DESIGN OF LOW POWER PIPELINED FFT PROCESSOR

1 N. Dakshinavarthini 2Mrs. G. Pratheepa 3Mr. A. Manimaran
1:PG Scholar 2: Assistant Professor 3: Associate Professor
Karpaga Vinayaga College of Engineering and Technology
GST Road, Chinna kolambakkam, Maduranthagam (T.K.)., Kancheepuram-603308
1 Email id: dakshna1206@gmail.com 2 Email id: prathipa_ragu@rediffmail.com 3 Email id: lims_01@rediffmail.com

ABSTRACT
Discrete Fourier transform (DFT) is a very important technique in modern digital signal processing (DSP) and telecommunications. The Fast Fourier transform is a highly efficient procedure for computing the DFT of a finite series and requires less number of computations than that of direct evaluation of DFT. In this work the efficient implementation of a pipeline FFT processor is presented. FFT reduces the computations by taking advantage of the fact that the calculation of the coefficients of the DFT can be carried out iteratively. Due to this, FFT computation technique is used in digital spectral analysis, filter simulation, autocorrelation and pattern recognition. The proposed architecture design adopts a single path delay feedback (SDF) and this SDF pipeline requires less memory space and its multiplication utilization is less. Such implementations are very much use full in OFDM architecture advantageous to low power design. Read only memories are used to store twiddle factors since most of the complexities are arise from multiplication of twiddle factor only. In order to achieve a ROM less FFT processor, the proposed architecture uses a reconfigurable complex multiplier and bit parallel multiplier. By the use of symmetry property of twiddle factor low power will be achieved.

Key Words: SDF, DFT, ROM less Structure

I. INTRODUCTION

The Fast Fourier Transform (FFT) processing is an important aspect of many Digital Signal Processing applications and systems. Real-time or fast execution is an important criterion for many applications. For example, wide-band Orthogonal Frequency Division Modulation (OFDM) systems, biomedical instrumentation and radar usage in military domain are some of the applications requiring high-speed and large-point FFT systems as one of their key components and it is a fastest method for finding DFT. Memory based architecture is commonly used to design FFT processor because of its low hardware cost and low power consumption. But it suffer problem of long latency and long throughput. The main contribution this work is demonstration of low power memory less pipeline FFT processor.

The Discrete Fourier Transform is a procedure, or process, that can analyze the data points of a “digitized” time domain function to determine a series of sinusoids which, when summed together, reproduce the data points of the original function.

Mathematically denoted as
X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk}, \quad 0 \leq k \leq N-1

Where \( W_N^{nk} \) called as twiddle factor

The FFT can make the simple stand most effective optimizations when the number of samples to be transformed is an exact power of two, for which it can eliminate many unnecessary operations. The results of the FFT are the same as with the DFT; the only difference is that the algorithm usually runs much faster and with less resources. In algorithmic terms, the DFT takes \( O(N^2) \) arithmetic operations to be computed, where as the FFT takes \( O(N\log_2 N) \) arithmetic operations. There are two basic DFT algorithm decimation in time and decimation in frequency.

This Paper sectioned as V Parts. The II Section relates the previous work carried on designing of FFT describes FFT structure and Section –III Consists of simulation results followed by conclusion and future enhancements

**II. FFT ARCHITECTURE**

a) FFT : Before the development of FFT architecture it was difficult to process the data for digital signal processing circuits. The different architecture were developed to improve the performance of DFT in terms of FFT. But each FFT architecture has a drawback that it need More ROM to store the twiddle. For each cycle of operation the number of hardware components increased and the power consumption problem arises and Mainly the FFT architecture of olden days does not follows Pipelined mechanism and the radix level power is very High, so this kind of circuits take more latency and produce the throughput after the long time. The FFT architecture consists of sequential processor, pipelined processor and parallel interactive processor. The basic sequential processor consists of a processing element(PE) that can compute a butterfly. The same memory can be used to store the data, intermediate results and the twiddle factors.

The amount of hardware involved is very small and it takes \( (N/2)\log_2 N \) sequential operations to compute the FFT. To improve the performance of the sequential processor, parallelism can be introduced by using a separate arithmetic unit for each stage of the FFT.

This increases the throughput by a factor of \( \log_2 N \) when the different units are pipelined. This architecture is also known as cascaded FFT architecture and will be used in our proposed design. By adding more processing elements to the processor in each sequential pipeline stage, performance can be improved even further. The butterflies can then be computed in parallel in any stage. The total execution time for the parallel iterative processor is \( \log_2 N \) cycles. Pipeline FFT Architecture consists of a series of computational blocks each composed of delay lines, co efficient storage, commutator, multipliers, and adders. Multi-path Delay Commutator is the most classical approach for pipeline implementation of radix-2 FFT algorithm in that input sequence has been broken into two parallel data stream and flowing forward, with correct distance between the data elements and entering to the butterfly scheduled by proper delays.

![Fig 1: R2MDC(N=16)](image)

Radix-2 Single-path Delay Feedback uses more registers to storing one output of each butterfly in feedback shift registers. A single data stream goes through the multiplier at every stage. It has same number of butterfly units and multipliers as in R2MDC approach, but with reduced memory requirement and \( N - 1 \) registers. Its data throughput is \( (1/x) \) times that of the corresponding RxMDC architecture.

![Fig 2: R2SDF(N=16)](image)

b) Drawbacks :
i. More ROM utilized to store each twiddle factor.
ii. More Latency and delayed Throughput.
iii. No pipelined architecture which increases the number of components.
iv. No reconfigurable memory.

**III. PIPELINED FFT LOW POWER PROCESSOR**

A reconfigurable complex constant multiplier is used to eliminate the twiddle factor ROM. This component plays an important role in reducing area and hardware cost. The architecture consists of processing elements which is divided into three stages and each stage is responsible to perform 8 point radix-2 FFT algorithms.

a) Processing Elements

The proposed architecture is composed of three different types of processing elements (PEs), a complex constant multiplier and delay-line (DL) buffers and using a single path delay feedback pipeline architecture.

In the PE1 stage, the calculation is more complex than the PE2 stage, which is responsible for computing the multiplications by $-j$, $W_{N/8}^N$, and $W_{N/4}^{2N/8}$, respectively. Since, $W_{N/8}^N = -jW_{N/8}^N$, it can be given by either the multiplication by $W_{N/8}^N$ first and then the multiplication by $-j$ or the reverse of the previous calculation. Hence, the designed hardware utilizes this kind of cascaded calculation and multiplexers to realize all the necessary calculations of the PE1 stage.

---

**Fig 3: Proposed radix-4 64-point pipeline FFT/IFFT processor**

PE3 stage is used to implement a simple radix-2 butterfly structure and also serves as the sub modules of the PE2 and PE1 stages.

**Fig 4: Proposed PE3 structure**

As for the PE2 stage, it is required to compute the multiplication by $-j$ or 1. Note that the multiplication by-1 in Fig. 4.3 is practically to take the 2’s complement of its input value.

**Fig 5: Proposed PE2 structure**

**Fig 6: Proposed PE1 structure**

The multiplication by $\frac{1}{\sqrt{2}}$ can employ a bit parallel multiplier to replace the word length multiplier and square root evaluation for chip area reduction. The realization of complex multiplication by $W_{N/8}^{N/2}$ using a radix-2
butterfly structure with its both outputs commonly multiplied by $\frac{1}{\sqrt{2}}$. Based on this reconfigurable low-complexity complex constant multiplier for computing $w_{e_4}^k$ is designed. This structure of this complex multiplier also adopts a cascaded scheme to achieve low-cost hardware. Meaning of two input signal (Iin and Iout) and two output signals (Qint and Qout) are the same as the signals in the PE1 stage.

![Fig 7: Reconfigurable complex constant multiplier](image)

The proposed Architecture works based on two widely used algorithms. It works on radix-8 which is sub classified as two radix-4 or four radix-2 structure. Here the pipelined architecture is proposed in order to reduce the delay and reconfiguration multipliers are used to reduce the ROM usage.

a). Radix-4/8 Algorithms

The proposed system is designed based on radix-8. The Radix-22 algorithm has the same multiplicative complexity of Radix-4 algorithms but has a signal flow graph similar to the Radix-2 algorithm.

It is mathematically expressed as shown. Grouping 4 samples together gives the Radix-4 Algorithm.

$$X(8k+i) = \sum_{k=0}^{\frac{N}{4}-1} (x(n) + x(n + \frac{N}{4}) w_4^k + x(n + \frac{N}{4}) w_4^k w_4^{2i} + x(n + \frac{3N}{4}) w_4^{2i} + x(n + \frac{3N}{4}) w_4^{2i} w_4^{2i})$$

Radix-8 Algorithm is given by Grouping 8 samples together gives the Radix-8 Algorithm. Mathematically, It is expressed as shown in

![Fig 8: Butterfly for Radix8 Algorithm](image)

IV. SIMULATION RESULTS

The proposed FFT architecture uses three stages of processing elements. The entire structure is simulated in Model simulator which creates the way to run top level entity or modules. Here the radix-8 structure is divided into sub modules and each module is simulated and every program work together as top level modeling. And the verilog Language is used since it has industry standards. The simulation result for top level module is shown in Fig 9. The tool Model simulator is used here to run the program in simulator mode. So the Xilinx ISE 11.1 is used to analyze the power level and area. The reduction in area can be found by calculating the no of gates or slices used by the top level module, the summary of synthesis report is shown in Fig 10.
The Register transfer level is to create high-level representations of a circuit, from which lower-level representations and ultimately actual wiring can be derived. This RTL is used to identify the Components involved or present in the processing elements. An RTL description is usually converted to a gate-level description of the circuit by a logic synthesis tool. The synthesis results are then used by placement and routing tools to create a physical layout. Logic simulation tools may use a design's RTL description to verify its correctness. The RTL view of top level module is shown in Fig 11.

**V. CONCLUSION & FUTURE ENHANCEMENT**

By using an FFT processor with less ROM Structure its observed that the area is
reduces and also increase the speed in terms of reducing the processing speed. The proposed architecture is expected to be provided low complexity, high speed and also can reduce the area. In future this low complexity FFT processor will be used to implement Radix-16, Radix-32 FFT. By using the reconfigurable techniques for constant multiplier the complexity to perform complex conjugate multiplication in coefficients of FFT can be reduced and there by consumable amount of memory usage to store the data is reduced. In future the same reconfigurable techniques can be implement at the processing blocks or computational blocks of FFT Architecture to reduce latency and increase throughput.

REFERENCE


