Variable-length FFT Processor Architecture

Cao Liu  
Department of Electrical Engineering  
University of California  
Los Angeles  
sjutphoenix@gmail.com

Lin Li  
Department of Electrical Engineering  
University of California  
Los Angeles  
njustilin@gmail.com

Abstract—In this paper, we proposed an variable length FFT processor architecture which could be used in multi-mode and multi-standard OFDM communication system. Based on radix-2 single-path delay feedback architecture, FFT is implemented with the aid of LUT and control logic. The expected architecture would complete FFT ranging 2048/1024/512-point.

Keywords—FFT; Radix-2 FFT; variable length FFT;

I. INTRODUCTION

Thanks to FFT, we could achieve the less computation complexity for DSP. However, modern communication systems applications call for various FFT-length architectures, especially in OFDM technique. Moreover, engineers also need to concern the hardware cost and power consumption when design the architecture. It is our objective to find a proper architecture of variable-length FFT processor, balancing the speed and cost to obtain a high efficiency [2].

[1] reports a variable length FFT processor design which is based on a radix-2/4/8 algorithm and a SDF architecture. This paper lays the foundation of our project which provides us the basic structure. This processor could perform FFT operations of variable-length with two radix-2 and three radix-8 stages. In our design, the range is highly expanded by modifying the length switching structure.

In our project, we focus on the architecture design of the 2048/1024/512-point FFT processor by constructing it with Simulink. Meanwhile, we proposed three cases for the Twiddle Factor LUTs design and evaluate the performance of them by comparing the simulation results.

This paper is organized as follows. Section II identifies the radix-2 FFT algorithm and shows the integrated block diagram of our architecture. Section III describes the implementation of the look-up-table for twiddle factor. Section IV focus on describing the how the variable-length FFT switches its range. In Section V, the simulation results are shown, including the comparison of different probable architectures over area and timing. Then the conclusions are drawn in Section VI.

II. RADIX-2 BASED ARCHITECTURE

In this project, to meet with the requirement of variable length FFT process, the Radix-2 FFT algorithm and architecture should be take into consideration.

The basis of this algorithm is Radix-2 FFT architecture which could be sorted into multi-path delay architecture and single path delay ones. To the multi-path delay, it is feedforward and PE could work at 100% utilization: after dividing the input sequence into two parallel data streams by a commutator, the scheduled data streams do butterfly operation in the processing element (PE) and then multiplied by twiddle factors. To the single-path delay feedback architecture (Figure 1), the delay elements are more efficiently used, however, the butterfly units and multipliers are only 50% utilized.

In this project, we use the radix-2 single-path delay feedback architecture and use ROM to store the twiddle factor if necessary. The MUX select signal is related to the number of point. (For 2^n-point, the period of selection is N, numerously.) To meet with the requirement of variable length, up to 2048-point, the block diagram of the variable length FFT processor is showed as Figure 2.

III. TWIDDLER FACTOR LOOKUP TABLE

According to the FFT size, the twiddle factors’ size is different. Our design is a variable length FFT which has maximum 2048 in size. As a result, at least 512 twiddle factors are necessary. They stand for π/2 of the sine and cosine values.

The radix-2 FFT algorithm introduces the exact position where the proper twiddle factors are needed.

The trick for Twiddle factor is that before each butterfly, the second part of its input should be multiplied by the twiddle factor. These twiddle factors are stored in the ROM, represented as LUT cells. Figure 4 illustrates that the size of the LUT table grows exponentially from the first stage to the last. Before the last stage of butterfly, totally 1024 twiddle factors with a phase spacing of 2π/N are stored in the ROM. [2] introduces the radix-2/4/8 algorithm stores only one-eighth cycle of the sine/cosine waveforms to reduce the ROM size. However, it limits the length choice for the variable length of FFT. In figure 4, it can be observed that C1 & C2 are symmetric, as well as C3 &C6, C4 &C5. The twiddle factor ROM can be designed to be only half of the original size. But the penalty is the size of the control part. This structure needs another look up table to choose the twiddle factor at exact instance. If x is the bits length of twiddle factor, then

\[ ROM = 2 \times 1024 \times x = 2048x \]

\[ ROM_{symmetric} = 512x + 1024 \times 9 \times 2 + 1024 = 512x + 1024 \times 19 \]

Assume ROM_{symmetric}<ROM, we can get that x must be larger than 12 bits. That means if the twiddle factors are 12
bits stored in the ROM, the symmetric ROM has benefits in size.

The proposed cases for the LUT structure are: single ROM, partial combined ROM and distributed ROM. For the single ROM structure, 10 complex multipliers are necessary for the longest 2048 FFT. If these multipliers share the same ROM, it is impossible to achieve high speed, and different part of the ROM has different utilization.

According to the tradeoff between time and area, in our design, the multipliers in the front half stages have their own LUT attached to them, and the last 4 stages of multipliers share the same LUT.

The shared ROM is divided into 4 parts: 128, 129-256, 257-512, and 513-1024. The 128 ROM will be interleaved by factor of 4, 129-256 by 3, and 257-512 by 2. The ROM which has larger size will result in bad delay. Interleaving uses up to 4 times frequency, that means the clock cycle for this part is at least 4*t. Large ROM is not proper to be interleaved too many times. In this structure, the area and speed are proper optimized.

The third case is to distribute the unique ROM to every stage. This structure shows its advantages in both area and speed according to the simulation results.

IV. VARIABLE-LENGTH SWITCH IMPLEMENTATION

The implementation of switching the variable-length option is shown in Figure 6. Our goal is to realize the FFT operation of 512/1024/2048 on this processor. To control the operation length, one additional control input “select” is needed: 0 stands for 2048, 1 stands for 1024, and 2 stands for 512.

To implement the variable length structure is nothing but add MUX gates at the first 3 stages. If it is selected to work as length 2048, then data is fed into the first stage, while in the 1024 mode, data will ignore the first stage and enter the second one directly. Data in 512 mode will pass over stage 1 and 2, starting from stage 3.

As the operational mode changes, the twiddle factors in the rest stages will change at the same time. The twiddle factor of mode 1024 is just the half of one in mode 2048, while the mode 512 is one quarter.

V. SIMULATION RESULT DISCUSSION

In this project, we do the functional verification for cosine signal, shown in Figure 7. To obtain a better performance, the different structures are considered with pipelining, interleaving, word length optimization respectively.

If these multipliers share the same ROM using time-multiplex, the clock cycle will be at least 8 times of reading propagation delay of 1024-words ROM. As a result, our design makes some compromise, and gives a better area and delay product. (The delay and resource are simulated on FPGA.)

The Table I gives the comparison of three different structures, with pipeline and without pipeline. It is shown that the sharing structure has advantage in both delay and area. The design without pipeline is resulted in the long critical path, from the entrance to output. It is the dominated delay that far more than the delay of reading ROM. After adding pipeline between stages, the critical path becomes reading ROM. As a result, the less users of ROM, the faster the system is. The results shows that partial combined structure suffers from more area and timing consumption because of the complexity, compared with the other two methods.

Table II shows the result of RC synthesis which gives the same evaluation for the ROM structure.

Word length optimization is taken into consideration. The input data is (12, 10), which is chosen according to the reference paper. In order to maintain the accuracy of data, we originally keep the word length as (20, 10), for 2048 length of FFT operation. After the word length optimization, the resource cost is reduced by 10%.

VI. CONCLUSION

In this paper, we have reported the architecture design of the variable-length, ranging 512/1024/2048-pint FFT processor which is suitable for OFDM communication systems. The implementation is based on the single-path-delay feedback architecture, with the modules including twiddle factor LUTs and switching length structure. We also show the variable performance of different architecture and make the comparison among them. This implementation could be upgraded by using the Radix -2/4/8 processing element, which could decrease the number of delay elements by half in every stage.

REFERENCES

APPENDIX

Figure 1 Diagram of radix-2 single path delay feedback architecture and PE

Figure 2 Multiplier part diagram

Figure 3 Length-16 FFT Twiddle Factor

Figure 4 block diagram of variable length FFT processor

Figure 5 Variable Length Data Flow Control Part
Fig 5. Interleaving to utilize the LUT

(a)  

(b)  

(c)  

Fig 7. Simulation result for cosine signal with variable length (a)(b)(c)

<table>
<thead>
<tr>
<th>LUT Structure</th>
<th>Utilization (LUTs)</th>
<th>Frequency(MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Pipeline</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single ROM</td>
<td>5148</td>
<td>10.6</td>
</tr>
<tr>
<td>Partial Combined ROM</td>
<td>6061</td>
<td>10.5</td>
</tr>
<tr>
<td>Distributed ROM</td>
<td>5566</td>
<td>10.5</td>
</tr>
<tr>
<td>Pipelined</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single ROM</td>
<td>5128</td>
<td>23.1</td>
</tr>
<tr>
<td>Partial Combined ROM</td>
<td>5428</td>
<td>72.9</td>
</tr>
<tr>
<td>Distributed ROM</td>
<td>5505</td>
<td>74.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LUT Structure</th>
<th>Area</th>
<th>Frequency(MHz)</th>
<th>Power(mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipelined</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Single ROM</td>
<td>1932805</td>
<td>38.58</td>
<td>90.432587</td>
</tr>
<tr>
<td>Distributed ROM</td>
<td>2309882</td>
<td>188.68</td>
<td>156.893480</td>
</tr>
</tbody>
</table>