Proceedings of the 10th WSEAS International Conference on CIRCUITS, Vouliagmeni, Athens, Greece, July 10-12, 2006 (pp275-279)

A FFT/IFFT Soft IP Generator for OFDM Communication System

Tsung-Han Tsai, Chen-Chi Peng and Tung-Mao Chen
Department of Electrical Engineering,
National Central University
Chung-Li, Taiwan

Abstract: - In this paper, we propose a FFT/IFFT soft intellectual property (Soft IP) generator for orthogonal frequency division multiplex (OFDM) systems. We have also presented a system-on-chip (SoC) platform and Soft IP architecture, and provided an easier programmable and reusable IP design. Therefore, the proposed design methods are suitable for the SoC applications and can reduce the development time-to-market. With different input parameters, our FFT/IFFT Soft IP generator can automatically generate a synthesizable Verilog HDL code, test bench, and synthesis scripts files between 8 to 8192 points of FFT/IFFT for OFDM systems.

Key-Words: - Soft IP generator, FFT, IFFT, OFDM, Verilog HDL.

1 Introduction

OFDM is a spread spectrum modulation technology. Because it has the advantage of saving bandwidth, high data transmitting rate and reducing inter symbol interference (ISI) effect, today OFDM has become one of the most popular choices of transmission modulation technology in many communication systems,

The key component in OFMD system is the Fast Fourier Transform (FFT) and its inverse FFT (IFFT). For several kinds of standard applications, such as IEEE802.11a, digital audio broadcast (DAB), and digital video broadcast (DVB), the specifications for FFT/IFFT are different. Especially, FFT/IFFT is also generally needed in some video and audio coding. This also increases the design complexity and makes it difficult to design several FFT/IFFT cores to match each application.

As a result of the surprising breakthrough for the process technology of IC, the size of chip has been significantly reduced while the capacity has been oppositely getting much more complicated. In SOC design environment, IP reused will be a key technique to success. If there is an efficient CAD tool for designing IP, the designing phase would be short for more prompt time-to-market. This is why s Soft IP generator in SOC design is more and more important [1].

In this paper we propose a FFT/IFFT Soft IP generator to provide this key component in SOC environment. Our main target is tried to cover all of the general communication standards in which OFDM is applied. We also introduce in detail about the IP features, implementation algorithms, and the results after synthesis.

1.1 System Description

In our FFT/IFFT Soft IP generator, it is consisted of three major parts: (1) design parameter manager, (2) module integration arrangement, and (3) automated synthesis interface. Figure 1 shows the design flow of the FFT/IFFT Soft IP generator. The detailed descriptions of these parts are discussed as follows:

- Design Parameter Manager

In our FFT/IFFT Soft IP generator, the parameters given by the user are calculated internally to analyse if the parameters are valid. If the input values are invalid, the system will respond error messages to the user. Otherwise, the system will call subsegment functions to handle it.

- Module Integration Arrangement

Module integration arrangement generates a circuit according to the design parameters. Each module is generated by one unit of the design hierarchy. This feature makes the module easy to be expanded and maintained.

- Automated Synthesis Interface

From the output HDL description, the automated synthesis interface links Verilog HDL file into Synopsys synthesis routines. It is possible to describe the synthesis routines like Verilog HDL. The files are called rapid script files. These files can be immediately reported to the users about the results of timing, area, and power.

1.2 Features

The FFT/IFFT has several important parameters in the hardware design, e.g. FFT/IFFT size, sampling rates, input/output wordlength [2], [3], and FFT coefficient. Different parameters strongly depend on

different specific applications domain such as DAB, DVB-T, and IEEE802.11a. Our FFT/IFFT Soft IP generator can provide these different application demands in operation speed and length of FFT/IFFT. The basic specifications are shown in Table 1.

Referring to the hardware module side, we provide two architecture modules. One is the pipeline FFT/IFFT hardware architecture, and the other is memory-base hardware architecture. Details will be expressed later.

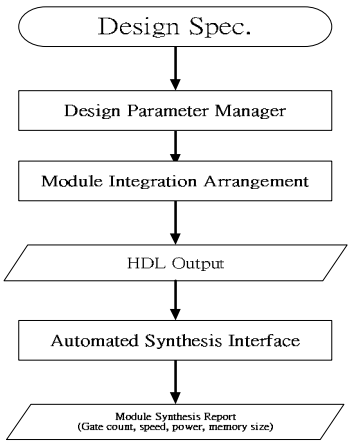

Figure 1. Design flow of the proposed FFT/IFFT IP Generator.

Table 1. Basic specifications of some OFDM based communication standards.

Communication System	FFT Size (Sampling Rate)
802.11a	64 (20MHz)
HiperLAN/2	64 (20MHz)
DAB	2048(2Mhz) ,1024(2MHz) , 512(2MHz),256(2MHz)
DVB-T	8192(8MHz),2048(8MHz)
ADSL	512(2.2MHz)
VDSL	8192(34.5MHz),4096(17.3MHz), 2048(8.6MHz),1024(4.3MHz), 512(2.2MHz)

2 Parameterized Design Flow

The broad application of digital signal processing has resulted in many different types of integrated circuit solutions. We integrate the IP generator with IP's design parameter to achieve better demand. The IP design parameters table is shown in Table 2.

The parameter selection can be divided three steps and discussed as follows:

- Basic Parameters Selection

In our FFT/IFFT Soft IP generator, we use fixed point numeric format. In the first step, designer can setup the input/output wordlength, and set sign bit, integer bits and decimal bits.

- Advanced Parameters Selection

In this step, designer can select FFT or IFFT, architecture and FFT size.

- Coefficient Parameters Selection

In this last step, designer can setup the coefficient effective length and coefficient weight.

Table 2. Design parameter of IP generator.

Parameter	Description	Range
Function	Select FFT or IFFT function	FFT/IFFT
Point	The points for FFT/IFFT	8~8192
WL	I/O Data Wordlength	8,16,32,64
position	Set point position (integer and decimal)	Wordlength-1 ~1
Architecture	Two architecture design for FFT	Pipeline /Memory-Base
CL	Coefficient effective length	1~8
CW	Coefficient Weight	>0

The parameter selection can be divided three steps and discussed as follows:

- Basic Parameters Selection

In our FFT/IFFT Soft IP generator, we use fixed point numeric format. In the first step, designer can setup the input/output wordlength, and set sign bit, integer bits and decimal bits.

- Advanced Parameters Selection

In this step, designer can select FFT or IFFT, architecture and FFT size.

- Coefficient Parameters Selection

In this last step, designer can setup the coefficient effective length and coefficient weight.

3 Architecture Implementation

3.1 Interface and File Analysis

In this paper, we use the cell-based design flow to implement our FFT/IFFT core. We choose C language to implement an FFT/IFFT Soft IP generator, and use Verilog HDL to describe FFT/IFFT IP. We develop the IP generator with interactive menu. The generator will list the descriptions for required input parameters. These code generators are coordinated by a Window GUI. By this GUI, Verilog RTL code of the FFT/IFFT core

can be generated after basic parameters, advanced parameters and coefficient parameters are determined.

3.2 FFT Architecture

In our IP generator, we use two popular hardware architectures to realize FFT algorithms for real-time applications. One is pipeline-based design and the other is memory-based design. In general, pipeline architecture can achieve high-throughput with moderate complexity and low area requirement. Therefore, which architecture will be adopted is usually dependent on the requirements of lower area or higher speed constrain. Basically, no matter what architecture is used, it must satisfy the throughput rate required by application specification.

3.2.1 FFT Pipeline Architecture

The pipeline-based architecture is the best choice for high throughput applications. Especially due to its regular structure and relatively simple control, it is the best choice to implement high-speed long size FFT. Several pipelined architecture have been developed, such as multi-path delay commutator (MDC) [4], single path delay feedback (SDF) [5], and single path delay commutator (SDC). Here we select SDF pipeline architecture which can be easily scaled and parameterized in hardware design. FFT core is implemented based on radix-2/4/8, radix-2/4, and radix-2 algorithm [6]. This pipeline architecture can be easily used on 8 to 8192-points FFT by adding several "processing element" (PE). Table 3 shows the FFT circuit in pipeline architecture. For example, the 128 points FFT pipeline architecture is demonstrated in Figure 2.

3.2.2 Radix-2/4/8 Butterfly Processor

A radix-2/4/8 butterfly [6] processor is shown in Figure 3. In the first *4/N* clock cycles, the first processing element (PE1) shifts the data stored in shift register and outputs it to next stage. At the same time, it stores data from previous output to shift register. In the next *4/N* clock cycles, multiplying a factor of *–j* with the data stored in shift register can achieve the multiplication by exchanging the imaginary part and real part.

The previous stage output data is moved to the shift register to next stage. In the next *2/N* clock cycles, data stored in shift register is added by the previous stage output and moved to next stage, then subtracted by two data and stored in shift register. The PE1 architecture is show in Figure 3(a). About the second processing element (PE2) and the third processing element (PE3) architecture, they are shown in Figure 3(b) and Figure 3(c) respectively.

Table 3. Implementation for 8 to 8192-points FFT circuit.

FFT/IFFT Size	Need circuit
8 points	Radix-2/4/8
16 points	Radix-2+Radix-2/4/8
32 points	Radix-2/4+Radix-2/4/8
64 points	2*Radix-2/4/8
128 points	Radix-2+2*Radix-2/4/8
256 points	Radix-2/4+2*Radix-2/4/8
512 points	3*Radix-2/4/8
1024 points	Radix-2+3*Radix-2/4/8
2048 points	Radix-2/4+3*Radix-2/4/8
4096 points	4*Radix-2/4/8
8192 points	Radix-2+4*Radix-2/4/8

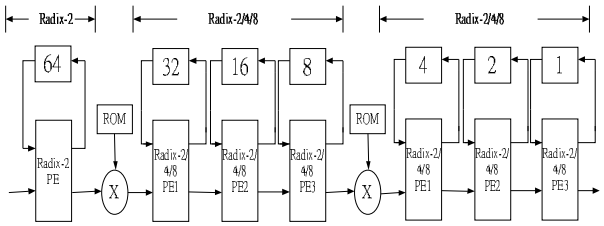

Figure 2. A pipeline 128 points FFT using SDF.

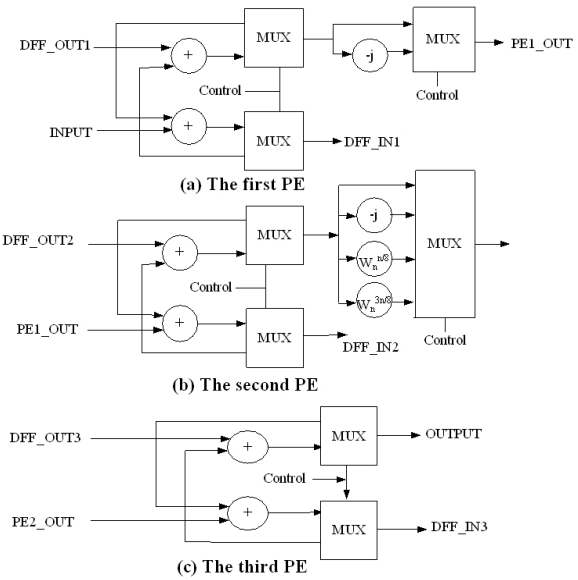

Figure 3. The structure of the radix-2/4/8.

The coefficient architecture with twiddle factors *1*, *-j*, $W^{N/8}$, and $W^{3N/8}$ are shown in Figure 4. In addition, the multiplication by $\sqrt{2}/2$ can be further reduced to additional shift operations to create a multiplication free operation [7], [8].

3.2.3 FFT Memory-based Architecture

The memory-based architecture is another popular architecture used to implement FFT algorithms. The

feature of this architecture is that it uses only one radix-*n* butterfly PE to compute all the radix-*n* butterflies presented. Such that the PE hardware cost is much lower than the pipeline structures. The block diagram of this memory-based architecture is shown in Figure 5.

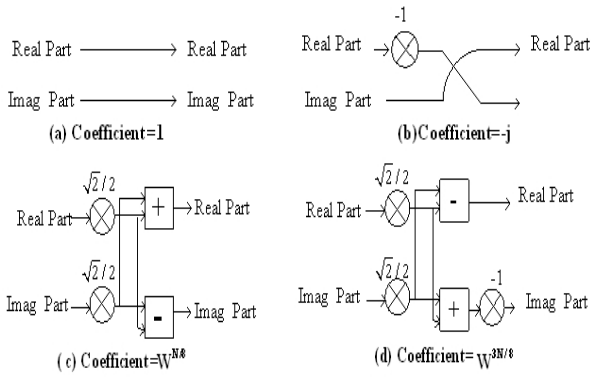

Figure 4. The coefficient architecture.

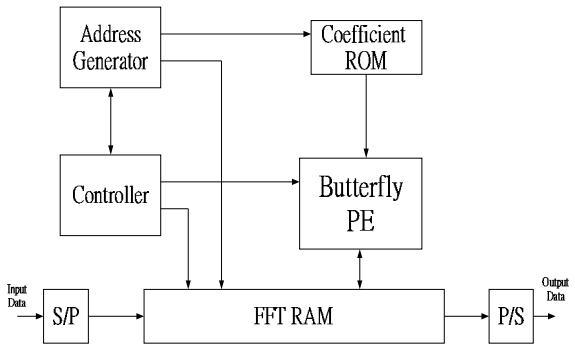

Figure 5. The memory-based architecture for continuous-flow FFT.

4 Structure of FFT IP Generator

The structure of proposed FFT IP generator is shown in Figure 6. The hardware architecture is composed of four basic units. They are stated as follows:

- Butterfly Processor Unit

The arithmetic operations of FFT and IFFT include:

a. complex add calculation.
b. complex subtract calculation.
c. complex multiply calculation.
d. complex divide calculation.

- Data Control Unit

In our architecture, several paths are switched between the data input and data output. Therefore, it requires a control unit to control the data paths. This data control unit can be further divided as follows:

a. data stream input control unit.
b. queue data stream write control unit.
c. queue data stream read control unit.

- Coefficient Table

Coefficient table is used to store all of the coefficients in FFT and IFFT calculation. In order to support more accurate calculation for designer, we provide several parameters to identify the coefficient in our IP generator.

- Memory Unit

In our architecture, all units in order to store the temporary data are called memory unit. It can delay the data rate to match the calculation rate in the architecture.

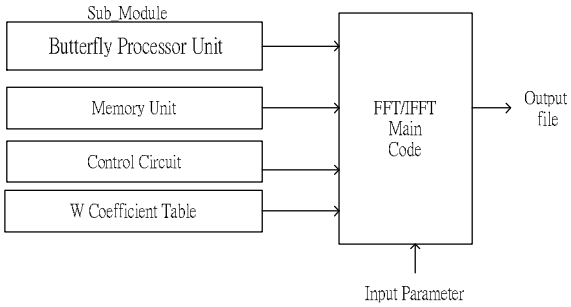

Figure 6. The structure of FFT IP generator.

5 Experimental Results

5.1 Test Bench and Synthesis Scripts Files

We develop a simple test bench file for the user to test the function blocks of FFT code. Also, we create rapid synthesis script files and synthesis constraint files to synthesize module.

5.2 Result Analysis

We first use our FFT/IFFT Soft IP generator to generate the IP core, and then synthesize and implement it. Figure 7 shows the 8-points FFT waveform produced from logic analyzer. Figure 8 shows the simulation waveform, which contains FFT input signal, FFT output signal, and some control signals. In this flexible environment, users can trace and debug the design result easily.

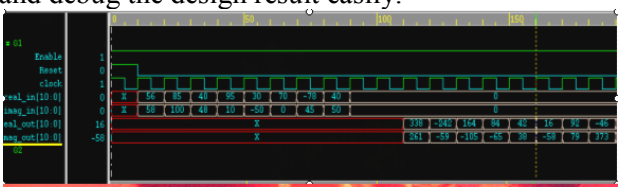

Figure 7. The Simulation of 8-points FFT.

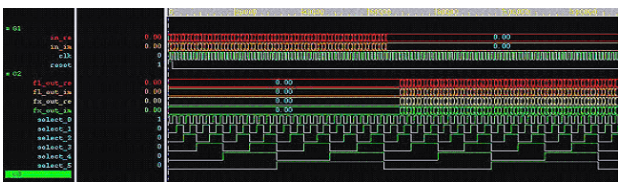

Figure 8. The Simulation waveform result.

6 Conclusions and Future Works

We have developed a parameterized FFT/IFFT Soft IP generator which can be applied to OFDM system in general applications. This Soft IP generator provides several parameters and two architectures for user to select. We have also presented a SoC platform and Soft IP architecture to design FFT/IFFT. The proposed design method is suitable for the SoC applications and reduces the development time in the Soc products.

Future works will address on the VLSI implementation on some OFDM-based communication systems. Our present target is on the DVB-T/H baseband receiver SOC design with our embedded FFT/IFFT Soft IP.

References:

[1] S. Pillement, D. Chillet, O. Sentieys, "Behavioral IP specification and integration framework for high-level design reuse," *2002. Proceedings. International Symposium on Quality Electronic Design,* pp. 388-393, Mar. 2002.

[2] S. Johansson, S. He and P. Nilsson, "Wordlength Optimization of a Pipelined FFT Procdssor," *Proceeding of Midwest Symposium on Circuits and Systems (MWSCAS)*, vol. 1, pp. 501-503, 2000.

[3] R. B. Perlow and T. C. Denk, "Finite Wordlength Design for VLSI FFT Processors," *2001. Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers*, vol. 2, pp. 1227-1231, Nov. 2001.

[4] L. R. Rabiner and B. Gold, "Theory and Application of Digital Signal Processing", *Prentice-Hall*, Inc., 1975.

[5] S. He and M. Torkelson, "Designing Pipeline FFT Processors for OFDM (de)Modulation," *Proceeding of 1998 URSI International Symposium on Signals, Systems, and Electronics,* pp. 256-262, 1998.

[6] L. Jia, Y. Gao, J. Isoacho and H. Tenhunen, "Implementation of A Low Power 128-Point FFT," *1998 5th International Conference*, pp. 369-372, 1998.

[7] L. Jia, Y. Gao, J. Isoacho and H. Tenhunen, "New VLSI-Oriented FFT Algorithm and Implementation," *IEEE ASIC Conference*, pp. 337-341, 1998.

[8] L. Jia, Y. Gao, J. Isoacho and H. Tenhunen, "Efficient VLSI Implementation of Radix-8 FFT Algorithm," *IEEE Pacific Rim Conference on Communication, Computers and Signal Processing*, pp. 468-471, 1999.