Design and implementation of a large points FFT acceleration unit in multi-processor system based on FPGA

Duo-li Zhang,Xue-peng Yang, Yu-kun Song
Institute of VLSI design
Hefei University of Technology
Hefei, Anhui, China
zhangduoli@hfut.edu.cn,francis_yang23@126.com

Abstract-**This paper introduces the design and implementation of a large points FFT acceleration unit of multi-processor system based on FPGA. It introduces radix-2 DIT-FFT algorithm and the features of hardware platform. It analyzes the FFT acceleration unit overall and divides it into several modules. Then hardware implementations of modules are discussed. The whole FFT acceleration unit is a part of one multi-processor image processing system and the whole system is tested by software and hardware co verification. Results are verified between the software simulation and Virtex6 evaluation board of Xilinx Company. The results are compatible in the range of error. The FFT acceleration unit meets the needs of the whole system and the design is successful.**

Keywords-FFT; FPGA; Radix-2; DIT-FFT; Multi-processor; software and hardware co verification.

I. INTRODUCTION

FFT(Fast Fourier Transform) is the fast algorithm of DFT(Discrete Fourier Transform), it increases DFT's operational efficiency 1~2 orders of magnitude and has become the important transform tool in the areas of image, audio and graphics of DSP (Digital Signal Processing).It has promoted the development of DSP technology greatly and has gotten extensive application in Engineering.

There are three ways for the hardware implementation of FFT algorithm: 1) DSP; 2) ASIC; 3) FPGA. The DSP scheme is easy to develop and can be programmed by software. It has great flexibility, but it has slow speed and the interface is not flexible. It is also lack of mass memories for FFT operation and needs special interfaces, control chips and RAM. So it is hard to achieve high speed, large scale FFT operation. ASIC scheme can get high speed operation and is suitable for the application of high speed signal processing system. But the custom ASIC has long development cycle and complex peripheral circuit, it also has bad scalability and it has risk. FPGA has the advantages of the flexibility of software programming and fast speed of ASIC design. With the development of IC technology and the rising of IC manufacturing, the configuration and performance of FPGA meets the requirements of FFT and it is suitable for large amount of real-time digital signal processing, it can get large points FFT operation. This paper introduces the design and implementation of a large points FFT acceleration unit in multi-processor system based on FPGA.

This FFT acceleration unit is the arithmetic unit of multi-processor image processing system, it takes the radix2 DIT-FFT (Decimation-In-Time FFT) algorithm and gets 32-bit single precision floating point operation from 32-point to 16K-point.After the implementation of unit, it is integrated into the whole system and tested by software and hardware co verification, the results show that the FFT acceleration unit meets the requirements of multi-processor image processing system and the design is suitable for the system.

II. PRINCIPLE AND ALGORITHM STRUCTURE OF FFT

The discrete fast Fourier transform of the N-point is defined by

$$X(k)=\sum_{n=0}^{N-1} x(n)W_N^{kn},k=0,1,...,N-1 \quad (1)$$

Wherein $W_N^{kn}=\exp[-j(2k\pi/N)]$. $x(n)$ is complex sequence, for a certain k value, it needs N times of complex multiplications and (N-1) times of complex additions. For all k values, it needs N^2 times of complex multiplications and $N(N-1)$ times of complex additions; the computation is so large that it is difficult to fulfill the requirements of computation speed in the areas of real time processing. In this case, the FFT algorithm appears and its operational efficiency affects the speed of signal processing directly. FFT algorithm is to divide long sequences of DFT into several short sequences of DFT constantly and use the periodicity and symmetry of twiddle factor W_N^{kn} to reduce the number of operation times for DFT, and then it can avoid repeated operations and improve the speed of operation for DFT. There are many kinds of algorithm pattern, such as radix-2 FFT algorithm, radix-4 FFT algorithm. There are two categories: one is Decimation-In-Time (DIT) FFT algorithm, the other is Decimation-In-Frequency (DIF) FFT algorithm. The two ways are essentially decomposition algorithm based on label and are same in the computation and complexity. The difference between them is that DIT algorithm does

The work is supported by National Natural Science Foundation of China (61006024, 61006020)

Published by Atlantis Press, Paris, France.

complex multiplication first and then does complex addition, but DIF algorithm is reverse. This design selects radix-2 DIT-FFT algorithm.

The basic unit in FFT operations is butterfly operation unit. There are three input variables and two output variables in one radix-2 butterfly operation unit. The algorithm flow symbol is as below:

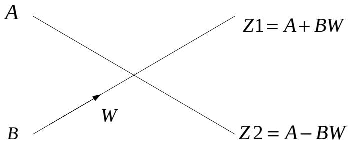

Figure 1. Butterfly unit algorithm flow symbol

In Figure 1, A and B are the two operands for butterfly operation, W is the twiddle factor, Z1 and Z2 are the output variables. From Figure 1, we can get the basic formula of butterfly operation:

$$\begin{cases} Z1 = A + BW \\ Z2 = A - BW \end{cases} \quad (2)$$

FFT algorithm can be constituted by a multi stage operation. There are many forms of concrete operation flow diagram; designers can select different operation flow diagrams based on requirements.

The characteristics of Radix2 DIT-FFT algorithm are as below: 1) $N = 2^M$ points need M-stage operation and the number of butterfly unit in each stage is $N/2$.The whole number of butterfly operation is $\frac{N}{2} \times M = \frac{N}{2} \times \log_2 N$ .In the same stage, the two input variables of each butterfly are only useful for the operation of this butterfly unit. 2) For N- point FFT, the amount of complex multiplication reduces from N^2 times to $\frac{N}{2} \times \log_2 N$ and the complex additions reduces from $N(N-1)$ times to $N \times \log_2 N$, it reduces operation load largely and improves the operation efficiency greatly. 3) After one butterfly operation, the output results can be stored in the memories where the previous input data are stored until the final results export. This method can be achieved by small memory sources and lower-cost storage. 4) Bit code reverse means that the serial number n will be written in binary code and made inverted, then decode the inverted binary code into decimal. 5) In N-point DIT-FFT operation, each stage has N/2 butterfly and each butterfly needs to multiply twiddle factor. Use L to represent arithmetic stage (L=1, 2… M), then there are total 2^{L-1} different twiddle factors. In stage L, the interval of two input variables is 2^{L-1} -point for each butterfly; for the same twiddle factor, there are 2^{M-L} butterfly and the interval is 2^L -point.

III. HIGH-PERFORMANCE FPGA FEATURES

This design selects the Virtex-6 series FPGA launched in 2009 by Xilinx Company. It uses 40nm copper process CMOS technology with a high-performance logic structure. In addition to containing a large number of Configurable Logic Blocks (CLB), the Input and Output Blocks (IOBs) logic resources and routing resources, it also has features as below:

1) Has high-performance, high-level FPGA logic, provides configurable Look-Up-Table (LUT) technologies, meets the applications of abundant register resources and has higher routing efficiency.

2) Has global clock resources and Digital Clock Management (DCM) modules and can get low jitter and high-performance clock.

3) Provides abundant programmable RAM modules and enhanced type of programmable FIFO logic.

4) Provides advanced DSP48E1 slice to meet arithmetic operations and provides a large number of pipelines and extended functions, then enhances speed and efficiency of many applications and also provides functions other than DSP.

5) Provides flexible configuration options, each device has system monitoring function.

6) Provides abundant interface resources and high-speed data transmission resources.

Features above simplify logic design, reduce design time and provide great convenience for the implementation of large-point, high speed and real-time FFT processing.

IV. HARDWARE IMPLEMENTATION

This acceleration unit uses radix2 DIT-FFT algorithm and completes 32-bit single precision floating point FFT operation at any point which varies from 32-point to 16K-point. Radix2 algorithm is suitable for hardware implementation, but the computing cycle is double compared to radix4 algorithm. Most FFT operation are based on radix4 algorithm, but it only completes 4^n -point FFT operation, points like 128,512,2K or 8K can't use radix4 algorithm directly. Based on the comprehensive consideration of hardware implementation and algorithm, the design uses dual parallel structure. It uses two butterfly operation units and completes read or write operations of 4 data simultaneously. By parallel operation, operation cycle reduces by half and operation speed is twice of original, then operation efficiency improved. The entire design includes Butterfly Operation Unit, Memory Unit, Address Generating Unit (AGU), Control Logic Unit and the structure is as below:

Published by Atlantis Press, Paris, France.

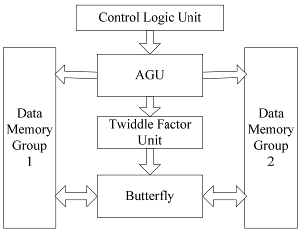

Figure 2. Structure of whole unit

Control Logic Unit: Control the implementation of the whole hardware operation that FFT/IFFT mode, FFT point, FFT stage are included.

AGU: Controlled by Control Logic Unit, generates the read-address, write-address of memories and read-address of twiddle factor.

Data Memory Group 1&2: Store raw data, intermediate processing data and result data.

Twiddle Factor Unit: Store the twiddle factors for butterfly operation.

Data Memory Group1&2 and Twiddle Factor Unit are included in Memory Unit.

Butterfly Operation Unit: It contains two radix2 butterfly operation units and simultaneously reads 4 data for twice butterfly operations, then generates 4 results.

Each unit works under the control of Logic Control Unit, Memory Unit uses Ping pang RAM structure to expand amount of data storage; Butterfly Operation Unit uses dual parallel structure to improve operation speed.

A. Butterfly Operation Unit

The data of the design are composed of a real part and an imaginary part. Suppose $A = Ar + Ai$; $B = Br + Bi$ and $W = Wr - Wi$ according to radix2 DIT-FFT algorithm, formula (2) can be written as below:

$$\begin{cases} Z1=A+BW=[Ar+(BrWr+BiWi)]+j[Ai+(BiWr-BrWi)] \\ Z2=A-BW=[Ar-(BrWr+BiWi)]+j[Ai-(BiWr-BrWi)] \end{cases} \quad (3)$$

One radix2 complex operation needs once complex multiplication and twice complex additions, that is to say, 4 real multiplications and 6 real additions. Because the IFFT operation module also can use the FFT module, we combine the two in this unit. The structure is as below:

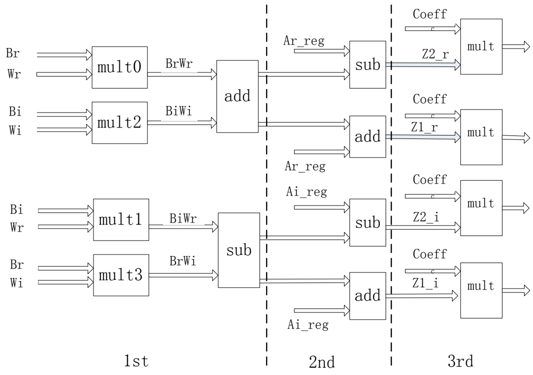

Figure 3. Butterfly Operation Unit Structure

The 1st stage completes twice complex multiplications; the 2nd stage completes addition (subtraction) of butterfly operation, A need to be stored for some cycles and then participates the operation of 2nd stage; the 3rd stage pre-processes the butterfly results according to the FFT/IFFT mode and then do next butterfly operation, wherein Coeff is constant factor which is 1 in FFT mode and 1/2 in IFFT mode in this design. Multipliers, adders and subtracters are IP cores from Core Generator of ISE.

B. Memory Unit-RAM

RAM cells are used to store input data and FFT results for each stage. The design uses DPRAM that generated by Core Generator to store data. Its width is 64-bit and the low 32-bit stores complex real part and high 32-bit for imaginary part. There are 2 groups of data memory and each group is constituted by 4 blocks of smaller memory. Every Butterfly Unit is corresponding to two groups of RAM which are Memory1 and Memory2. It uses Ping pang RAM structure. The structure between operation and storage is as below:

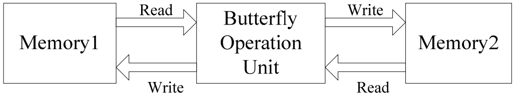

Figure 4. Structure between operation and storage

The raw data are stored in Memory1 according to certain rules, then configure relevant registers and start calculation. Results are sent into Memory2. After one stage of calculation, data are read from Memory2 and results are sent into Memory1.The full operation are completed following this rule and the final results are stored in memory decided by the number of stage.

C. Memory Unit-Twiddle Factor Unit

There are $m = \log_2 N$ stages for N-point FFT and there are N/2 different twiddle factors at last stage and 2^{m-l} different twiddle factors at other stages wherein m is the operation stage. For 16K-point, the number of twiddle factors for FFT is 8K. The formula of twiddle factor is as below:

$$W_N^m = \cos(2\pi m / N) - j\sin(2\pi m / N) \quad (4)$$

The real part and imaginary part are calculated according to formula (4). The 8K results are stored in ROM. There are two ROMs for the dual parallel structure to read twiddle factor simultaneously.

D. Address Generating Unit (AGU)

AGU will generate 3 groups of addresses: RAM read-address for Butterfly Operation Data, ROM read-address for twiddle factors and RAM write-address of Butterfly Operation Results.

RAM read-address has great relation with raw data. There are two data memory groups and each group has 4 same smaller blocks of RAM. The smaller blocks are named as Mem1-4 and the storage is (N/4)*64bit wherein N is the FFT point. The conflict-free address of raw data for storage is as below:

TABLE I. CONFLICT-FREE ADDRESS TABLE

Mem1	0	4	…	N/2+3	…	N-1
Mem2	1	5	…	N/2+2	…	N-2
Mem3	2	6	…	N/2+1	…	N-3
Mem4	3	7	…	N/2	…	N-4
Address	0	1	…	…	…	N/4-1

Four data are read from Mem1-4 and the read-addresses can be referred in TABLEI. The read-addresses are controlled by counters inside and also related to the operation times.

Four results are generated for each operation and will be written into Mem1-4.The write-addresses of each operation are same for Mem1-4 and the storage sequence is related to the number of operations. The relations between input data and output storage sequence are as below:

TABLE II. RELATION BETWEEN INPUT DATA SEQUENCE AND OUTPUT STORAGE SEQUENCE

Butterfly Unit Input data sequence	**Current operation times X**	
	odd	***even***
A_f1	Mem1	Mem3
B_f1	Mem4	Mem2
A_f2	Mem2	Mem4
B_f2	Mem3	Mem1
Butterfly Unit Output storage sequence	**Current storage times X**	
	X < N/8	***X> N/8***
Z1_f1	Mem1	Mem4
Z2_f1	Mem2	Mem3
Z1_f2	Mem3	Mem2
Z2_f2	Mem4	Mem1

RAM write-address is also controlled by counters inside.

ROM read-address for twiddle factor is controlled by 13-bit counter. It uses address bitwise reverse method for the addressing of twiddle factors in ROM and easily implemented in hardware.

E. Control Logic Unit

Control Logic Unit is connected to the multi-processor system by AHB port. It configures FFT mode by register and the configure information includes FFT point, FFT stage, FFT/IFFT mode, FFT start, etc. It also generates enable signals and control signals for each module. After calculation, FFT acceleration unit will send Interrupt signal to the system.

V. EXPERIMENT RESULTS AND ANALYSIS

Simulation of specific module should be carried out after the completion of the FFT acceleration unit. The method is simple and convenient, but it only adapts to smaller or relatively independent unit.FFT acceleration unit is a part of the multi-processor image processing system. In order to verify its correctness in the whole system, it is not a wise choice to use RTL simulation. The design uses software and hardware co verification for this FFT acceleration unit. Take software simulation as a standard for comparison and use FPGA Evaluation Board for hardware results. Software and hardware co verification helps to achieve verification purpose quickly and directly.

Verification development platforms use ISE of Xilinx Company; FPGA selects Virtex-6 Evaluation board of Xilinx Company, too. Take 2048-point 32-bit single precision floating point complex FFT computation as an example. Choose 5 groups of results randomly between FPGA evaluation and software simulation and the results are as below:

TABLE III. RESULTS COMPARISON

Software Simulation results	**FPGA Evaluation results**
6808.145508	6808.149414
-4232.591797	-4232.588867
767.498413	767.5015870
-67.538452	-67.535034
-1109.832397	-1109.833252

It shows that the experimental results can be accurate to thousandths. This is caused by the difference of software parameter defined types and the carry bit of complex floating point, etc. It is introduced error and can be accepted. The conclusion is that results of FPGA verification are right within the range of allowable error.

VI. CONCLUSIONS

The paper introduces the design and implementation of a large points FFT acceleration unit in multi-processor system based on FPGA. The structure of hardware implementation is simple. It passes the software and hardware co verification as a part of the multi-processor image processing system. Results of FPGA verification are right within the range of allowable error. FPGA evaluation results prove that the design of this large point FFT acceleration unit is valid.

With the development of FPGA technologies, it will become feasible for the implementation of large point FFT operation and make more progress in speed, volume and flexibility.

REFERENCES

[1] Y.M,Ding and X.Q.Gao, Digital Signal Processing. Xi'an: Xidian University press house, 2000.

[2] Y.L.Hang, J.J.Liu and J.F.Zhang, "Design of FFT processor based on FPGA," Proceedings of the 24th Chinese Universities Symposium on Electrical Power System Automation pp1347-1350, Oct, 2008.

[3] G.D.Liu, B.X.Chen and D.F.Chen, "Design of FFT processor based on FPGA," Aeronautical Computer Technique, Vol.34 No.3, pp.101-104, Sep.2004.

[4] Y.Zhang and K.L.Fang,"Design of General FFT Processor Based on FPGA," Computer Technology and Development, Vol.20 No.8, pp87-90, Aug.2010.

[5] C.W.Zhao, Research on MPSoC Testing Technology,(Master Thesis) Hefei University of Technology,Hefei,Anhui,China,2012.

[6] N.Hou, Research on Design and Implementation Technology of Embedded Multiprocessor,(Doctor Thesis) Hefei University of Technology,Hefei,Anhui,China,2012.

Published by Atlantis Press, Paris, France.