IJCSET |September 2012 | Vol 2, Issue 9, 1383-1385 www.ijcset.net |ISSN:2231-0711

A FPGA Implementation of Memory Efficient Distributed Arithmetic Fir Filter

Bhagyalakshmi.Basuvula, B.N.Srinivasa Rao

Avanthi Institute of Engineering & Technology,
Narsipatnam, Visakhapatam.

Abstract: Finite impulse response (FIR) filters are the most popular type of filters implemented in software. An FIR filter is usually implemented by using a series of delays, multipliers, and adders to create the filter's output. To implement fir filter based on DA(distributed arithmetic) algorithm is directly applied in FPGA(field programmable gate array),it is difficult to achieve the best configuration in the coefficients of fir filter, memory and computational speed. This paper provides the Distributed Arithmetic can save considerable hardware 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 direct method and consume considerable hardware resources. The design based on Altera chips is synthesized under the integrated environment of QUARTUS II 9.0. The results of simulation and test shows that this method reduces the FPGA hardware resources, memory size and high speed is achieved. This design has greatly improves memory efficient though FPGA realization.

Keywords: FIR filter, DA algorithm, FPGA;

I. INTRODUCTION:

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 resources through using LUT to take the place of MAC units.

Distributed Arithmetic was first brought up by Crosier, 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.

Pratical Design of DA algorithm:

The principle of DA algorithm based on FPGA LUT (look-up-table) was invented by Minx in the early 90's and effectively applied to FIR filters.

The output of linear time invariant system is expressed as fallows equation [1]

$Y = \sum_{m=0}^{M-1} a_m Xm$ …[1]

Where am is a constant factor, Xm is the input data (|Xm| < 1) Xm can be expressed as equation [2]

$Xm = -x_{m0} + \sum_{n=1}^{N-1} x_{mn} 2^{(-n)}$ …[2]

Where $x_{m0\ is\ sign}$ bit, x_m, N-1 is the least significant bits

Then output Y can be formed by substitute eq [2] in [1].

The value of x_{mn} is 0 or 1, there are 2M kinds of different. If we construct a LUT which can store all the possible combination of values [2], we can calculate the value of 2M in advance and store them in the LUT. Using x_{mn} as the LUT address signal, the shifting (2-1 operation) and adding operation are carried out on the output of the LUT. It can be realized through N-1 cycles and the result of multiplication accumulation can be achieved directly. So the complicated multiplication-accumulation operation is converted to the shifting and adding operation.

The parallel computing is adopted to improve the speed of calculation. The complicated multiplication-accumulation operation is converted to the shifting and adding operation when the DA algorithm is directly applied to realize linear time invariant system. However, the scale of the LUT will increase exponentially with the coefficient. If the coefficient is small, it is very convenient to realize through the rich structure of FPGA LUT; while the coefficient is large, it will take up a lot of storage resources of FPGA and reduce the calculation speed.

Meanwhile, the N-1 cycles also result in the too long LUT time and the low computing speed. The paper presents the improvement and optimization of the DA algorithm aiming at the problems of the configuration in the coefficient of FIR filter, the storage resource and the calculating speed, which make the memory size smaller and the operation speed faster to improve the computational performance.

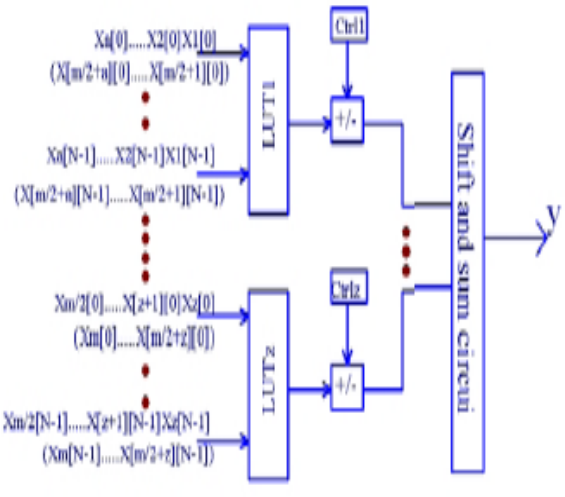

Fig.1.The circuit structure through the algorithm improvement

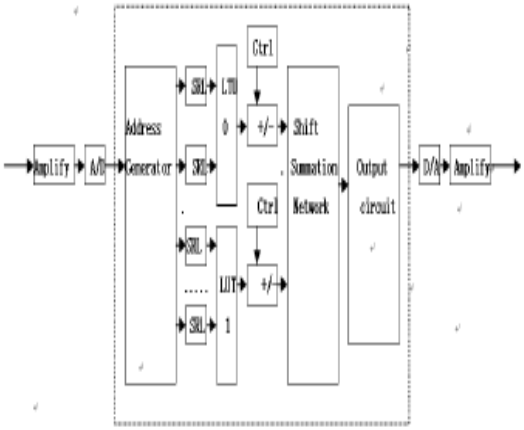

Fig.2 The circuit structure of FIR system

II. RECONFIGURABLE IMPROVED DESIGN OF THE DA ALGORITHM

The N-length FIR filter can be described as:

$$x[n] = -2^{B} x_B[n] + \sum_{b=0}^{B-1} 2^b x_b[n]$$

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 consume N MAC blocks of FPGA, which are expensive in high speed system. Compared with traditional direct arithmetic, Distributed Arithmetic can save considerable hardware resources through using LUT to take the place of MAC units [2]. 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 direct method and consume considerable hardware resources [3].

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

Distributed Arithmetic is introduced into the design of FIR filters as follows.

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

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

III. FILTERS DESIGN

In the course of FIR filters design, ringings 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 providding good performance. We can get a variety of windows like Rectangular window, Hanning window, Hamming window, and Blackman window with the adjustment. A 32-order FIR low-pass filter is designed using Kaiser window, and the parameter is as follows: X=3.39, w=0.18. We can obtain the filter coefficients using Matlab 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 Matlab 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)=h25)=-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 Matlab, the frequency-amplitude characteristic of the filter is described as Fig.3.

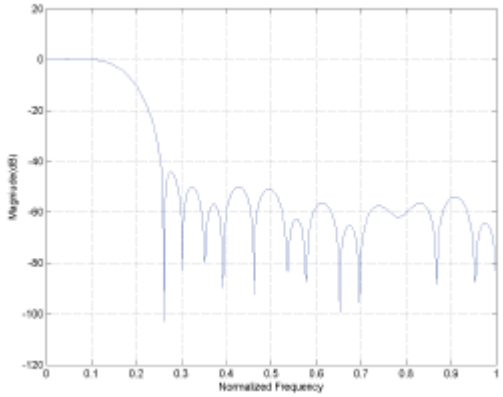

Fig.3 Frequency-amplitude characteristic of the filter

Pipeline structure is aslo used to increase the system speed. The pipelining technology is to divide combinational circuit into small parts, and then insert a register in the middle of the two parts to increase the system speed . The filter designed in this paper contains 3 level registers. Although it will increase the time delay, but helps to increase the system speed. Considering all the factors above, we achieve the new structure based on Distributed Arithmetic.

IV SYNTHESIS PROCESS:

Synthesis consists of three main steps. The first step is the *"Translation"*, which involves converting the RTL description of a design into a non-optimized intermediate representation that is used by the synthesis tool. The second step is the *"logic optimization"*, which optimizes the internal

representation by removing redundant logic and performing Boolean logic optimizations. The third step is called *"technology mapping & optimization"* which maps the internal representation to an optimized gate level representation using the technology library cells based on design constraints.

V. Synthesis Area Report:

The synthesis area report shows the total number of cells and nets in the netlist. It also uses the area parameter associated with each cell in the LSI_10K library file, to calculate the total combinational and sequential area of the netlist. The total area of the gate level netlist is unknown since it depends on total area of the interconnects, which itself is a function of the wiring load model used in physical design. The total cell area in the netlist is reported as 22978 units, which is the sum of combinational and sequential areas. The synthesis area report is shown below:

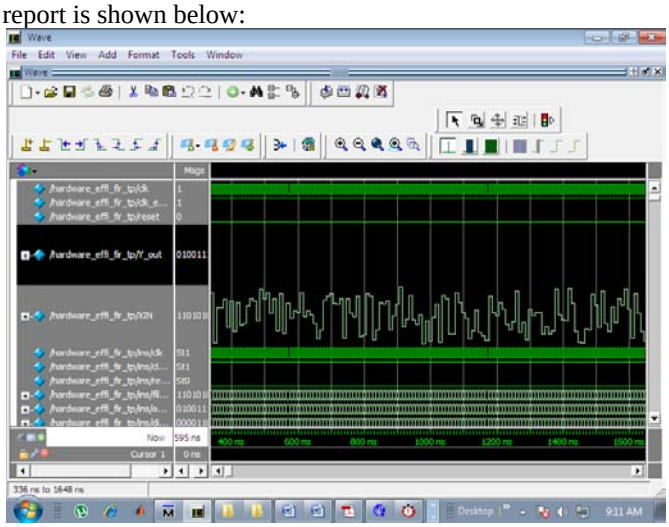

Figure 4 : Simulated output.

Area Utilization Report:

Flow Summary	
Flow Status	Successful - Mon Aug 27 21:26:21 2012
Quartus II Version	11.0 Build 208 07/03/2011 SP 1 SJ Web Editic
Revision Name	fir
Top-level Entity Name	fir_filter
Family	Cyclone III
Device	EP3C16F484C6
Timing Models	Final
Total logic elements	764 / 15,408 (5 %)
Total combinational functions	764 / 15,408 (5 %)
Dedicated logic registers	144 / 15,408 (< 1 %)
Total registers	144
Total pins	26 / 347 (7 %)
Total virtual pins	0
Total memory bits	0 / 516,096 (0 %)
Embedded Multiplier 9-bit elements	0 / 112 (0 %)
Total PLLs	0 / 4 (0 %)

Figure 5: Flow summary report

VI. Conclusion

The complicated multiplication-accumulation operation is converted to the shifting and adding operation when the DA algorithm is directly applied to realize FIR filter. Aiming at the problems of the best configuration in the coefficient of FIR filter, the storage resource and the calculating speed, the DA algorithm is optimized and improved in the algorithm structure, the memory size and the look-up table speed. The arithmetic expression has clear layers of derivation process and the circuit structure is reasonable, which make the memory size smaller and the operation speed faster. The design improves greatly compared to the conventional FPGA realization and it can be flexibility applied to implement high-pass, low-pass and band-stop filters by changing the order and the LUT coefficient.

References

[1] L. Zhao, W. H. Bi, F. Liu, "Design of digital FIR bandpass filter using distributed algorithm based on FPGA," Electronic Measurement Technology, 2007,vol. 30, pp.101-104.

[2] H. Chen, C. H. Xiong, S. N. Zhong, "FPGA-based efficient programmable polyphase FIR filter," Journal of Beijing Insititute of Technology, 2005, vol. 14, pp. 4-8.

[3] Y. T. Xu, C. G. Wang, J. L. Wang, "Hardware Implementation of FIR Filter Based on DA Algorithm," Journal of PLA University of Science and Technology, 2003, vol. 4, pp. 22-25.

[4] D. Wu, Y. H. Wang, H. Z. Lu," Distributed Arithmetic and its Implementation in FPGA," Journal of National University of Defense Technology, 2000, vol. 22, pp.16-19.

[5] L. Wei, R. J. Yang, X. T. Cui, "Design of FIR filter based on distributed arithmetic and its FPGA implementation, " Chinese Journal of Scientific Instrument, 2008, vol. 29, pp. 2100-2104.

[6] W. Zhu, G. M. Zhang, Z. M. Zhang, "Design of FIR Filter Based on Distributed Algorithm with Parallel Structure," Journal of Electronic Measurement and Instrument, 2007, vol. 21, pp. 87-92.

[7] W. Wang, M. N. S. Swamy, M. O. Ahmad, "Novel Design and FPGA Implemention of DA-RNS FIR Filters," Journal of Circuits Systems and Computers, 2004, vol. 13, pp. 1233-1249.

[8] P. Girard, O. Héron, S. Pravossoudovitch, and M. Renovell, "Delay Fault Testing of Look-Up Tables in SRAM-Based FPGAs," Journal of Electronic Testing, 2005, vol. 21, pp. 43-55