Filter Design Based on DSP Builder

Ding Ji-feng, Xu Shuang*, Zhang Jun-xing and Yang Ya-ning

Information & Communication Engineering, Dalian Nationalities University, Dalian Liaoning 116600, China

Abstract: Presents a method to realize FIR filter based on DSP Builder, FIR design using a relatively independent function of circuit module and subsystems in the design, it can easily realize the filtering function of FIR, simplified the design, modular design method greatly shortens the development cycle, make the system have been optimized in the chip area and timing analysis etc. Experimental results show that the filtering performance of the filter is good, and completely suitable for use in practical engineering in various signal processing.

Keywords: DSP Builder, FIR, signal processing, time series analysis.

1. INTRODUCTION

Digital filter can be divided into two categories according to the time domain characteristics of unit impulse response function: Infinite Impulse Response (IIR) filter and Finite Impulse Response (FIR) filter. Compared with the IIR filter, FIR filter has the advantages of exact linear phase, easily implemented in hardware and system stability, and are widely used in digital audio, image processing, satellite navigation and military communication fields [1-3].

At present, there are many kinds of main implementation method of FIR digital filter. The design method based on DSP chip adopt a unique internal hardware structure to realize the filtering algorithm, it not only need instruction programming but also debugging online simulation real-time on DSP hardware, so the development process is very complex [4]; the design method based on MATLAB and DSP hardware aided design using MATLAB to extracting the filter coefficients, but needed to write special instruction code for digital signal processing on DSP, so dependency on hardware, poor portability; the design method based on FPGA chip uses the VHDL language for algorithm design, the hardware can be modified and the code does not depend on the chip, but the code simulation and debugging is relatively complex and the development cycle is relatively long [5, 6]. Therefore, this paper puts forward a design method based on DSP Builder. This method works in Simulink environment of MATLAB software, using Altera's DSP Builder toolbox to realize system-level graphical model simulation, and ultimately generate VHDL code for FPGA download. The method can flexibly designed filter structure, and easy to develop, in addition easy function expansion and upgrading.

2. THE BASIC PRINCIPLE OF FIR FILTER

The difference equation expression of FIR filter is:

$$y(n)=\sum_{k=0}^{k-1}h(k)\bullet x(n-k) \tag{1}$$

Among it, N is the filter order, the unit impulse response $h(n)$ is the filter coefficients, $x(n)$ is the input signal, and $y(n)$ is the output signal after filtering. Seen from equation (1), FIR filtering algorithm is essentially a multiply-accumulate operations. Accordingly shows the realization of direct form structure of FIR filter is relatively simply. As shown in Fig. (**1**), input signal $x(n)$ weighted with the appropriate coefficient $h(k)$after different delay factor z^{-1}, then the products are added to get the output signal $y(n)$. $x(n)$ is the digital sampling sequence obtained by the A/D converted, $y(n)$ is an output digital sequence after filter, in this structure, the key to realize the FIR filter is the determination of coefficient $H(k)$ and the realization of delay factor z^{-1} [7].

3. DSP BUILDER DESIGN OF 16 ORDER FIR FILTER

3.1. Filter Design Process

When using DSP Builder to complete the design, firstly use FDATool of MATLAB design filter and generate a filter coefficient, then build the model file in Matlab/Simulink software (.mdl), simulation design in Simulink and scope module can be used to monitor the simulation results. DSP Builder Signal Compiler module reads the Simulink modeling file built by DSP Builder and Mega Core module (.mdl), generate the VHDL file and the tool command language (Tcl) script, DSP Builder Testbench module generates test bench file and automatically invoke modelsim for functional simulation, timing simulation using Quartus II. The design

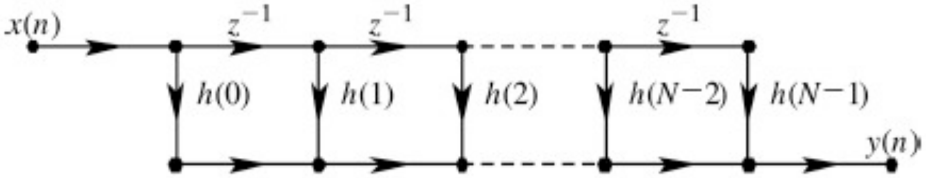


Fig. (1). FIR filter structure.

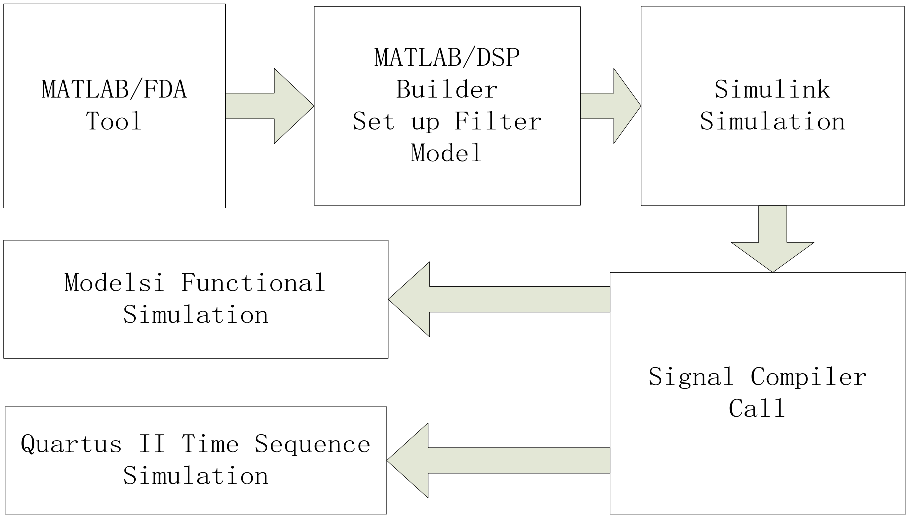

MATLAB/FDA Tool
MATLAB/DSP Builder Set up Filter Model
Simulink Simulation
Modelsi Functional Simulation
Signal Compiler Call
Quartus II Time Sequence Simulation

Fig. (2). Process of FIR filter.

process of FIR filter based on DSP Builder is shown in Fig. (**2**).

3.2. Filter Parameter Selection

Use the special toolbox of FDATool of filter design provided by MATLAB can easily design the filter coefficient [8]. Complete the corresponding parameters according to the technical requirements, setting and click DesignFilter to design FIR filter which meet the requirements, take the 16 order FIR low-pass filter as an example, set the sampling frequency of 4 Hz, cut-off frequency of 0.2 Hz. Because the filter coefficient calculated by FDATool as a symbol of decimal, therefore requires quantification and normalization for signed integer for FPGA implementation. Set the 8 bit wide as an example, the filter coefficients obtained after treatment is: h(1)=h(16)=-21;h(2)=h(15)=-20;h(3)=h(14)=-9;h(4)=h-(13)=11;h(5)=h(12)=38;h(6)=h(11)=66;h(7)=h(10)=89;h(8)=h(9)=102.

3.3. The Establishment of 16 Order Filter Model

Enter simulink in the MATLAB command window, and then create a MDL model file in the pop Simulink environment, find the Altera DSP Builder toolbox, graphically invoke Delay module of Storage Library, Product module of Arithmetic Library and Parallel Adder Substractor module of Altera DSP Builder. According to the principle structure of FIR filter design a 4 order FIR filter, change the input bit width to 9 bits, output bit width of product to 18 bits, output bit width of Parallel Adder Substractor to 20 bits, select the 4 order filter design and choose create subsystem under edit. As shown in Fig. (**3**).

Copy the four order system above, connecting the four subsystems, the determined filter coefficients are added as 16 constant ports, put the four output terminal of the four order filter subsystem connected to access an adder, thereby obtaining an output 16 order low pass FIR digital filter. Placed two different frequency sine waves in the input terminal, a frequency is set to 0.04 Hz, another frequency is set to 0.46 Hz, place an oscilloscope on the output terminal, observe the waveform of each stage, then validate weather the model correctly. Spectrum analyzer can also be placed in each port for easy observation of each waveform frequency spectrum. In addition to put signal complier and testbench. 16 order filter model is shown in Fig. (**4**).

3.4 Simulink Simulation and VHDL Code Generation

Select the start to simulate under simulation and the waveform appears as shown in Fig. (**5**), as can be seen from figure, the first wave is 0.04 Hz sine wave, the second is 0.46 Hz sine wave, the third mixing two different frequency sine waves, and the fourth is the waveform after filter, obviously, the high frequency components of mixed wave are filtered out, and simulation graphics shows that the filter realize the filtering effect.

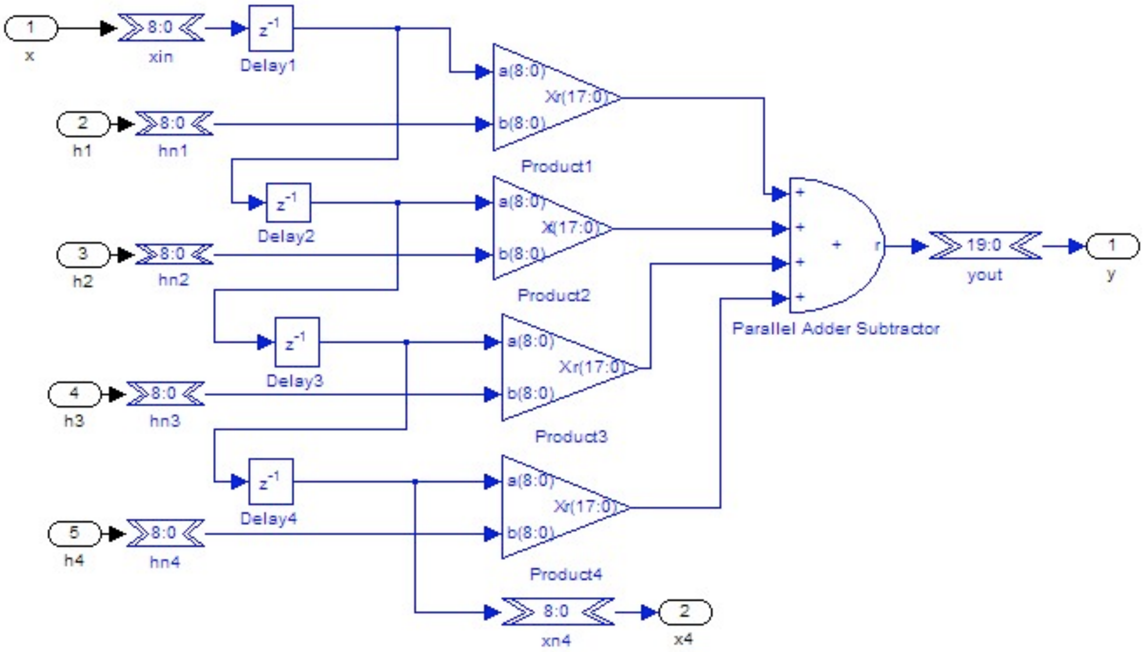


Fig. (3). Four order filter subsystem.

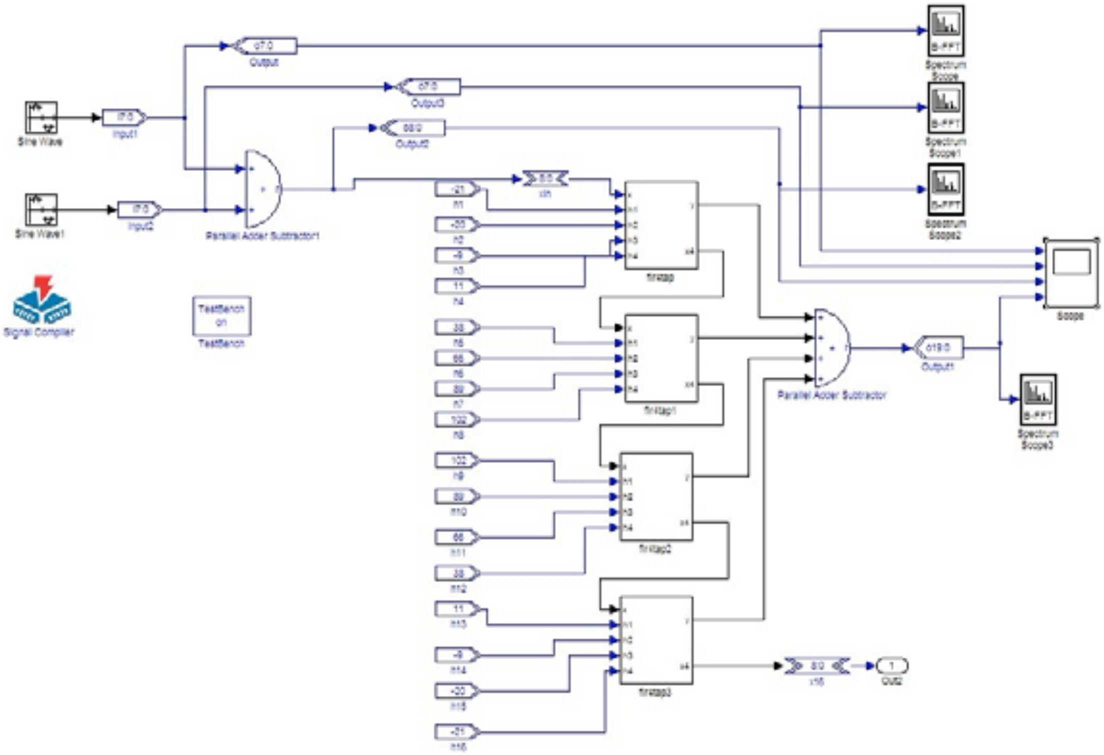


Fig. (4). 16 order filter model.

Each case of spectrum analyzer is shown in Fig. (**5**), the first frequency is 0.04 Hz, the second is 0.46 Hz, the third figure is the mixed of two waveforms, and the fourth is the spectrum after filter, it can be seen the frequency components of 0.46 Hz is filtered out.

3.5. Filter Functional Simulation

The properties of simulation carried out in Simulink is system verification, it is the simulation of MDL file, not on the generated VHDL code. Actually, generation VHDL

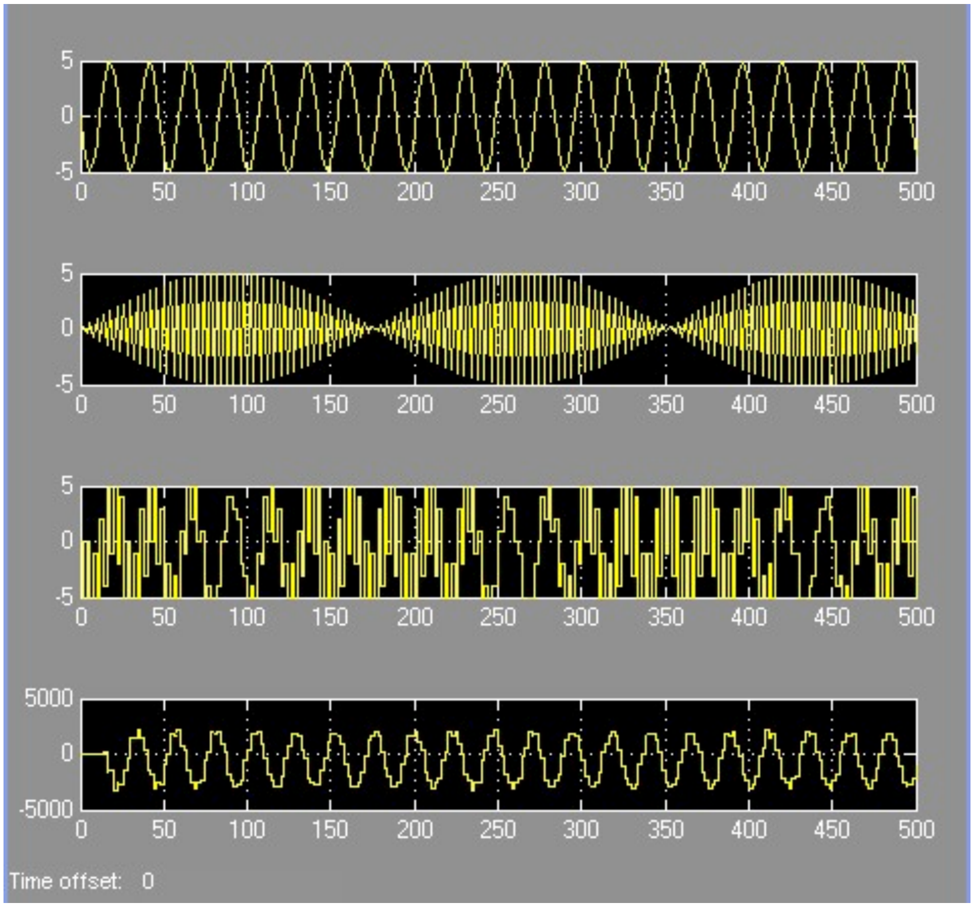
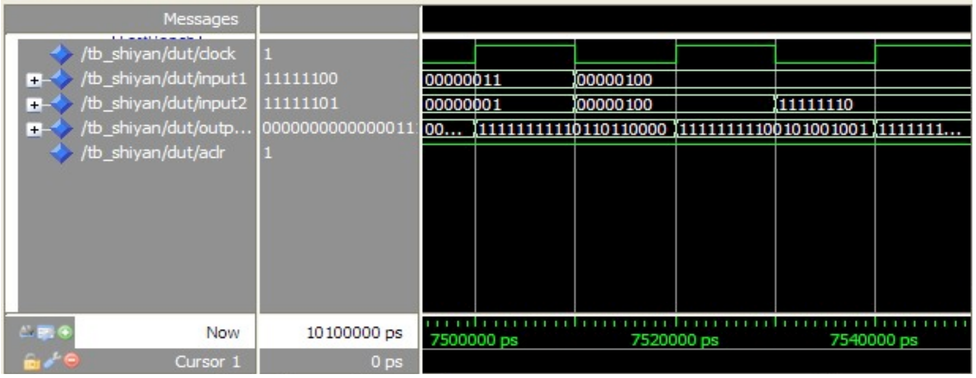


Fig. (5). Simulink simulation waveforms.



```
Messages
/tb_shiyan/dut/clock     1
/tb_shiyan/dut/input1    11111100
/tb_shiyan/dut/input2    11111101
/tb_shiyan/dut/outp...   0000000000000011
/tb_shiyan/dut/adr       1

Now                      10100000 ps
Cursor 1                 0 ps
```

Fig. (6). Modelsim simulation results.

represent RTL level, it is for the specific hardware architecture, and in MATLAB simulation model in Simulink is the algorithm level, and there may be difference between the two softwares on understanding. The VHDL code after converting is not fully consistent with the situation described by .mdl model.

To the simulation to the specific hardware architecture, it is needed to do functional simulation to the generated RTL level VHDL code with ModelSim software, find the TestBench module in the Altlab library of Altera DSP Builder toolbox and then added it to the design documents, then invoke the ModelSim simulation of graphic interface (GUI) form, its simulation setup, input parameters are consistent with simulation using Matlab/Simulink, therefore is the same as the algorithm level simulation. Modelsim is a Verilog/VHDL hybrid simulator based on single kernel, is the product of Model Technology, which is the subsidiary of Mentor Graphics. Double click the testbench, select Enable Test Bench generation, click on the second Advanced, and then click Generate HDL in proper order, finally generate simulation stimulus files; Run Simulink and rerun the Simulink simulation; Run Modelsim, and select the Launch GUI behind it, start the modelsim for functional simulation, and then appear the waveform window, as shown in Fig. (**6**).

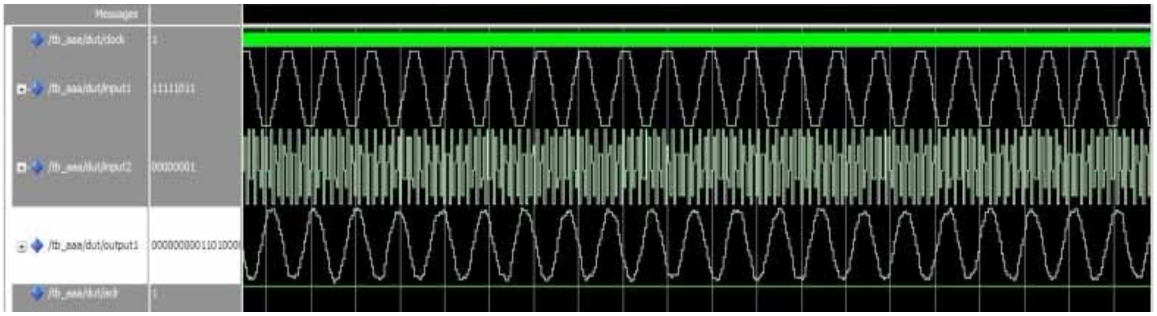

Messages
/tb_aaa/dut/clock 1
/tb_aaa/dut/input1 11111011
/tb_aaa/dut/input2 00000001
/tb_aaa/dut/output1 0000000001101000
/tb_aaa/dut/aclr 1

Fig. (7). Modelsim simulation waveforms.

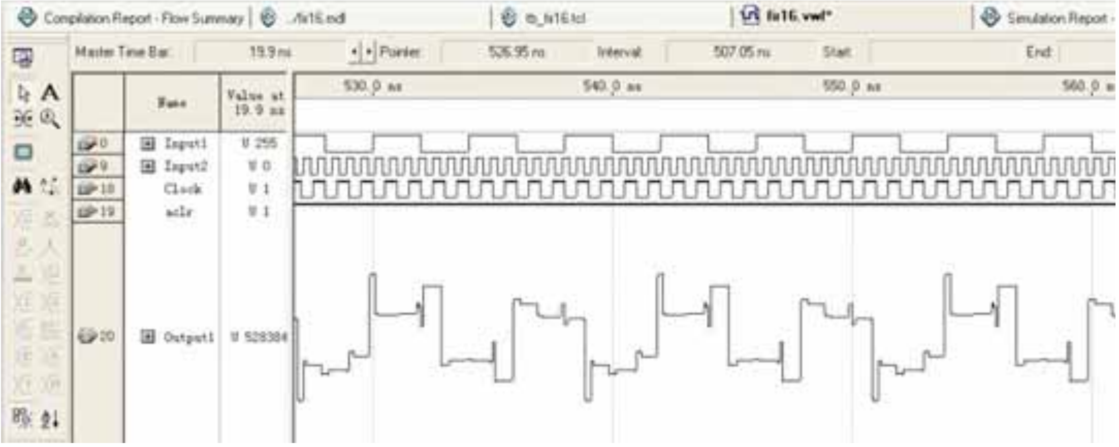


Fig. (8). Timing simulation waveforms.

The simulation results in Fig. (**6**) are not comparable with the simulation results in Simulink. So it is needed to convert the input signal and output signal of the wave window into the display format similar to the analog signal. Double click the signal and then pop wave properties window, different colors of waveform can be selected in view, set radix to symbolic; change the format under format to analog, modify the height and the maximum and minimum values of signal to the right height. Click the OK button to confirm, as shown in Fig. (**7**), it is can be seen that the results are basic consistent to the simulation results in Simulink, the high frequency waves are filtered out.

After the functional simulation, click the testbench generator of simulink the fourth compare results, and then verify whether the modelsim simulation waveforms are consistent with the simulink waveforms.

3.6. Filter Timing Simulation

RTL simulation Modelsim completed only the functional simulation, it simulation results can not reflect all the hardware characteristics of the circuit accurately, so gate-level timing simulation is still very important. Timing simulation is for concrete hardware chip, adding the time information to the chip during simulation, including setup time, hold time and critical path delay. In simulink, double click signal compiler, select the analyze, synthesis and fitter of Advanced in the second for analysis, comprehensive and adaptation, Simulink graphics will be automatically converted to VHDL language and generate programming of .pof and .sof files.

After these were completed, open the Quuartus II environment, and choose the File Open Project, located to the directory of filter model, open Quartus II project files shiyan,qpf which is established by Dsp Builder automatically. Quartus II compile specific device in the signal compile is determined by Quartus II automatically, but in actual use, the device is not the model which Quartus II selected automatically, and pin is not the pin Quartus II automatically assigned. All of these needs to be modified in the Quartus II, so there must be selected for the device number. Select Assignments Device, choosing the appropriate device in the dialog box, such as EP1C6Q240C8, and then start the compiler to execute the start compilation command under processing. Double click on the left side of the project name, opening the converted VHDL file to understand the generation process, especially the port entities. Then create a wave file, select the Vector waveform file in the other files of file-new, and then enter the Quartus II software waveform editor window. Add the input and output nodes in the Waveform Editor window by click the right mouse button, double click the added nodes to change the corresponding parameters, such as display name, bandwidth and so on. Then save this wave file. Select settings in the assignments, and then click simulation settings, add the simulation input file, select the waveform simulation file which has just set up, and click OK. Finally click start simulation in processing. After it the simulation waveform can be seen, as shown in Fig. (**8**), as can be seen from the figure, the high frequency wave is filtered out and the filter realize the filtering effect.

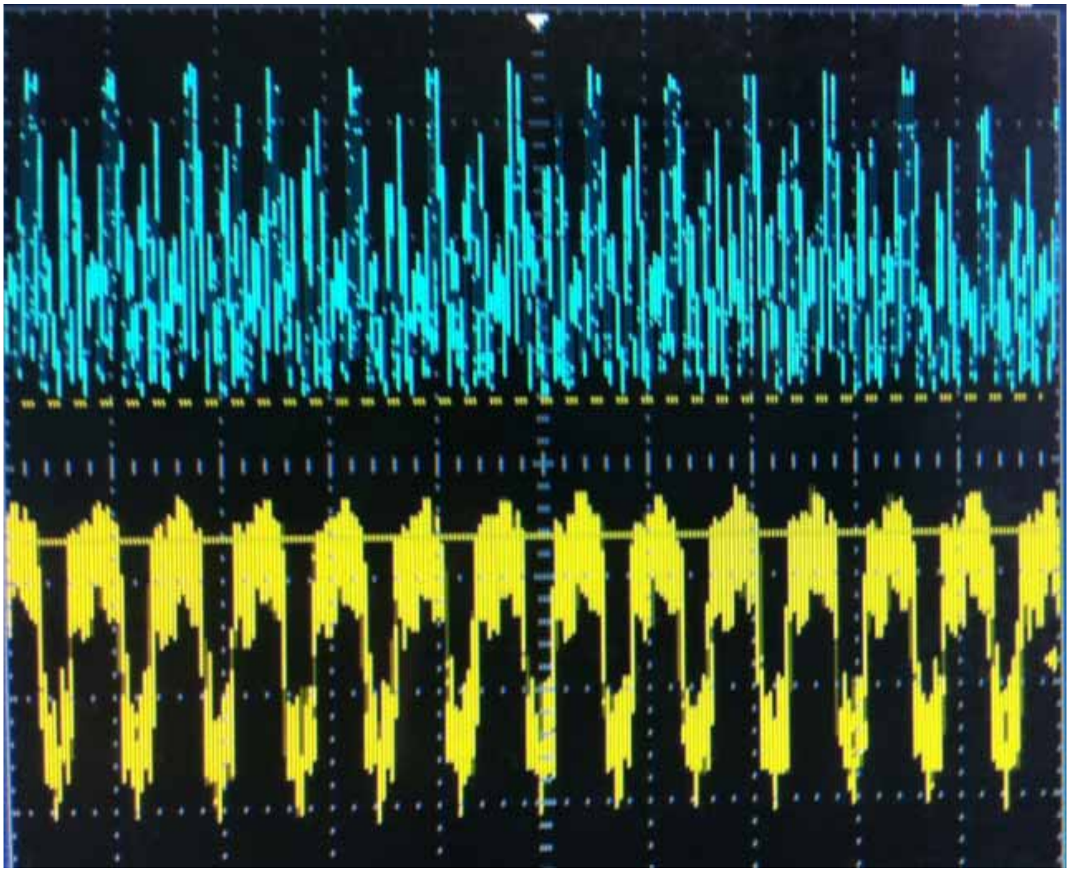


Fig. (9). Filter waveform comparisons.

4. TEST ANALYSES

The design of hardware circuit using KX_DN8+_3C55 experimental box, DA conversion circuit using DAC0832 module, the DAC0832 is a 8 bit dual channel A/D conversion chip, working frequency is 250 KHz, the conversion time is 32 us; The main chip is cyclone III series EP3C55F484C8, 55856 logic macro cell LCs (including 55856 D trigger); 2600000 programmable embedded RAM bit; 4 phase locked loop (ultra wide super high output frequency: 1300 MHz to 2 kHz). Configure the FPGA with Flash memory EPCS16. The system test waveform is shown in Fig. (**9**), among it the picture above shows mixing wave, and the yellow for the filtered waveform, although the filtering effect is not very ideal, but also plays the effect of filtering, the filtered waveform is a low frequency sine wave, and the wave contrast can also validate the role of filter. The correctness and simplicity of digital filter based on DSP Builder can be verified through the realization of the hardware circuit.

CONCLUSION

The design method of DSP Builder FIR filter using Altera DSP Builder and Simulink library graphics module for module design, configuring graphic module parameters, effectively completed the FIR filter model and model simulation. This method is convenient to operate, greatly reducing the development cycle of the filter, at the same avoid the complicated VHDL language programming, the process of simulation and verification is also simple, moreover it improves the efficiency of the development of mathematical signal processing algorithm. In general, this method will provide a wider space for FPGA chip in the application in the field of signal processing.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This works are partly supported by the Fundamental Research Funds for the Central Universities (DC13010320) and Jinzhou New Area Science and Technology Project (2013-GX1-004).

REFERENCE

[1] Z. Zheng, “Receiving system adapted to TV frequency band”, *Video Eng.*, vol. 35, no. 19, pp. 62-64, 2011.

[2] Z. Chen, “Design of driving circuit of fiber-optic transmitter HFBR-1414”, *Foreig. Elect. Measure. Technol.*, vol. 29, no. 5, pp. 79-80, 2010.

[3] Z. Zheng, “Design of an open communication teaching experiment system”, *Res. Explorat. Lab.*, vol. 31, no. 3, pp. 55-58, 2012.

[4] T. Han, “Based on VHDL the development of stepper motor controller”, *Comput. Knowl. Technol.*, vol. 29, no. 5, pp. 7300-7302, 2009.

[5] Z. Zheng, “Design of FSK modulation and demodulation system based on FPGA”, *J. Shaanxi Univ. Technol. (Nat. Sci. Ed.)*, vol. 28, no. 5, pp. 20-21, 2012.

[6] Z. Zheng, “Design of FIR Filter Based on FPGA”, *Telecom Power Technol.*, vol. 25, no. 3, pp. 45-46, 2008.

[7] H. Luo, M. Liu, and C. Wang, “Design and implementation of FIR based on DSP builder”, *Microcomput. Inform.*, vol. 25, no. 1, pp. 148-149, 2009.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014