Available ONLINE www.ijcae.org

INTERNATIONAL JOURNAL OF COMMUNICATIONS AND ENGINEERING

National Conference on Emerging Trends in Information, Digital & Embedded Systems (NC'e-TIDES -12), Dept. of ECE, Annamacharya Institute of Technology and Sciences, Rajampet

IJCAE, Vol.3 Issue 1,July 2012, 16- 23

ISSN NO: 0988-0382E

RESEARCH ARTICLE

EFFICIENT IMPLEMENTATION OF FPGA BASED DISTRIBUTED ARITHMETIC ARCHITECTURE FOR FIR FILTER

[1*]V. S. Sivakumar, [2]K.Venkateshwarlu, [3]P.V.Gopikrishnarao and [4]G.Ramesh

[1]P G Student, Prakasam Engineering College, Kandukuru, AP, India

[2]Asst.Professor Dept. of ECE, Prakasam Engineering College, Kandukuru, AP, India.

3 Associate professor Dept of EIE RGMCET NANDYAL AP India

4Asst. professor Dept of EIE RGMCET NANDYAL AP India

ABSTRACT

A filter is used to modify an input signal in order to facilitate further processing. A digital filter works on a digital input (a sequence of numbers, resulting from sampling and quantizing an analog signal) and produces a digital output**.** Designing an LTI involves arriving at the filter coefficients which, in turn, represents the impulse response of the proposed filter design. These coefficients, in linear convolution with the input sequence will result in the desired output. Signal processing ranks among the most demanding applications of digital design concepts and practices. It is a mature technology domain wherein the demands for enhanced performance and reduced resource utilization have risen exponentially over the years. Recent advancements in Field Programmable Gate Array (FPGA) design technology, has resulted in FPGA(s) becoming the preferred platform for evaluating and implementing signal processing algorithms.

Keywords: *Distributed Arithmetic, FIR, Pipeline, LUT, FPGA.*

1. INTRODUCTION

Digital filters are the essential units for digital signal processing systems. Traditionally, digital filters are achieved in Digital Signal Processor (DSP), but DSP-based solution cannot meet the high speed requirements in some applications for its sequential structure. Nowadays, Field Programmable Gate Array (FPGA) technology is widely used in digital signal processing area because FPGA-based solution can achieve high speed due to its parallel structure and configurable logic, which provides great flexibility and high reliability in the course of design and later maintenance. In general, Digital filters are divided into two categories, including Finite Impulse Response(FIR) and Infinite Impulse Response(IIR). And FIR filters are widely applied to a variety of digital signal processing areas for the virtues of providing linear phase and system stability.

The FPGA-based FIR filters using traditional direct arithmetic costs considerable multiply-and-accumulate (MAC) blocks with the augment of the filter order. However, according to Distributed Arithmetic, we can make a Look-Up-Table (LUT) to conserve the MAC values and callout the values according to the input data if necessary. Therefore, LUT can be created to take the place of MAC units so as to save the hardware resources.

This paper provide the principles of Distributed Arithmetic, and introduce it into the FIR filters design, and then presents a 31-order FIR low-pass filter using Distributed Arithmetic, which save considerable MAC blocks to decrease the circuit scale, meanwhile, divided LUT method is used to decrease the required memory units and pipeline structure is also used to increase the system speed

2. DISTRIBUTED ARITHMETIC

Distributed Arithmetic was first brought up by Crosier [1], and was extended to cover the signed data system by Liu, and then was introduced into FPGA design to save MAC blocks with the development of FPGA technology.

The N-length FIR filter can be described as

$$y =< hx >= \sum_{n=0}^{N-1} h[n]x[n] \quad (1)$$

Where h[n] is the filter coefficient and x[n] is the input sequence to be processed. The FIR structure consists of a series of multiplication and addition units, and consumes N MAC blocks of FPGA, which are expensive in high speed system. Compared with traditional direct arithmetic, Distributed Arithmetic can save considerable hardware [2] resources through using LUT to take the place of MAC units . Another virtue of this method is that it can avoid system speed decrease with the increase of the input data bit width or the filter coefficient bit width, which can occur in traditional[3]direct method and consume considerable hardware resources. Distributed Arithmetic is introduced into the design of FIR filters as follows.

In the two's complement system, x[n] can be described as:

$$x[n] = -2^b x_b[n] + \sum_{b=0}^{b-1} 2^b X_b[n] \quad (2)$$

Substituting eq.(2) into eq.(1) yields

$$\sum_{b=0}^{B-1} h[n] \sum_{n=0}^{N-1} 2^b x_b[n] = \sum_{b=0}^{B-1} 2^b \sum_{n=0}^{N-1} h[n]x_b[n] \quad (3)$$

$$\sum_{b=0}^{B-1} h[n] \sum_{n=0}^{N-1} 2^b x_b[n] = \sum_{b=0}^{B-1} 2^b \sum_{n=0}^{N-1} h[n] x_b[n] \quad (4)$$

Substituting eq.(4) into eq. (3) yields to the final form of Distributed Arithmetic

$$y = -2^b x_b[n] h[n] + \sum_{b=0}^{B-1} 2^b \sum_{n=0}^{N-1} h[n] x_b[n] \quad (5)$$

Take a close look at the right part of eq. (5), considering the limited possibility of input data, we can conserve the values of $\sum_{n=0}^{N-1} h[n] x_b[n]$ into a LUT unit and then callout the relevant value according to the input data to save MAC blocks[4]. And then the weighted sum of $\sum_{n=0}^{N-1} h[n] x_b[n]$ is calculated through shift registers, the result is $\sum_{b=0}^{B-1} 2^b \sum_{n=0}^{N-1} h[n] x_b[n]$. In signed system, the signed bit should be taken into consideration so $-2^B x_B[n] h[n]$ is also added. As a result, the final form of Distributed Arithmetic is defined as Eq. (5) and the implementation can be achieved on FPGA through LUT units. As the expatiation above, the basic Distributed Arithmetic structure can be described as Fig.1. The dotted rectangle is the register.

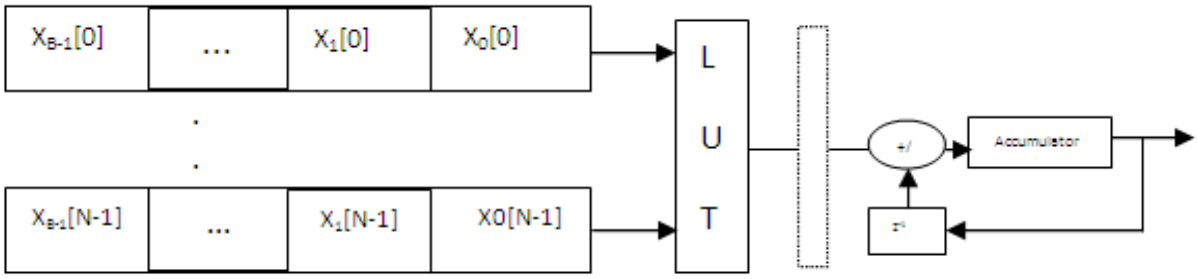

Fig.1The basic Distributed Arithmetic structure

According to the input sequence, we can conserve the coefficient values in LUT unit, the LUT constructing formula is given in Table 1.

Table 1 LUT Constructing formula

$x_b[N-1]$	$x_b[N-2]$	---	$x_b[0]$	Coefficient Value
0	0	---	0	0
---	---	---	---	---
b_N^{-1}	b_N^{-2}	---	b_0	$h[0]\bullet b_0$+---+$h[N-1]\bullet b_N^{-1}$
---	----	----	----	----
1	1	----	1	h[0]+---+h[N-1]

With above structure and coefficient values of LUT, we can achieve a variety of filters to meet various requirements.

3. FILTERS DESIGN

In the course of FIR filters design, Ringling's can be generated at the edge of transition band for the reason that finite series Fourier transform cannot produce sharp edges [5]. So windows are often used to produce suitable transition band, and Kaiser Window is widely used for providing good performance. The parameter is an important coefficient of Kaiser Window which involves the windows types. We can get a variety of windows like Rectangular window, Hamming window, Hamming window, and Blackman window with the adjustment of. A 31-order FIR low-pass filter is designed using Kaiser Window, and the parameter is as follows: =3.39, w=0.18. We can obtain the filter coefficients using Mat lab as follows.

h(0)=h(31)=0.0019;	h(1)=h(30)=0.0043;	h(2)=h(29)=0.0062;
h(3)=h(28)=0.0061;	h(4)=h(27)=0.0025;	h(5)=h(26)=-0.0050;
h(6)=h(25)=-0.0148;	h(7)=h(24)=-0.0236;	h(8)=h(23)=-0.0266;
h(9)=h(22)=-0.0192;	h(10)=h(21)=0.0015;	h(11)=h(20)=0.0351;
h(12)=h(19)= 0.0774;	h(13)=h(18)=0.1208;	h(14)=h(17)=0.1566;
h(15)=h(16)=0.1768.		

In Mat lab data is described in the floating-point form while described in the fixed-point form in this FPGA system. After quantizing the filter coefficients using 12-bit-width signed binary [6], we can obtain the final coefficients as follows:

h(0)=h(31)=4;	h(1)=h(30)=9;	h(2)=h(29)=13;
h(3)=h(28)=12;	h(4)=h(27)=5;	h(5)=h(26)=-10;
h(6)=h(25)=-30;	h(7)=h(24)=-48;	h(8)=h(23)=-55;
h(9)=h(22)=-39;	h(10)=h(21)=3;	h(11)=h(20)=72;
h(12)=h(19)=158;	h(13)=h(18)=247;	h(14)=h(17)=321;
h(15)=h(16)=362.		

With above coefficients in Xilinx, the frequency amplitude characteristics for the filter are described in fig2. we can observe that the low-pass filter has a good performance for low-pass filtering, and the cut-off frequency is 2.88 MHz if the sampling frequency is defined as 32 MHz's We can achieve the filter on FPGA according to Fig.1 and Tab.1. However, considering the even symmetry of the coefficients, we can use eq. (6) to simplify the system and the final values are defined as the input of shift registers.

$$Y[i]=x[i]+x[31-i], \quad i=0,1...15 \qquad (6)$$

However, with the increase of filter order, the scale of LUT will increase dramatically [7], which will cost more time to

look up the table and more memory to store the values. Therefore, we can divide the LUT unit into four small LUT units to solve this problem[8]. Then the values of the four divided LUT units are added as the final value. Coefficient values of small LUT is given in Table2.

Table 2 Coefficient Values of LUT

$b_3b_2b_1b_0$	Data
0000	0
0001	h[0]
0010	h[1]
0011	h[0]+h[1]
0100	h[2]
0101	h[0]+h[2]
0110	h[1]+ h[2]
0110	h[0]+ h[1]+ h[2]
0111	h[3]
1000	h[0]+ h[3]
1010	h[1]+ h[3]
1011	h[0]+ h[1]+ h[3]
1100	h[2]+ h[3]
1101	h[0]+ h[2]+ h[3]
1110	h[1]+ h[2]+ h[3]
1111	h[0]+ h[1]+ h[2]+ h[3]

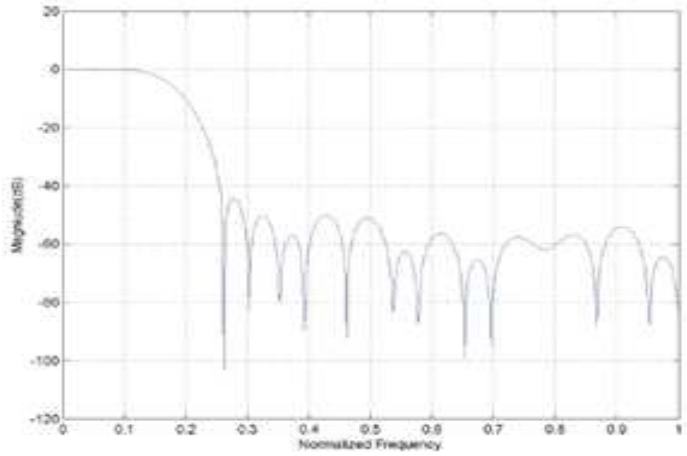

FIG: 2 CHARACTERSTICS OF FIR FILTER

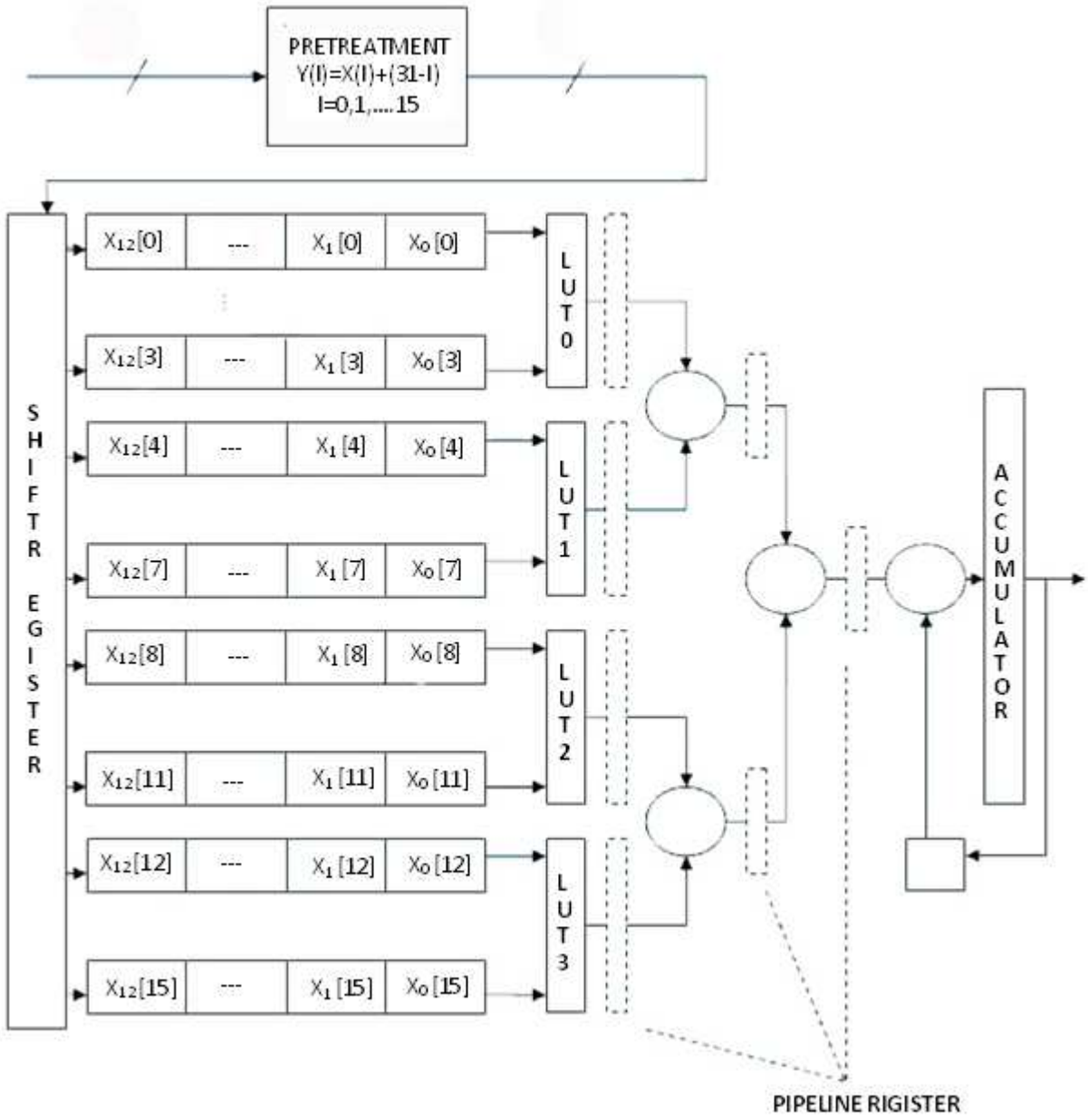

FIG: 3 STRUCTURE OF FIR FILTER BASED ON DISTRIBUTED ARITHEMETIC

IV RESULTS

Xilinx is used as the simulation platform. We can analysis the changes between the input wave and the output wave to observe the permanence of the designed filter through Xilinx, while observing the real-time implementation performance of FPGA through Xilinx. In Fig.4, the waveforms are in the Xilinx Model. By using the basic formula we are assigning the values. As we have said MAC i.e.by multiplying and adding the corresponding coefficient values the final resultant value of fir filter is shown. We can conclude that the filter coefficients are suitable for the test.

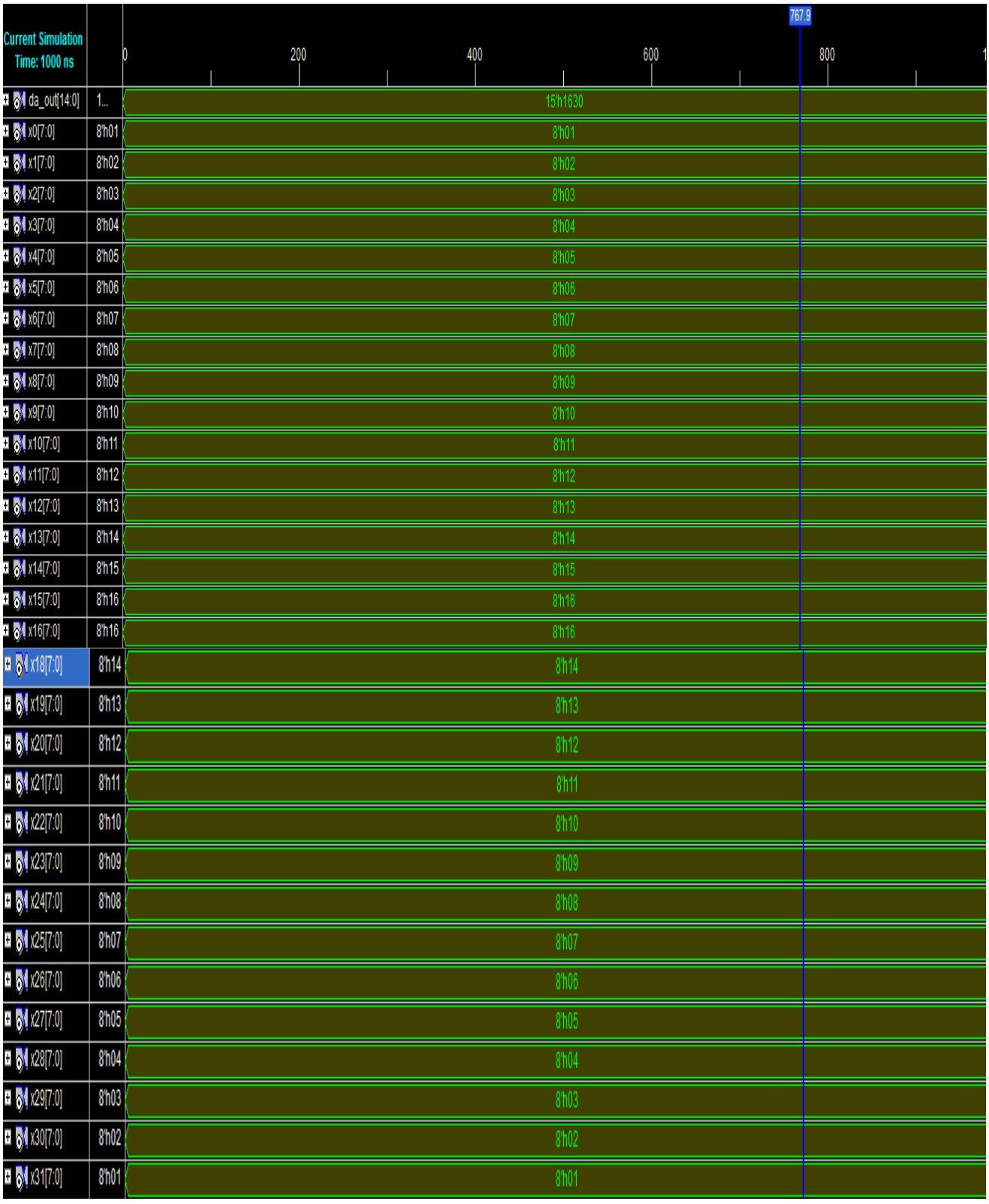

Fig: 4 simulation waveforms for FIR filter

5. CONCLUSION

This paper presents the design and implementation based on Distributed Arithmetic, which is used to realize a 31-order FIR low-pass filter. Distributed Arithmetic structure is used to increase the resource usage while pipeline structure is used to increase the system speed. The test results indicate that the designed filter using Distributed Arithmetic can work stable with high speed and can save almost 50 percent hardware resources. Meanwhile, it is very easy to transplant the filter to other applications through modifying the order parameter or bit width and other parameters, and therefore have great practical applications in digital signal processing. Further this FIR filter has many applications in audio, video signal processing, Digital Signal Processing

REFERENCES

[1] Uwe Meyer-Baese.Digital signal processing with FPGA [M]. Beijing:Tsinghua University Press,2006:50~51

[2] Tsao Y C and Choi K. Area-Efficient Parallel FIR Digital Filter Structures for Symmetric Convolutions Based on Fast FIR Algorithm [J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems,* 2010, PP (99):1~5.

[3] Chao Cheng and Keshab K Parhi. Low-Cost Parallel FIR Filter Structures With 2-Stage Parallelism [J].*IEEE Transactions on Circuits and Systems I*: *Regular, 2007,(54*2):280~290.

[4] Tearney G J and Bouma B E. Real-Time FPGA Processing for High-Speed Optical Frequency Domain Imaging [J]. *IEEE Transactions on Medical Imaging,* 2009, 28(9):1468~1472.

[5] Hu Guang-shu. Digital signal processing-theory, algorithm and realizes [M]. 2nd ed. Beijing: Tsinghua University Press, 2003:296~307.

[6] Chun Hok Ho, Chi Wail Yu and Leong P. Floating-Point FPGA: Architecture and Modeling [J]. *IEEE Transactions on Very Large Scale Integration Systems,* 2008, 17(12): 1709~1718.

[7] Evans J B. Efficient FIR filter architectures suitable for FPGA implementation [J].*IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, 2002, 41(7):490~493.

[8] Meher P K, Chandrasekaran S and Amira A. FPGA Realization of FIR Filters by s and Flexible Systemizations Using Distributed Arithmetic [J]. *IEEE Transactions on Signal Processing,* 2008s, 56(7): 3009~3017.

[9] Xia Yu-wen. Digital system design with Verilog [M]. 2nd ed. Beijing: Higher Education Press, 2008:102~103.

[10] Sungwook Yu and Swartziander EE. DCT implementation with distributed s]. *IEEE Transactions on Computers,* 2001, 50(9):985~991.