ISSN 2348 – 8034
Impact Factor- 3.155

GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

FPGA IMPLEMENTATION OF A FFT PROCESSOR USING VEDIC ALGORITHM

Sandesh Deshmukh[*1], Nilesh Shinde[2], Nikhil Shinde[3], Prof. Smita Modi[4]
[*1,2,3,4]Department of E&TC Parvatibai Genba Moze College Of Engineering, Wagholi, Pune

ABSTRACT

Fast Fourier Transform is an essential data processing technique in communication systems and DSP systems. Here, we propose high speed and area efficient 64-point FFT processor using Vedic algorithm. To reduce computational complexity and area, we develop FFT architecture by devising a Radix-4 algorithm and optimizing the realization by Vedic algorithm. Furthermore, it can be used in decimation in frequency (DIF) and decimation in time (DIT) decompositions. Moreover, the design can achieve very high speed, which makes them suitable for the most demanding applications of FFT. Indeed, the proposed Radix-4 Vedic algorithm based architecture requires fewer hardware resources. The synthesis results are same as that of theoretical analysis and it is observed that more than 15% reduction can be achieved in terms of slices count. In addition, the dynamic power consumption can be reduced and speed can be increased by as much as 16% using Vedic algorithm.

Keywords- *FFT, Vedic algorithm, DSP, radix-4*

I. INTRODUCTION

Now a days, one of the important application of FFT is the orthogonal frequency division multiplexing (OFDM)[1]. It is mainly used in wireless local area network (WLAN), digital audio broadcasting (DAB), digital video broadcasting-terrestrial (DVB-T) and digital video broadcasting-handheld (DVB-H). Due to such diverse application of FFT, it is desirable to develop efficient FFT to meet the requirement of various OFDM communication standards[1]. The FFT is a faster version of the Discrete Fourier Transform (DFT) and calculates Discrete Fourier Transform efficiently in our work. By reducing the computational complexity. Memory-based architecture is widely adopted to design an FFT

processor. It consists of butterfly processing element and memory units. It has low power consumption but long latency and low throughput. To improve the efficiency of memory based FFT architecture, radix-4 butterfly processing units along with dual port memory is adopted. Urdhva-Tiryakbhyam Sutra is first applied to the binary number system and is used to develop digital multiplier architecture. This Sutra also shows the effectiveness of reducing the N×N multiplier structure into an efficient 4×4 multiplier structures. This work presents a systematic design methodology for fast and area efficient digital multiplier based on Vedic mathematics.

II. SYSTEM DEVELOPMENT

In order to improve performance, we propose a radix-4 FFT processor architecture that operates on data length of 64-points with minimum area consumption. Fig.1 depicts the block diagram of proposed FFT. The proposed FFT processor is comprised of radix-4 butterfly calculation units implemented using Vedic algorithm, an address generator, memory unit, twiddle factor generator and commutator. Our primary concern is improving the performance of the inefficient computational block of the FFT processor by eliminating the complex critical path components and by using Vedic Algorithm. Vedic mathematics is based on 16 Sutras dealing with various branches of mathematics like arithmetic,algebra, geometry.

- **FFT ALGORITHM**

The DFT of the sequence of N complex of numbers x0,…,xN-1 is defined by:

$^{nk}{}_{N}$ for k=0,1,…,N-1 (1)

[with $W_{N=e}{}^{-j(2\pi/N)}=\cos(2\pi/N)-j\sin(2\pi/N)$]

ISSN 2348 – 8034
Impact Factor- 3.155

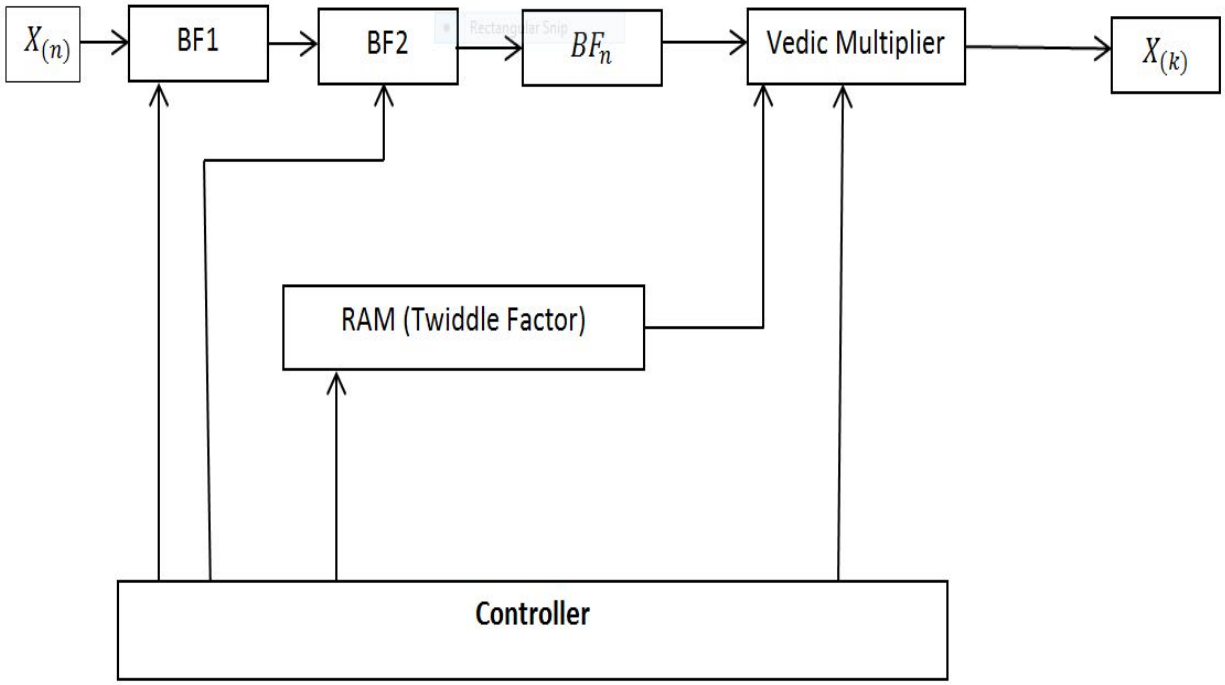

Fig 1. System block diagram of FFT

$W=e^{-j(2\pi/N)}$ indicates the value of coefficients of
the FFT and are often referred as twiddle factors, x(n) is a time sequence and X[k] is a frequency sequence. Direct implementation of (1) yields in large hardware and high complexity. Therefore, to minimize its computational complexity and reduce the hardware cost the fast Fourier transform

- **RADIX-4 ALGORITHM**

An efficient class of radix algorithm for designing FFT is radix-4 algorithm. The complexity of radix-2 FFT algorithm can be further reduced by using radix-4 algorithm. In radix-4 algorithm, there are log4N stages and each stage has N/4 4-point butterflies. The radix-4algorithm operates by decomposing N point input data sequence into N/4 different points. The *N*-point FFT can be decomposed to repeat micro operations called *butterfly* operations. When the dimension of the butterfly is *r*, the FFT operation is called a radix-*r* FFT. For FFT hardware realization, if only one butterfly structure is enforced in the chip, this butterfly unit will execute all the calculations recursively. If parallel and pipeline processing techniques are used, an *N* point radix-*r* FFT can be executed by *(N/r)* logr N clock cycles [18]. This indicates that a radix-4 FFT can be four times faster than a radix-2 FFT. The total number of complex multiplication is (3N/4) and the number of required complex additions is (3N/4). Thus, radix-4 algorithm reduces number complex multiplier. Also, the numbers of stages required are also reduced by half.

- **VEDIC ALGORITHM**

The proposed Vedic multiplier is based on the Vedic multiplication formulae (Sutras). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware.

The proposed Vedic multiplier is based on the "Urdhva Tiryagbhyam" sutra (algorithm). These Sutras have been traditionally used for the multiplication of two numbers in the decimal number system. In this work, we apply the same ideas to the binary number system to make the proposed algorithm compatible with the digital hardware. It is a general multiplication formula applicable to all cases of multiplication. It literally means "Vertically and Crosswise". It is based on a novel concept through which the generation of all partial products can be done with the concurrent

ISSN 2348 – 8034
Impact Factor- 3.155

addition of these partial products. The algorithm can be generalized for n x n bit number. Since the partial products and their sums are calculated in parallel, the multiplier is independent of the clock frequency of the processor. Due to its regular structure, it can be easily layout in microprocessors and designers can easily circumvent these problems to avoid catastrophic device failures.

The processing power of multiplier can easily be increased by increasing the input and output data bus widths since it has a quite a regular structure. Due to its regular structure, it can be easily layout in a silicon chip. The Multiplier based on this sutra has the advantage that as the number of bits increases, gate delay and area increases very slowly as compared to other conventional multipliers.

III. PROPOSED SYSTEM

We have proposed a radix-4 FFT processor architecture that operates on data length of 64-points with minimum area consumption. driver cards are driven by microcontroller. The FFT processor reads and writes from and to the 8 twin port memory banks concurrently. There are 8 read address buses, and 8 write address buses. They have two signals, "Start" and "Busy". The first one enables the processor to process the data and the second one indicates processing of data inputs. The incoming data samples are partitioned into even and odd samples. After the assertion of the "Start"
signal 64 input samples are clocked into the memory bank. A 6-bit counter controls the serial input of the datathe transformer.

A) Radix-4 Butterfly Unit using Vedic Algorithm

To perform 64-point FFT a single 4-point FFT unit is recursively used. This 4-point FFT is designed using high speed radix-4 algorithm. Vedic algorithm is an ancient and well known technique for arithmetic operation. The method which is used for multiplications is 'Urdhva-tiryagbhyam' which means vertical and crosswise. The very first step is to vertically multiply LSB's of two numbers. Carry bit generated due to this is transferred to the next step and the result bit goes to the final result. In second step, it performs crosswise multiplication with adjacent number.

Again, the previous carry bit is added here to yield final result and carry bit is propagated to next step. In third step, the algorithm executes vertical and crosswise multiplication and previous carry is added to give final product. In this way, this algorithm performs multiplication of two given numbers in vertical and crosswise manner until left with only MSB bits. The proposed FFT utilizes Urdhva-tiryagbhyam method of Vedic algorithm to perform complex twiddle factor multiplications.

B) RAM

A memory unit is composed of 8 dual-port memory banks which facilitate 8-way parallel data access. Each memory bank is 8-bit wide. From memory perceptive, in-place memory scheme is adopted so as to minimize the hardware resources and speed up the memory access time. For radix-*r* FFT, r banks of memory are needed to store data, and each memory bank could be dual-port memory.With "in-place" strategy, the *r* outputs of the butterfly can be written back to the same memory locations of the r inputs, and replace the old data.

C) HARDWARE ARCHITECTURE

The main advantage of the vedic multiplication algorithm (Urdhva-Tiryak Sutra) stems from the fact that it can be easily realized in hardware. The hardware realization of a 4-bit multiplier using this Sutra is shown in Fig. 4. This hardware design is very similar to that of the famous array multiplier
where an array of adders is requited to arrive at the final product. All the partial products are calculated in parallel and the delay associated is mainly the time taken by the carry to propagate through the adders which form the multiplication array.

IV. CONCLUSION

The proposed architecture of FFT processor was modeled using VHDL language. The entire architecture was synthesized and implemented using Xilinx ISE v13.1.The functionality was tested by creating test bench waveform and used in behavioral and post layout simulations.

The hardware architecture of the Vedic multiplier is also depicted and is found to be very similar to that of the so called array multiplier. This is just one of the many possible applications of the Vedic Mathematics to Engineering and some serious efforts are required to fully utilize the potential of this interesting field for the advancement of

ISSN 2348 – 8034
Impact Factor- 3.155

Engineering and Technology. Although, Vedic mathematics provides many interesting Sutras, but their application to the field of engineering is not
yet fully studied. Knowingly or unknowingly we always use Vedic Sutras in everyday world of technology

REFERENCES

1. *More T.V. , Panat A.R.* ***"FPGA implementation of FFT using Vedic algorithm"****, Computationalintelligence and Computing Research (ICCIC), pp-1-5, 2013.*
2. *Harpreet Singh Dhillon, Abhijit Mitra, "A Digital Multiplier Architecture using UrdhvaTirkabhyam Sutra of Vedic Mathematics", IITG , pp-1-4, 2010.*
3. *Asmita Haveliya, "FPGA implementation of a Vedic convolution algorithm", International Journal of Engineering Research and Applications, Vol. 2, Issue 1, pp.678-684, Jan-Feb 2012,*
4. *www.xilinx.com, www.wikipedia.com*