* and *k* need to be minimized. However, since *Cload* and *Vdd* are relative to the target technology, *k* becomes the main point of improvement. In[2] implementation is done by reducing the switching activity only. Therefore, this paper focuses on implementing commutator with no switching activity, hence achieving a significant power saving as compared to previous commutator architectures[3].

16 point 2-parallel-pipelined FFT: The DFT of *N* complex data points x (*n*) is defined by :

$$X(k) = \sum_{n=o}^{N-1} x(n) W_N^{nk}, \text{ k = 0, 1, 2... N-1} \quad (2)$$

Where $W_N = e^{-j\frac{2\pi}{N}}$ is twiddle factor. The original DFT needs 256 multiplications, 240 additions for 16 points. Since 16 is a power of four, radix-4 decimation-in-frequency algorithm is used to break the 16-point DFT formula into four smaller DFTs. The FFT is the speed-up algorithm of DFT. The final sets of transforms look like

$$X(4k) = \sum_{n=0}^{N/4-1} [x(n)+x(n+\tfrac{N}{4})+x(n+\tfrac{N}{2})+x(n+\tfrac{3N}{4})]W_N^0 W_{N/4}^{kn} \quad (3)$$

$$X(4k+1) = \sum_{n=0}^{N/4-1} [x(n)-jx(n+\tfrac{N}{4})-x(n+\tfrac{N}{2})+jx(n+\tfrac{3N}{4})]W_N^n W_{N/4}^{kn} \quad (4)$$

$$X(4k+2) = \sum_{n=0}^{N/4-1} [x(n)-x(n+\tfrac{N}{4})+x(n+\tfrac{N}{2})-x(n+\tfrac{3N}{4})]W_N^{2n} W_{N/4}^{kn} \quad (5)$$

$$X(4k+3) = \sum_{n=0}^{N/4-1} [x(n)+jx(n+\tfrac{N}{4})-x(n+\tfrac{N}{2})-jx(n+\tfrac{3N}{4})]W_N^{3n} W_{N/4}^{kn} \quad (6)$$

Table 1: Dragonfly (butterfly) equations

0	$(a_r + c_r) + (b_r + d_r) + j((a_i + c_i) + (b_i + d_i))$
1	$(a_r - c_r) + (b_i - d_i) + j((a_i - c_i) - (b_r - d_r))$
2	$(a_r + c_r) - (b_r + d_r) + j((a_i + c_i) - (b_i + d_i))$
3	$(a_r - c_r) - (b_i - d_i) + j((a_i - c_i) + (b_r - d_r))$

For k=0, 1, 2, 3 we get 16 equations for computing 16 points[4]. The Radix-4 FFT takes only 64 multiplications and 192 additions for computations. The flow graph of 16-point FFT can be seen in Fig. 1. The number inside the open circle as shown in Fig. 1 represents equations as in Table 1 which is used for

Corresponding Author: Prof. S. K. Srivatsa, Department of Electronics Engineering,, Madras Institute of Technology, Anna University, Chennai-44, India

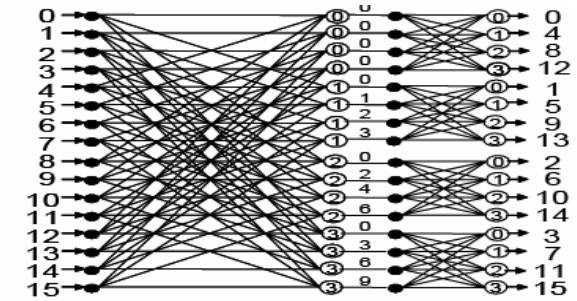

Fig. 1: Signal flow graph of a radix-4 16-point FFT (DIF algorithm)

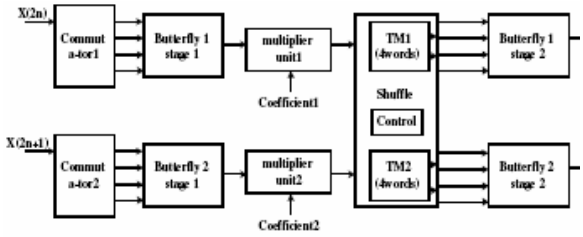

Fig. 2: Block diagram of 16-point 2-parallel-pipelined FFT architecture

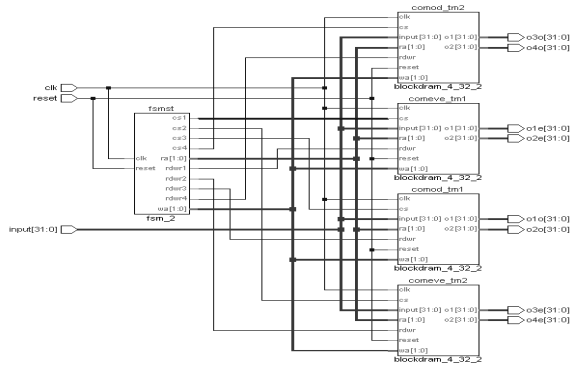

Fig. 3: The RTL block diagram of commutator

computing butterfly stage. Where a_r, b_r, c_r, d_r represents the real value and a_i, b_i, c_i, d_i represents the imaginary value as the input is represented in complex value.

The number outside the open circle is the twiddle factor used. Two radix-4 PEs in each stage of the 16-point pipelined FFT are allocated. The structure is shown in Fig. 2, is 2-parallel-pipelined FFT. It can achieve double the throughput, compared to the pipelined FFT at the same operation frequency. As shown in Fig. 2, the input data are separated into two streams as even and odd and sent to two commutators in stage1. The 4 outputs from two commutators are fed into each simplified butterfly unit. The butterfly unit computes the four equations given in Table 1 in a clock cycle. The coefficients are divided into two responding sections, in terms of even and odd. Two coefficient sections are fed into two complex multipliers, respectively.

In[3], as shown in Fig. 2 a shuffle unit is needed in Stage2 to implement the interstage data shuffle. Instead of that here registers are used to store the intermediate values. These values are fed to butterfly stage 2. The output is stored in same commutators.

Low power techniques

DR commutator: For the commutator, previous implementation approaches include shift register architecture (SR)[5], conventional dual port RAM architecture (DR)[2]. In this paper, a new architecture based on dual port RAM (DR) with no switching activity (No shifting of data's) is used. Dual Port RAM is reduced to four from six as in[2] and with no MUX. The RTL block diagram of commutator (even and odd) with FSM is depicted in Fig. 3 which reduces the power and area when compared with[2,6].

Low power butterfly: The butterfly operation is the heart of the FFT algorithm. It takes data words from memory and computes the FFT. Low Power Butterfly (LB) architecture is employed to replace the conventional butterfly based on adder/subtracters.

Due to the 2's complement arithmetic no separate subtracter is needed for Subtraction. In real-time implementation imaginary part is zero. In Table 1 when '0' is substituted for ai, c_i, b_i and d_i. We get simplified equations as in Table 2.

Table 2: For butterfly stage 1

0	$(a_r + c_r) + (b_r + d_r) + j\,0$
1	$(a_r - c_r) - j\,(b_r - d_r)$
2	$(a_r + c_r) - (b_r + d_r) + j\,0$
3	$(a_r - c_r) + j\,(b_r - d_r)$

In[3] 8 clock cycles is needed for Butterfly stage1 as they are computing one equation at a time. As we are computing four equations at a time 2 clock cycles is enough saving 6 clock cycles which increase the performance as well comparing[3]. Equations in Table 1 are used for computing Butterfly stage 2. The block diagram for butterfly stage 1 and stage 2 are shown in Fig. 4 and 5 respectively.

Multiplier-less unit: In synthesizing DSP architectures, it is important to minimize the silicon area of the integrated circuits, which is achieved by reducing the number of functional units (such as adders and multipliers), registers, multiplexers and interconnection wires. In FFTs, the conventional complex multiplier

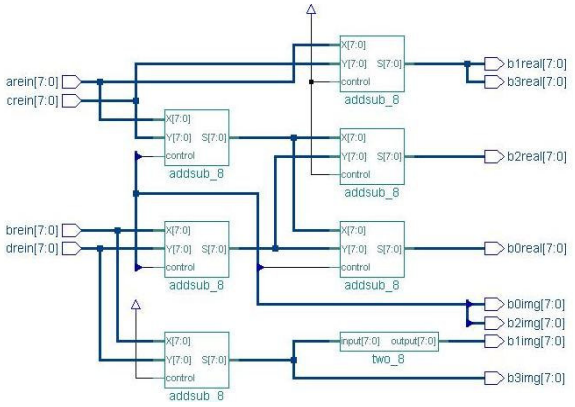

Fig. 4: Butterfly block diagram for stage 1

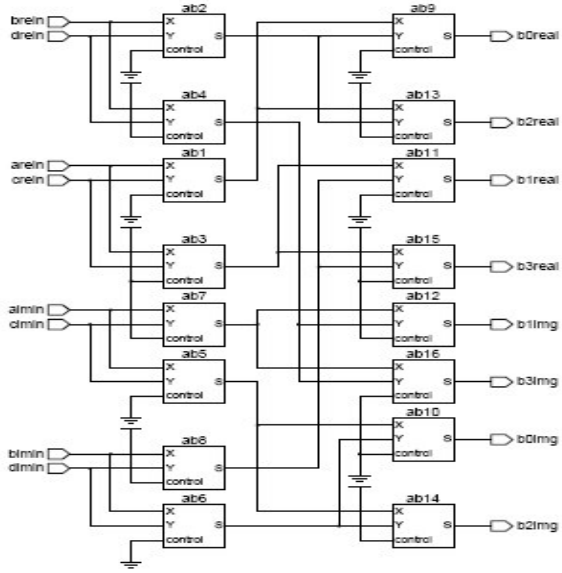

Fig. 5: Butterfly block diagram for stage 2

consists of four real multipliers, one adder and one subtracter. However, since coefficients for all stages can be pre-computed, we can apply shift and addition operations with common sub expression sharing to replace those complex multiplications which reduce area as well as power[7].

For example, the number of coefficients for the first stage of 16-point FFTs is 16. These coefficients are shown in Table 3. The multiplier-less unit as shown in Fig. 10 consist of shift and addition operations with common sub expression sharing to replace complex multiplications. A close observation reveals that the seven coefficients (7fff, 0000) and (0000, 8000) are the trivial coefficients which are the quantized representation for (1, 0) and (0,-1) in 16-bit two's complement format respectively.

Table 3: The coefficients for 16-point

Coefficient Sequence m1 = 0,1	Original quantized coefficient	Coefficient sequence m1 = 2,3	Original quantized coefficient
W_0	7fff,0000	W_0	7fff,0000
W_0	7fff,0000	W_2	5a82,a57d
W_0	7fff,0000	W_4	0000,8000
W_0	7fff,0000	W_6	a57d,a57d
W_0	7fff,0000	W_0	7fff,0000
W_1	7641,cf04	W_3	30fb,89be
W_2	5a82,a57d	W_6	a57d,a57d
W_3	30fb,89be	W_9	89be,30fb

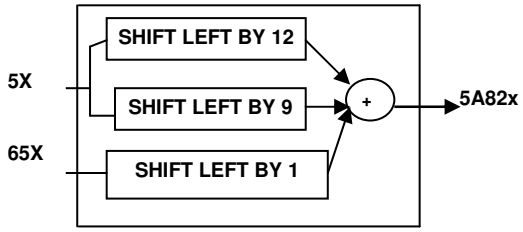

Fig. 6: Constant 5A82X

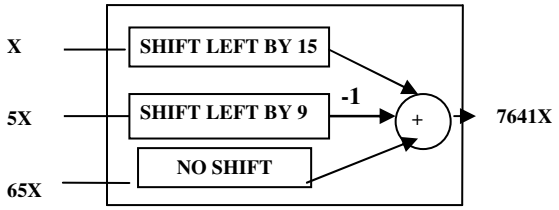

Fig. 7: Constant 7641X

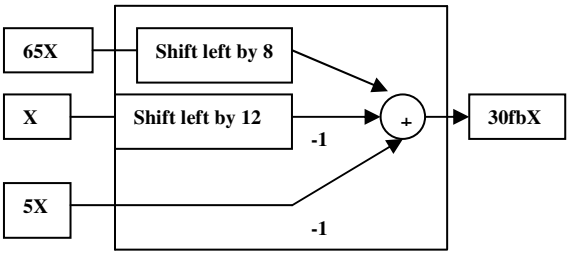

Fig. 8: Constant 30fbX

In each set, the first entry corresponds to the cosine function (the real part, Wr) and the second one corresponds to the sine function (the imaginary part, Wi). For the trivial coefficients (7fff, 0000) and (0000, 8000), the complex multiplication is not necessary. Data can directly pass through the multiplier unit without any multiplication, when data is multiplied with (7fff, 0000). Only an additional unit, which swaps the real and imaginary parts of input data and inverts the imaginary

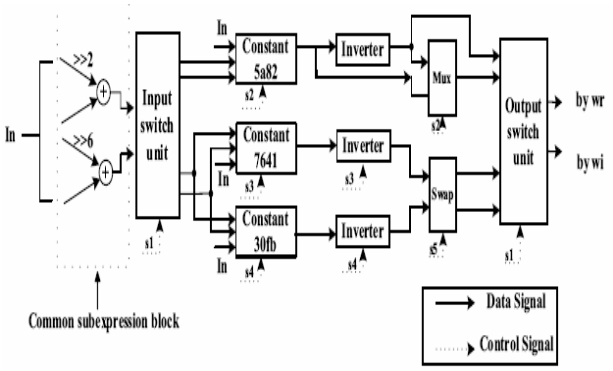

Fig. 9: Block diagram of shift-and-addition module

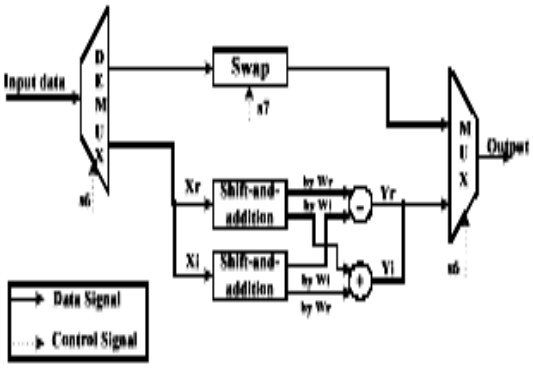

Fig. 10: Block diagram of the Multiplier-less unit

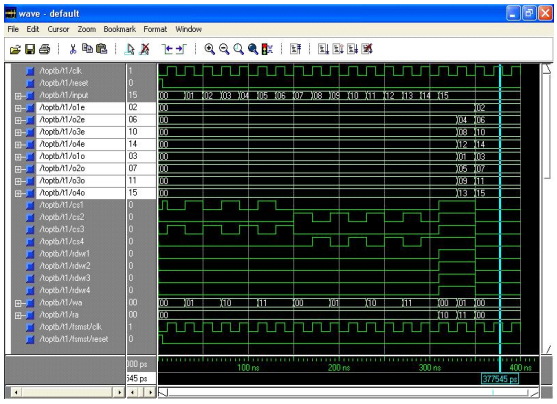

Fig. 11: Simulation results for commutator block

part is needed for those data (0000, 8000).The rest of the coefficients can be represented by three constants (7641, 5a82 and 30fb). For example, a multiplication with the constant a57d could be realized by first multiplying the data with 5a82 and then two's complementing the result. The other two constants (89be and cf04) can be realized in a similar manner, using constants 7641 and 30fb respectively. 5a82 is represented in two's complement format, 7641 and 30fb are represented in Canonical Signed-Digit (CSD) format: 5a82 (0101101010000010), 7641 (1000-10-001000001) and 30fb (010-1000100000-10-1). We can use shifters and adders based on the three constants to carry out those nontrivial complex multiplications as shown below:

5a82X = 5X << 12 + 5X << 9 + 65X <<1

7641X = X << 15 + 65X – 5X <<9

30fbX = 65X << 8 – X << 12 – 5X

Where X means input data. The shift and addition module for the constant 5A82X, 7641X, 30fbX, are shown in Fig. 6-8, respectively.

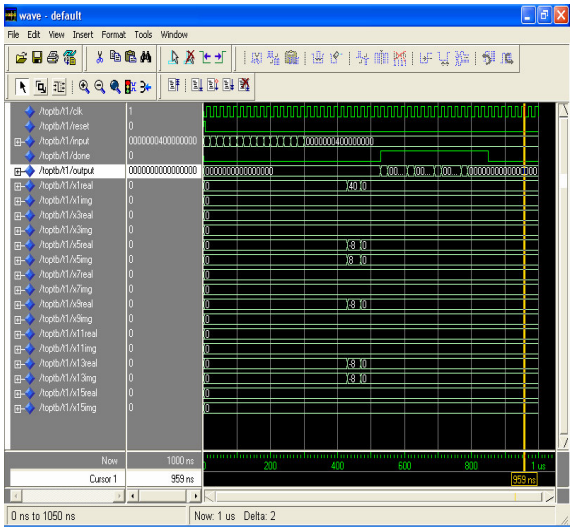

Fig .12: Simulation results for FFT module

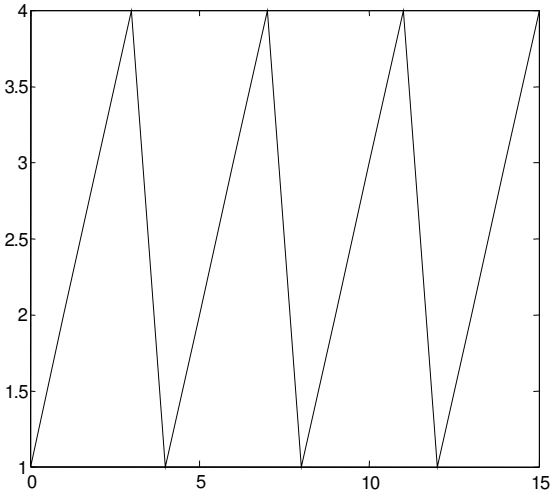

Fig. 13: Plotted input samples (fixed point)

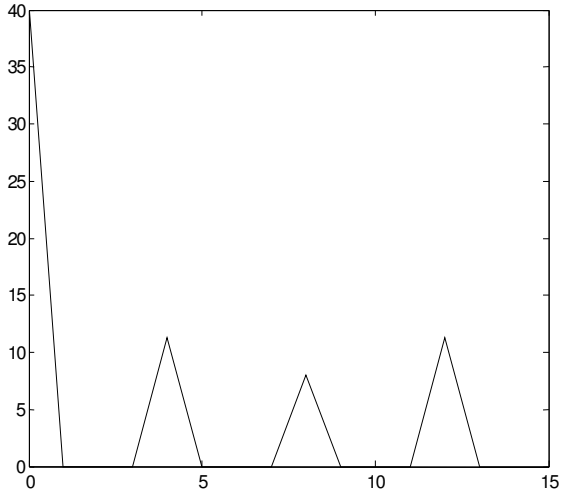

Fig. 14: Plotted output samples (fixed point)

Table 4: Operation required before common sub expression

Operation	5a82x	7641x	30fbx
Addition	5	2	2
Subtraction	0	2	3
Shift	6	4	4

Table 5: Operation required after common sub expression

Operation	5a82x	7641x	30fbx
Addition	2	1	0
Subtraction	0	1	2
Shift	3	3	2

The common sub expressions for the three constants are 101 (5) and 1000001 (65). The operation required before common sub expression block and after common sub expression block is shown in Table 4 and 5. Figure 9 shows the shift-and-addition module for the three constants in the multiplier-less unit. Totally, 11 adders are used to compose the shift-and-addition module. In the multiplier-less unit, 22 adders substitute the four real multipliers in the complex multiplier unit.

RESULTS

Simulation results using modelsim tool: The FFT blocks are simulated and the results are shown below using Modelsim Tool in VHDL. VHDL is a programming language that has been designed and optimized for describing the behavior of digital systems.VHDL has many features appropriate for describing the behavior of electronic components ranging from simple logic gates to complete microprocessors and custom chips.

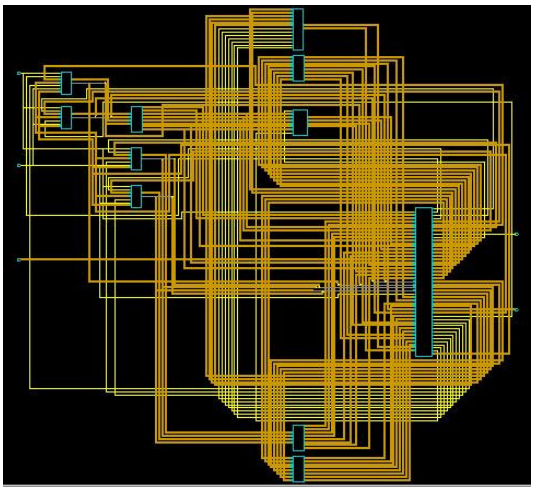

Fig. 15: RTL block diagram of FFT core

Features of VHDL allow electrical aspects of circuit behavior (such as rise and fall times of signals, delays through gates and functional operation) to be precisely described. The resulting VHDL simulation models can then be used as building blocks in larger circuits (using schematics, block diagrams or system-level VHDL descriptions) for the purpose of simulation.

The Simulation result for Commutator Block is shown in Fig. 11. The Simulation result for FFT module is shown in Fig. 12. Only eight cycles are used for transform calculation. Along with data load twenty four cycles are used.

The top module is simulated for 64 bits (complex data) fixed point using 16 point radix_4 DIF FFT algorithm. The given input and corresponding output is as follows.

Input: 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4.

Output : 40 ,0 ,0 , 0 ,-8i +8i ,0 ,0 ,0 ,-8 ,0 ,0 ,0 , -8 -8i, 0, 0, 0.

Mat lab simulation results: The same input is given to mat lab tool for simulation. The input and output are plotted and shown in Fig. 13 and 14, respectively.

Input: 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4,

Output : 40 ,0 ,0 ,0 ,-8i +8i ,0 ,0 ,0 ,-8 ,0 ,0 ,0 , -8 -8i, 0, 0, 0

Synthesis results: The proposed project 16-point FFTs are synthesized at 1.2ns clock cycles to maximize timing slack, using cadence RTL Compiler targeting the TSMC 0.18μ CMOS technology library. The RTL block diagram for complete FFT module is shown in Fig. 15.

Table 6: Power report for FFT core

Cells	12771
Leakage(nw)	2130.107
Internal (nw)	11951828.069
Net (nw)	3208335.885
Switching (nw)	15160163.954
Total (mw)	30.322458

Table 7: Power report for different modules

Unit (mw)	Conventional	1	proposed
Butterfly1	9.530	5.132	1.796(4 outputs)
Butterfly2	8.446	4.552	5.346(4 outputs)
Commutator	18.704	13.004	5.828
Total	63.243	40.908	30.32(64 bits)

Table 8: Timing Report (10^6 sample/sec)

16-point FFT (fixed)	Sample rate(MHz)
Proposed scheme	833.333

Power report: RTL Compiler was used to evaluate power for 16-point FFT at 1.2ns clock cycle. Table 6 shows the power report for FFT core. Table 7 provides power information regarding to main modules for each implementation comparing[7].

The Fig. 16 and 17 gives clear picture about the power usage for instance and Net. The power achievement in this work is 30.3mW which is 52% reduction comparing[7].

Timing analysis: The proposed project 16-point FFTs are synthesized at 1.2ns clock cycles to maximize timing

Table 9: FFT module architecture details

Points	16
Width	64(32+32)bits
Engine Architecture	Single Output
Number of Engines	4
F_{MAX} (MHz)	833.333
Transform Calculation Time (ns)	
Cycles/Time	8 & 9.6 ns
Data Load & transform Calculation (ns)	
Cycles/ Time	24 & 28.8 ns

Table 10: Area report

Instance	Gates	cells	Cell Area (μm^2)
Top	39630	12773	395479.0

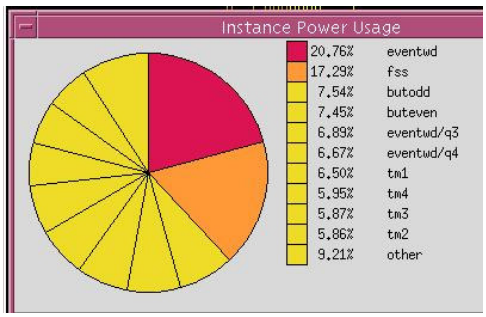

Fig. 16: Instance power usage from cadence tool

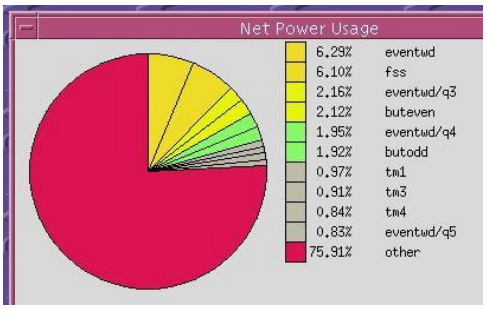

Fig. 17: Net power usage from cadence tool

slack, using cadence RTL Compiler targeting the TSMC 0.18μ CMOS technology library. The FFT IP core achieve up to 833.333 MHz sample rate which is.3.3 times greater than previous paper which work for up to 250 MHz sample rate[3] as shown in Table 8.

FFT core details: The FFT core computes for 16 points with the speed of 833.333 MHz. Table .9 shows the detailed information about the FFT architecture.

Area report: The area report is shown in Table 10. The design was optimized for area and the size of the synthesized Netlist was 39630 gates, 12773 cells and area 395479.0 μm^2.

Features of FFT core:

* High-performance 16-point Complex FFT
* Two's Complement Arithmetic
* Flexible I/O And Memory Configurations
* Naturally Ordered Input And Output Data
* In Place Algorithm
* Parallel Pipelined Processor
* Maximum Speed Up To 833.33 MHz
* Low Power Processor
* Light Weight Processor

CONCLUSION

In this project a parallel pipelined architecture for 16 point radix-4 DIF FFT in fixed point representation is proposed and implemented. Several novel low power techniques: multiplier-less, DR commutator and LB butterfly are implemented. Based on the combination of above mentioned techniques low power can be achieved without transferring data between RAMs and also through maintaining the unused outputs of RAMs at their previous values in the commutator block.

The commutator block in this scheme achieved only 5.828 mw which is reduced to half compared to

previous work (13mw power). This commutator reduces the number of write operations to memory blocks. Low power FFT processor is implemented by using multiplier less (shift add) approach for multiplying twiddle coefficient.

This project presents a novel multiplier-less parallel pipelined FFT processor architecture suitable for shorter FFTs. This design approach can also be applied for the last stages of longer FFTs. The multiplier-less architecture employs the minimum number of shift and addition operations to realize the complex multiplications. This reduces the power consumption of the multiplier by half.

The parameterization impact on power /speed performance has been compared. Up to 52% power savings were achieved, as compared to 16-point R4SDC pipelined FFTs. Previous papers have implemented for 32 bits (one word) complex data whereas our scheme done for 64 bits complex data and achieved 30.32 mw power. These IP cores can also achieve up to 833 MHz sample rate which is 3.3 times greater than[2] which run for 250 MHz.

REFERENCES

1. Cooley, J.W. and J.W. Tukey, 1965. An algorithm for the machine computation of the complex fourier series. Math. Computation, 19: 297-301.
2. Han, W., T. Arslan, A.T. Erdogan and M. Hasan, 2005. Low power commutator for pipelined FFT processors. IEEE, pp: 5274-5277.
3. Han, W., T. Arslan, A.T. Erdogan and M. Hasan, 2005. The development of high performance FFT IP cores through hybrid low power algorithmic methodology. IEEE, pp: 549-552.
4. Rabiner, L.R. and B. Gold, 1975. Theory and Application of Digital Signal Processing. Prentice-Hall.
5. John, G.P. and D.G. Manolakis, 1988. Introduction to Digital Signal Processing. Mac Millian.
6. Hasan, M. and T. Arslan, 2003, 2003. A triple port RAM based low power commutator architecture for a pipelined FFT processor. Circuits and Systems, ISCAS'03. Proc. Intl. Symp., 5: V-353 - V-356.
7. Han, W., T. Arslan, A.T. Erdogan and M. Hasan, 2004. A novel low power pipelined FFT based on sub expression sharing for wireless LAN applications. IEEE Workshop on Signal Processing Systems, pp: 83-88.