International Journal of Control and Automation
Vol. 9, No. 11 (2016), pp.137-142
http//dx.doi.org/10.14257/ijca.2016.9.11.12

Design FFT Processor Based on FPGA

Yan Dai

Department of Information Engineering College of TangShan Tangshan ,Hebei, China
daiyan0620@foxmail.com

Abstract

In this paper, the design of decimation in time Radix-2 FFT processor is presented, which is integrated in FPGA chip, regarding FPGA as design carrier. New butterfly-shaped operation method is proposed based on the traditional FFT algorithm and the analysis of hardware unit, which can reduce multiplication. And it adopted the Look-up table method .The results of hardware test of FFT processor shows that the processor works well and has high speed, and the design was realized by using FPGA.

Keywords*: FFT; FPGA; Radix-2 butterfly algorithm*

1. Introduction

In real life, the combination of advanced signal processing algorithms and hardware is used by all fields [1-3]. A fast algorithm of DFT was published by Cooley and Tukey in 1965, which reduced amount of calculation of DFT. As a basic operation of digital signal processing, there are different requirement of FFT processor in different areas. The traditional method of FFT processor design is DSP by software programming. Limit the performance of DSP itself; it is difficult to achieve high-speed, large-scale FFT operations. Researching high-speed and high-precision FFT processor becomes a point of digital processing.

Because of the advantages of flexible scheme, low power consumption, a lot of logical unit and pin ,this paper describes the architectural design of FFT processor which is implemented using field programmable gate arrays(FPGA).FPGA realization is very much useful, particularly in the developing countries, due to low investment required for the same compared to the prototyping cost of an ASIC. Among other benefits are the short design cycle and the scope of reprogramming ability for improvement in the design without any additional cost along with the facility of desktop testing [9].

2. FFT Algorithm

FFT algorithm can be divided into two categories. Firstly, points--N of FFT is equal to 2^m,(m is any natural number); Secondly, N is not equal to 2^m. The second situation can be transformed into the first case, so we used FFT which belongs to the first category. FFT algorithm in the first category included the radix-2, radix-4 and so on. In theory, the radix number is larger, the number of operation is fewer, but the process will become complicated. The relatively widespread use of the algorithm is radix-2.

In radix-2 algorithm, $N = 2^M$. The discrete Fourier transform of N complex samples $x(k)$, $k = 0,1,.....,N-1$ is defined as Equation (1):

$$X(k) = \sum_{n=0}^{n-1} x(n) W_N^{kn} \tag{1}$$

$x(n)$ is divided into groups, $n = 2r$ and $n = 2r+1$. The N point DFT of $x(k)$ as depicted in Equation (1) may be rewritten as

ISSN: 2005-4297 IJCA

$$X(k)=\sum_{r=0}^{\frac{N}{2}-1}x(2r)W_N^{2rk}+\sum_{r=0}^{\frac{N}{2}-1}x(2r+1)W_N^{(2r+1)k} \quad (r=0,1.....,\frac{N}{2}-1) \quad (2)$$
$$=\sum_{r=0}^{\frac{N}{2}-1}x(2r)W_N^{2rk}+W_N^k\sum_{r=0}^{\frac{N}{2}-1}x(2r+1)W_N^{2rk}$$

Where $W_N^{2rk}=e^{-j\frac{2\pi}{N}2rk}=e^{-j\frac{2\pi}{N/2}rk}=W_{N/2}^{rk}$. The Equation (2) can be written as Equation(3):

$$X(k)=\sum_{r=0}^{\frac{N}{2}-1}x(2r)W_{\frac{N}{2}}^{rk}+W_N^k\sum_{r=0}^{\frac{N}{2}-1}x(2r+1)W_{N/2}^{rk} \quad (3)$$
$$=G(k)+W_N^kH(k)$$

Where $G(k)=\sum_{r=0}^{\frac{N}{2}-1}x(2r)W_{N/2}^{rk}$, $H(k)=\sum_{r=0}^{\frac{N}{2}-1}x(2r+1)W_{N/2}^{rk}$.

$G(k)$ and $H(k)$ are DFT of N/2. $G(k)$ includes even sequence from original sequence. $H(k)$ includes only the odd sequence from original sequence. In addition, their period is N/2. Because of $W_N^{N/2}=-1$, $W_N^{k+N/2}=-W_N^k$, we can get the Equation (4) and Equation(5):

$$X(k)=G(k)+W_N^kH(k),k=0,1,2,\cdots,\frac{N}{2}-1 \quad (4)$$

$$X(k+\frac{N}{2})=G(k)-W_N^kH(k),k=0,1,2,\cdots,\frac{N}{2}-1 \quad (5)$$

From the above, DFT of N point sequence is calculated from two DFT of N/2 sequence. And so on, $G(k)$ and $H(k)$ can continue to be decomposed and they synthesize DFT of N sequence finally.

The FFT algorithm in which the input data samples are split into odd- and even-numbered ones, is called decimation in time.

3. The Designed of FFT based on FPGA

The operation of the processer is partitioned into three main processes. These are the Date input, FFT Computation and Data Output Processes. This partitioning is depicted in Figure 1.

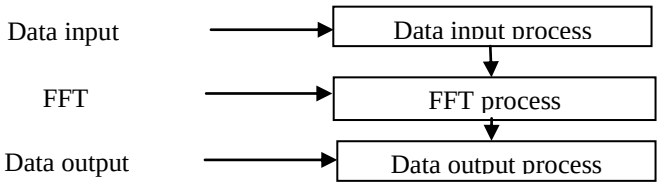

Figure 1. FFT Computation Process

The processing cycle starts with the data input process, during which sampled data is read in and stored in memory. During the FFT computation process, the FFT is computed on the stored data. During the Output process results of the FFT computation process are read out to the outside world. These process are then mapped to hardware resources.

3.1. Butterfly Processing Module

The butterfly is the basic operator of the FFT. It computes a two point FFT. It takes two data words from memory and compute the FFT. The butterfly processing module computes one butterfly every four cycles. It consists of one multiplier and two adders. The RTL for it is shown in Figure 2.

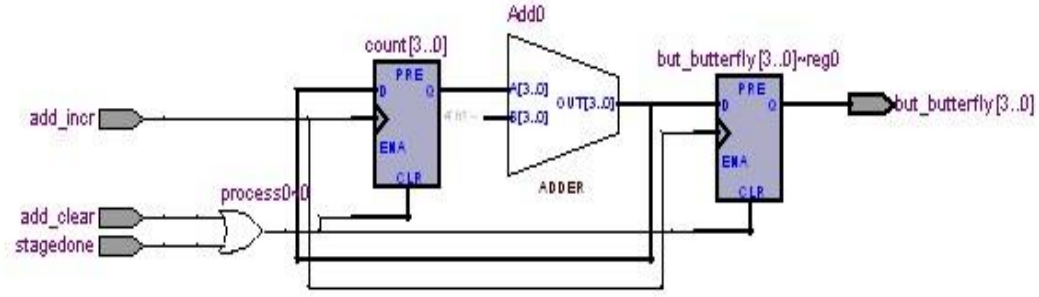

Figure 2. RTL of Butterfly Processing Module

Because of 4 cycles required to calculate one butterfly, note that during data input and data output the butterfly is incremented by the clock while during FFT computation mode. Whenever "clear" or "stage done" signal goes high, the butterfly generator is reset. The simulation waveform of the butterfly generator is shown in Figure 3.

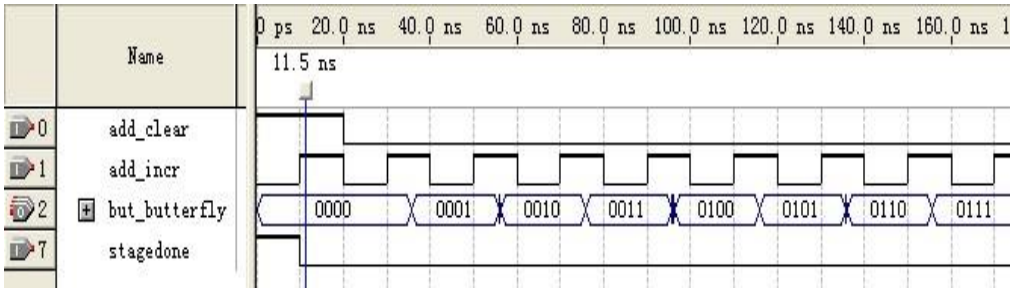

Figure 3. Simulation Waveform of the Butterfly Generator

3.2. The Design of FFT Processor

The FFT processor architecture consists of a single radix-2 butterfly, a dual-port FIFO RAM, a coefficient ROM, a controller and an address generation unit.

The address generation unit mainly generated ROM、RAM access address which is used in the FFT processor.

The controller of FFT processor is the core of FFT processor. It enabled or disabled the work of FFT processor, memory modules, address generation and data distribution. The controller is modelled as a finite machine. It has been states ranging from rst1 and rst7. The actions performed in each state are clearly commented in the code. The signal to and from the controller are given in Figure 4.

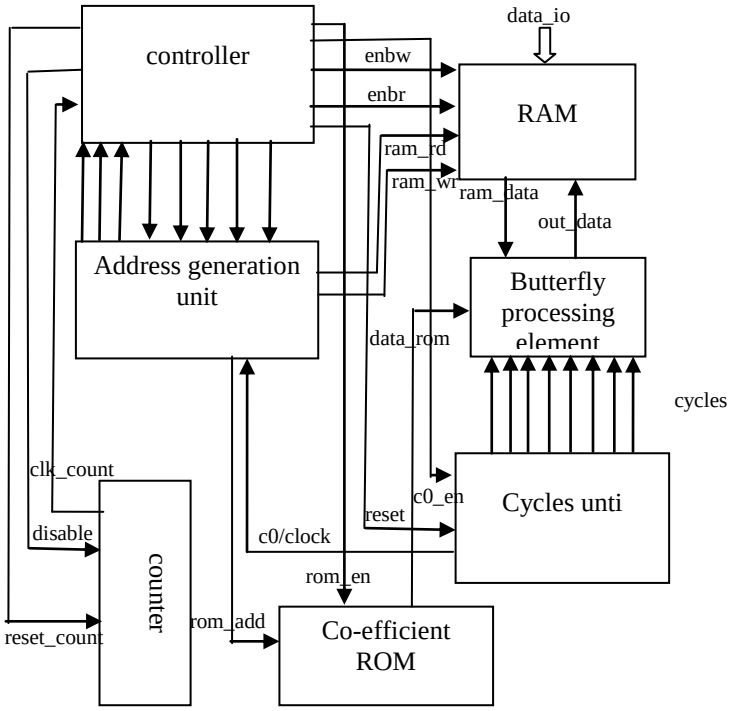

Figure 4. Block Diagram of FFT Processor

3.3. Simulation and Results

Validation data is x=[4000,3000,2000, 6000,7000,8000,9000,0], The simulation waveform of FFT processor based FPGA is shown in Figure 5. The simulation results in matlab is shown in Figure 6.The comparison between them is shown in table 1.

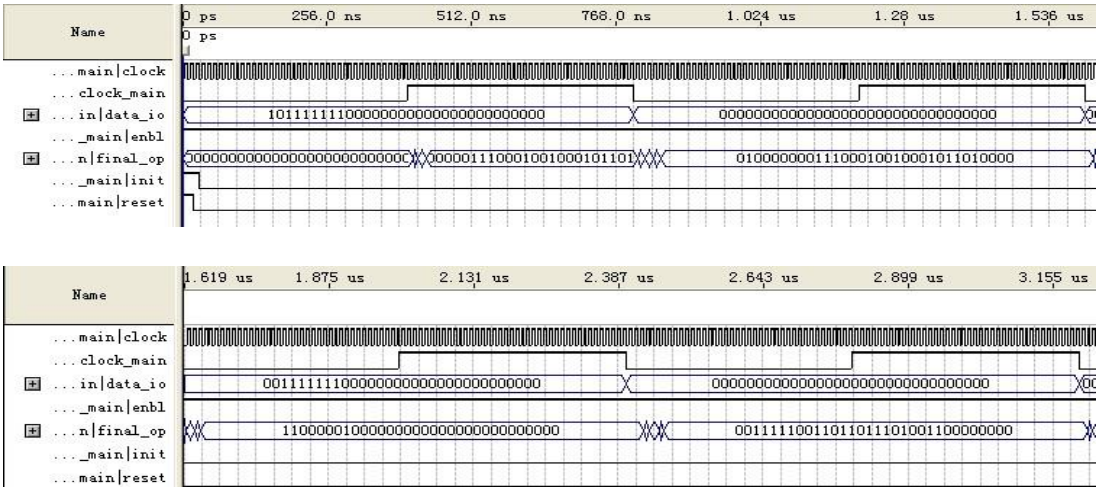

Figure 5. Simulation of FFT Processor Based On FPGA

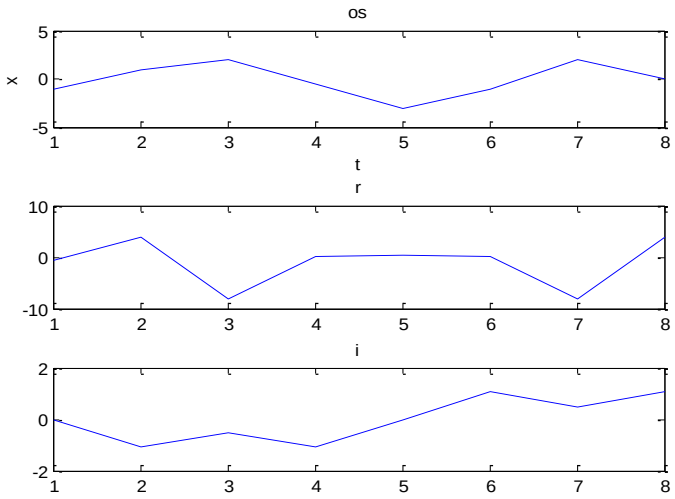

Figure 6. Simulation of FFT Processor in MATLAB

Table 1. The Comparison Between theoretical and calculated value

	Real part theoretical value	Imaginary part theoretical value	Real part calculated value	Imaginary part calculated value	Real part error	Imaginary part error
X(0)	0	0	0	0	0.00%	0.00%
X(1)	3.768	-1.061	3.76775	-1.06065	0.01%	0.03%
X(2)	-8	-0.5	-8	-0.5	0.00%	0.00%
X(3)	0.2322	-1.061	0.23225	-1.06065	0.02%	0.03%
X(4)	0.5	0	0.5	0	0.00%	0.00%
X(5)	0.2322	1.061	0.23225	1.06065	0.02%	0.03%
X(6)	-8	0.5	-8	0.5	0.00%	0.00%
X(7)	3.768	1.061	3.76775	1.06065	0.01%	0.03%

4. Conclusions

The design of FFT processor algorithm using FPGA is presented. The ASIC implementation would be faster in terms of the maximum operating clock frequency but the design with FPGA offers a more cost effective solution having short design cycle and the desktop development capability with the speed which is adequate for our specific application.[5] This paper introduced butterfly element, address generation unit and controller. Finally, it shows that the FFT processor operational results can meet requirements.

References

[1] D. Liu, Z. Wang and G. Yu, "Design of high-accuracy frequency measuring instrument based on FPGA", Hebei Journal of Industrial Science and Technology, vol. 27, no. 1, **(2010)**.
[2] S.-I. Satake, Y. Hiroi, Y. Suzuki, N. Masuda and T. Ito, "Specical-purpose computer for two-dimensional FFT", Computer Physics Communications, vol. 179, no.6, **(2008)**.
[3] A. Banerjee, A. Sundar Dhar and S. Banerjee, "FPGA realization of a Cordic based FFT processor for biomedical signal processing", Microprocessors and Microsystem, vol. 25, no. 1, **(2001)**.
[4] O. Nibouche, S. Boussakta and M. Darnell, "Algotithms and pipeline architectures for 2-D FFT and FFT-like trasforms", Digital Signal Processing, vol.20, no.4, **(2010)**.
[5] Y. Zhou, J.M. Noras and S. Shepherd, "Novel design of multiplier-less FFT processor", Signal Processing, vol.87, no.6, **(2007)**.
[6] Y.E. Cabal, T.D. Carozzi and T.J. Romero, "FPGA-based system for frequency detection of the main periodic component in time series information", Digital Signal Processing, vol.18, no.6, **(2008)**.
[7] G. Zhang, Y. Lu and X. Fu, "Design of 1024-point FFT Processor based on FPGA", Microcomputer Inforamtion, vol.28, no.8, **(2012)**.
[8] A. J. Acosta, E. J. Peralias, A. Rueda and J.L. Huertas, "VHDL behavioural modeling of pipeline analog to digital converters", Measurement, vol.31, no.2, **(2000)**.
[9] Z. Szadkowsk, "16-point discrete Fourier transform based on the radix-2 FFT algorithm implemented into cyclone FPGA as the UHECR trigger for horizontal air showers in the Pierre Auger Observatory", Nuclear Instruments and Methods in Physics Research A, vol.560, no.2, **(2006)**.

