



PERFORMANCE ANALYSIS OF AN ENERGY EFFICIENT FFT PROCESSOR USING 32nm CMOS TECHNOLOGY

V. Santhiya and N. Mathan

Department of Electronics and Communication Engineering, Sathyabama University, Chennai, Tamilnadu, India

E-Mail: santhiyavijayan@gmail.com

ABSTRACT

This paper presents an energy-efficient Fast Fourier Transform (FFT) processor which meets the requirements of DSP applications. Fast Fourier Transform is one of the widely used digital signals processing (DSP) algorithms which analysis the signal in its frequency domain. Modified DOMS-FF (Duration Observation Master Slave- Flip Flop) is introduced to reduce the power dissipation and makes the computation of the result much faster than the existing system. The goal of this work is to get less area and energy efficient FFT processor with build in all requirements necessary for DSP applications.

Keywords: Fast Fourier Transforms (FFT), DFT, digital signal processing (DSP), frequency, radix-2, DOMS-FF.

INTRODUCTION

Communication market faces multiple new standards and strong competition. For this reason, several systems require new embedded processors. These embedded processors can either be general purpose, such as digital signal processors and microcontrollers or application specific [10]. The Fast Fourier Transform (FFT) is mainly used as the basic component of all Digital Signal Processing systems. Frequency domain signals are analyzed by using FFT [6]. Fast Fourier Transform is used to convert time domain into frequency domain.

Most of the fields make use of discrete and digital data. Thus the determination of Fourier Transform of discrete signal is of more important and therefore this transform is called DFT (Discrete Fourier Transform) [10]. FFT is an efficient algorithm to compute Discrete Fourier Transform.

Fast Fourier Transform

The fast Fourier Transform (FFT) is a widely used algorithm to computes the discrete Fourier Transform (DFT). The complex value sequence of $X(n)$ is given below [12].

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}, \text{ where } 0 \leq k \leq N-1 \quad (1)$$

Where $X(n)$ is the DFT sequence over length N

The equation (1) can be represented by

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi nk/N}$$

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

Here W_N represents the twiddle factor, which is the N th root of unity and expressed as

$$W_N = e^{-j2\pi/N} \quad (3)$$

Fast Fourier Transform is a method to compute the same results in $O(N \log N)$ operations. More precisely, all known Fast Fourier Transform algorithms need $O(N \log N)$ operations (technically, O only denotes an upper bound), although there is no well known algorithm with less complexity when compare to Fast Fourier Transform.

$x(k)$ and $x(n)$ are frequency and time-domain sequence. To avoid the direct implementation of the equation (1), the FFT algorithm factorizes a DFT recursively from large point into many small point DFT in order to reduce the overall operation time and complexity. FFT algorithm is classified in to two types of decompositions. They are Decimation in Time (DIT) and Decimation in Frequency (DIF) FFT [2]. The only difference between Decimation in Time and Decimation in Frequency algorithms is that, DIT starts with bit reverse order input and generates normal order output. Nevertheless the input of the Decimation in Frequency (DIF) starts with normal order and generates reverse order output [2].

Radix-2 Butterfly processor

The radix-2 Butterfly Processor is the basic building block of an FFT. It also used for the graphical representation of data flow during the FFT computation [13]. In this design, a BP that performs a 2-point Discrete Fourier Transform which is referred as a radix-2 BP. The radix number represent the number of inputs and outputs to the Butterfly Processor. The radix-2 has two inputs and



two outputs. The symbol for a radix-2 BP is shown in Figure-1.

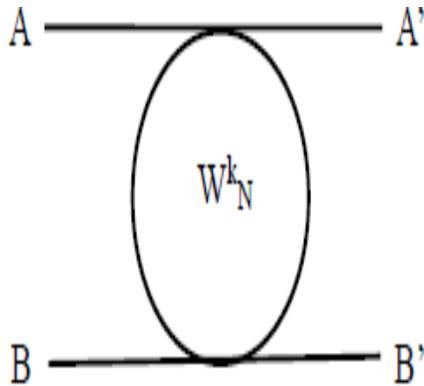


Figure-1. Radix-2 BP.

$$A' = A + W_N^k B \tag{4}$$

$$B' = A - W_N^k B \tag{5}$$

Where A and B are inputs and W_N^k is the twiddle factor, which is also known as the coefficient factor. It is simply a mathematical constant and is given by

$$W_N^k = e^{-j \frac{2\pi k}{N}} \tag{6}$$

RADIX-2 FFT ALGORITHM

The main goal of the FFT algorithms is to compute the DFT of the signal efficiently [7]. In FFT $X(k)$ of a signal $x(n)$ can be calculated using equation (1), where W_N represent the twiddle factor, i is named as imaginary component and N is the number of points of the FFT. For the radix-2 FFT algorithm with decimation in time (DIT), the butterfly allows the computation of complex terms or complex number according to the Figure-2.

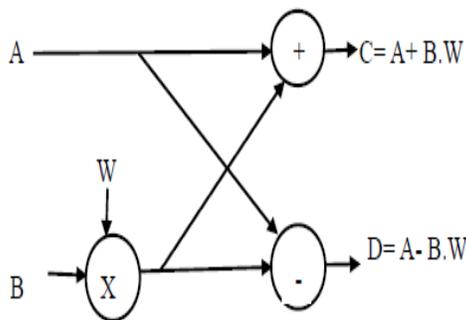


Figure-2. Butterfly structure of the Radix-2 DIT.

The butterfly operation is composed by addition, subtraction and multiplication, where these arithmetic operations involve complex numbers. The radix-2 structure with address compressor shown in the Figure-3 is used to calculate the complex number radix-2 operation. Three inputs are given to the 3 x 2 compressor and it generates two outputs. This basic structure can also be employed as a full adder, where the carry input from the previous compressor block is given as the third input for the second adder compressor block. This structure can be designed using one MUX and two XOR gates to generate Carry output Sum term, as shown in the Figure-4. Two XOR gates of the Sum term calculation give the critical path of the 3 x 2 adder compressor.

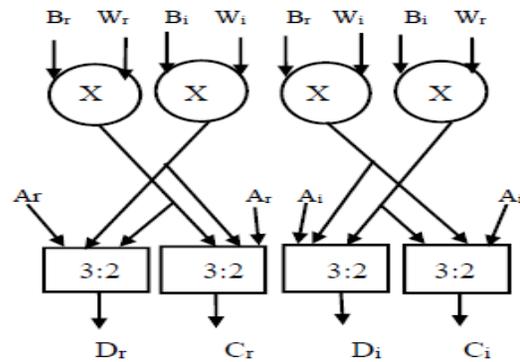


Figure-3. Butterfly structure with adder compressor.

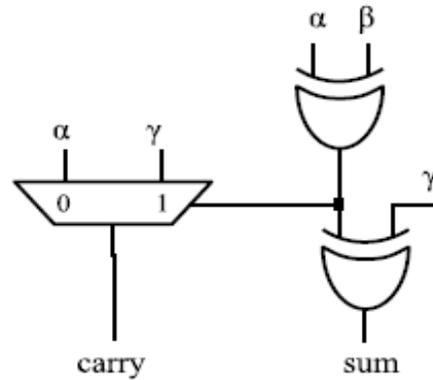


Figure-4. 3:2 address compressor.

2's complement operations are used to implement the negative inputs in the adder compressors. The N number of adder circuit is cascade to for the structures of the butterflies. The gate level representation of 3 x 2 address compressor is shown in the Figure-4.

EXISTING SYSTEM

Fast Fourier Transform plays a major role in signal processing and communication applications. To improve the energy efficiency supply voltage scaling has been extensively used in Fast Fourier Transform



processors. Radix-2 and embedded SRAM memory block are used to verify the output of the FFT cause by SRAM failure.

SRAM memory block is used to store the outputs of all the radix-2 block temporarily and the stored data is read again for the next iteration. Read Only Memory (ROM) is used to storing the twiddle factors required during each FFT calculation, since these values are constant for each FFT of the same point size. FFT processor design is implemented by using HSPICE and the performances of FFT processor are observed.

DOMS-FF

The Duration Observation Master Slave -Flip Flop shown in Figure-5 monitors the duration of the data signal for the master latch. Then, basis on that time, it distinguishes whether the data signal is proper data or a noise pulse.

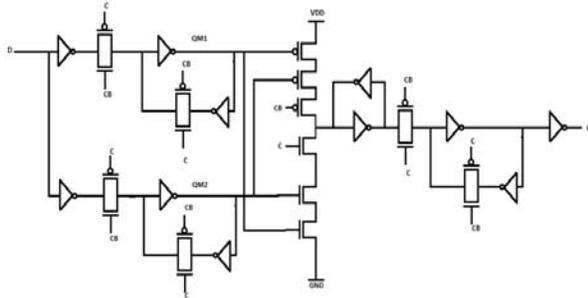


Figure-5. DOMS-FF.

At the first step master latch in the flipflop samples the input signal once at the original clock edge. After that, the input signal is sampled again with the delayed clock edge. If both the sampled values are same then the data signal is consider as a proper signal, if not the data signal is consider as a noise pulse signal. Proper data signals are propagated to the slave latch and the master latch blocks the noise pulse signal. Where C, Cd, Cb represents the clock, clock bar and clock signal with delay.

PROPOSED SYSTEM

Modified DOMS-FF

The second master-slave latch is provided for sampling the data signal at the second sampling time. The second master latch synchronizes with the internal delayed clock signal, which corresponds to the second sampling time. In the modified Duration Observation Master Slave-FF transmission gates are replaced by pass transistors in order to reduce the number of transistors, area and power. The modified DOMS-FF is shown in Figure-6.

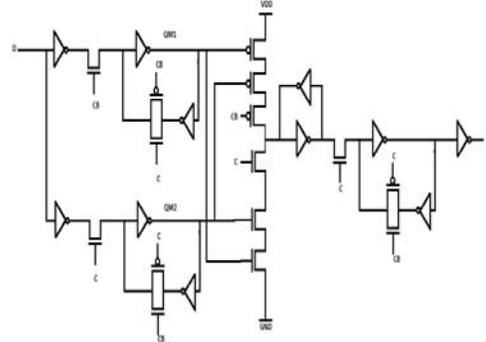


Figure-6. Modified DOMS-FF.

Modified FFT-Processor

FFT processor consist of flip flop, MUX, SRAM and radix-2 blocks. Here, flip flop block is replaced by the modified low power flip flop (modified DOMS-FF), which makes the computation of the result much faster than the existing system. As a result FFT processor can be used for high speed and low power applications. Block diagram for modified FFT processor is shown in the Figure-7.

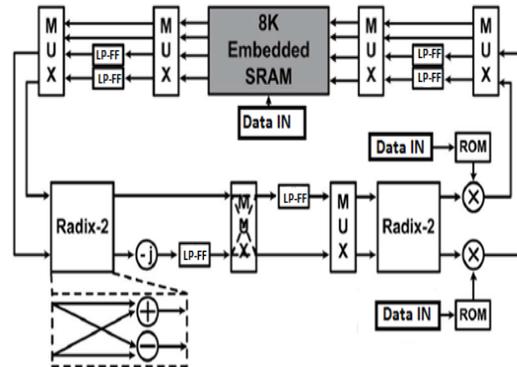


Figure-7. Modified FFT-processor.

Read Only Memory used to store the coefficient factors (or twiddle factor) required during each FFT calculation. Output of the radix-2 blocks is temporarily stored in embedded SRAM memory and the stored date is read for next iteration. Simulation result for modified FFT-Processor is shown in the Figure-11.

TRANSIENT ANALYSIS

The proposed FFT Processor is simulated using HSPICE and when comparing to the existing system 50% of power consumption is reduced. Simulation result show great improvement in terms of power consumption and delay. The simulation result for modified FFT processor and Duration Observation Master Slave-Flip Flop is shown in the Figure-8.



Figure-8 illustrate the waveform representation of modified Duration Observation of Master Slave- Flip Flop at 32nm and 130nm Technology.

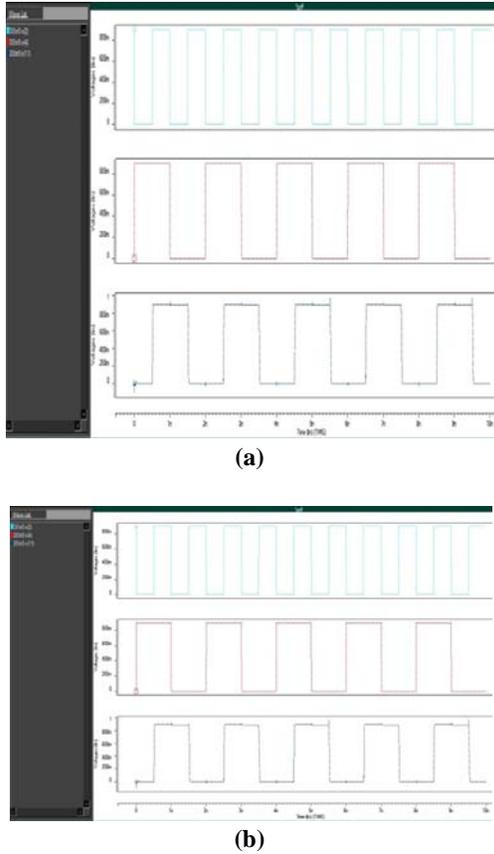


Figure-8. (a) Simulation result for modified DOMS-FF (32nm technology) (b) simulation result for modified DOMS-FF(130nm technology)

Where v(2), v(4), v(11) represent clock, input D and output.

Simulation result for modified FFT Processor at 130nm and 32nm technology is shown in the figure 9.

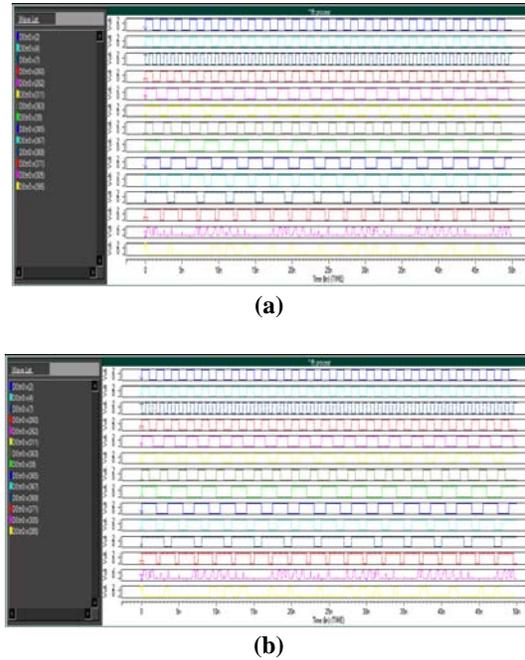


Figure-9. (a) simulation result for modified FFT Processor at 130nm technology (b) simulation result for modified FFT Processor at 32nm technology.

Where node v(2), v(4) represent the input and v(7), v(23), v(260), v(263), v(371), v(105), v(102) v(396) represent the output of each stage.

PERFORMANCE ANALYSIS

The existing and proposed Duration Observation Master Slave-FF are simulated at 32nm and 130nm technology, and numerical result for different technolgs is shown in Table-1.

Table-1. Comparison of existing and proposed DOMS-FF At 130nm and 32nm technology.

		DOMS-FF		
Technology		Avg. power (W)	Delay (pS)	PDP (J)
Existing system	130nm technology	1.3624e-04	40.02	54.52e-16
	32nm technology	8.141e-06	7.43	60.54e-18
Proposed system	130nm technology	1.265e-04	29.932	37.86e-16
	32nm technology	4.529e-06	9.6942	43.90e-18



Table-2 shows the comparison of existing and proposed FFT Processor at 32nm and 130nm technology.

Table-2. Average Power Comparison of Modified FFT Processor at 130nm and 32nm Technology.

FFT Processor				
Technology		Avg. power (W)	Delay (pS)	PDP (J)
Existing System	130nm technology	1.812e-03	49.993	90.58e-15
	32nm technology	6.289e-04	7.8138	49.13e-16
Proposed system	130nm technology	1.364e-04	49.857	68e-16
	32nm technology	5.483e-04	5.0481	27.67e-16

PERFORMANCE ANALYSIS CHART

The average power dissipation of proposed FFT Processor demonstrates a better power reduction when compared to existing FFT Processor.

The graphical representation of existing and modified Duration Observation Master Slave Flip-Flop is shown in Figure-10.

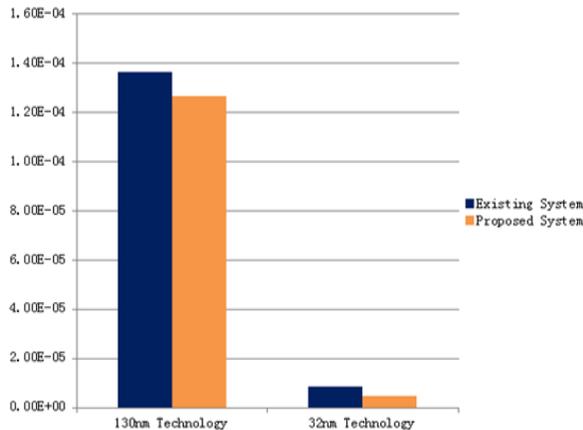


Figure-10. Comparison of DOMS-FF and modified DOMS-FF

Figure-11 illustrate the average power comparison of existing and proposed FFT Processor results graphically.

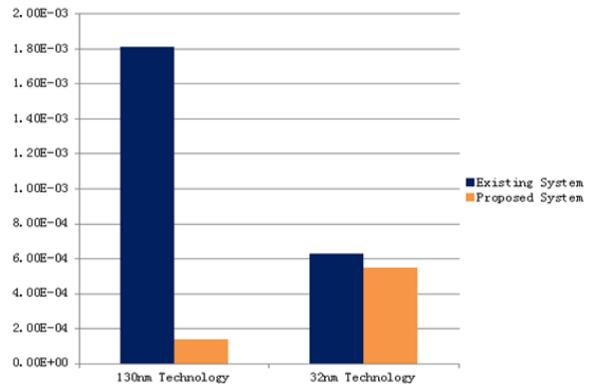


Figure-11. Average power of FFT processor (130nm and 32nm).

CONCLUSIONS

FFT processor consists of flip flop, MUX, SRAM and radix-2 blocks. Here, flip flop block is replaced by the modified DOMS-FF, which makes the computation of the result much faster than the existing system. As a result FFT processor can be used for high speed and low power applications. Twiddle factors are fetched and stored in SRAM memory, stored data is used for next iterations.

REFERENCES

- [1] Akamatsu.H, Satomi. K, Suzuki. T, Yamagami. Y and Yamauchi.H. 2008. A stable 2-port SRAM cell design against simultaneously read/write disturbed accesses. IEEE J. Solid-State Circuits. 43(9): 2109-2119.
- [2] Debalina Ghosh, Depanwita Debnath, Dr. Amlan Chakrabarathi. 2012. FPGA Based Implementation of FFT Processor Using Different Architectre. IJAITI.



www.arpnjournals.com

- [3] G.Purna Chandra Rao, B. Ashok, B. Saritha. 2013. Design and Implantation of High Speed FFT Processor for OFDMA System Using FPGA. 2(7).
- [4] H. Qin *et al.* 2004. SRAM leakage suppression by minimizing standby supply voltage. In Intl. Symposium on Quality Electronic Design, pp. 55-60.
- [5] H. Sorensen, D. Jones, M. Heideman and C. Burrus. 1987. Real-valued fast Fourier transforms algorithms. IEEE Trans. Acoust., Speech Signal Process. 35(6): 849-863.
- [6] Jangwon Park, Jongsun Park, Swarup Bhunia. 2014. VL-ECC: Variable Data-Length Error Correction Code for Embedded Memory in DSP Applications. IEEE Transactions on Circuits and Systems-IIS. 61(2).
- [7] J. Crols and M. Steyaert. 1995. A single-chip 900-MHz CMOS receiver front-end with a high performance low-IF topology. IEEE J. Solid-State Circuits. 30: 1483 -1492.
- [8] J. Lee, H. Lee, S. I. Cho and S. S. Choi. 2006. A high-speed two parallel radix- 2^4 FFT/IFFT processor for MB-OFDM UWB systems. Proc. IEEE Int. Symp. Circuits Syst. pp. 4719-4722.
- [9] J. Caravella. 1997. A Low Voltage SRAM for Embedded Applications. IEEE Jourrial of Solid-state Circuits. 32(3): 428- 432.
- [10] N.Mathan. 2014. CNTFET based Highly Durable Radix-4 Multiplier using an Efficient Hybrid Adder. Biosciences Biotechnology Research Asia. 1(3): 1855-1860.
- [11] Pavan Kumar Jain. 2012. Design of FFT-Processor.
- [12] Qingwang Lu, Xin'an Wang and Jiuchong Niu. 2009. A Low-power Variable-length FFT Processor Base on Radix-24 Algorithm.
- [13] Renu yadav, Mukesh Pathela. 2013. Design and Performance Analysis of FFT in OFDM applications using VHDL. IJEST. 5(10).
- [14] Richard G. Rozier, Fouad E. Kiamilev, Ashok V. Krishnamoorthy. 1996. Design of a Parallel photonic FFT Processor. MPPPI.
- [15] Yukiya Miura, Yoshihiro Ohkawa. 2014. A Noise-tolerant Master-slave Flip-flop. IEEE 20th International On-Line sTesting Symposium (IOLTS).