SOPC Based Multi-Channel Sliding Correlation Processing System

Xin Liu*, Dajun Sun, Tingting Teng
Science and Technology on Underwater Acoustic Laboratory, Harbin Engineering University,
Harbin, 150001, P. R. China
*Corresponding author, e-mail: 122688188@qq.com

Abstract

Real time multi-channel sliding correlation processing is widely applied in underwater communication system or underwater positioning system. The traditional implementation in FPGA always employs parallel method which reduces the design work, however wastes considerable FPGA resources. This paper described a new kind of SOPC structure which based on AVALON bus, taken NIOS CPU as system controller, DMA used for accurate data transmission between computing units. This kind of time-multiplexed processing structure improved controlling flexibility and saved FPGA resources. System stability was proved by lake experiments.

***Keywords**: SOPC, FPGA, sliding correlation, NIOS*

1. Introduction

Sliding correlation computation is always the foundation of advanced algorithm in many fields, such as acquisition for pseudo code in DS communication system [1], [2], digital pulse compression of SAR [3], time-delay estimation in positioning system [4] etc.

The real time sliding correlation algorithm can be implemented in DSP chip or FPGA chip. Traditionally, the hardware platform of underwater positioning or communication system often composed of DSP chip for real-time signal processing and FPGA chip for logic and data transmission controlling. Due to the limitation of processing capacity, the DSP chip hardly has sufficient time to accomplish successive algorithm after a long point sliding correlation computation (more than 8192 points). The design proposed by this paper utilizes FPGA to compute real time multi-channel sliding correlation processing, liberates the DSP from correlation work, and finally improves the real time signal processing capacity of entire system.

In FPGA, real time correlation processing system is usually composed of data buffer, band pass filter, sliding correlator, low pass filter and data transmission interfaces. When the channel number increases, the traditional parallel pattern design requires more FPGA resources due to the parallel placement of modules. Although the design work is simplified, the resources required will be unacceptable when the length of correlation grows larger. Moreover， sometimes, when internal data results are required to output or control, an additional data synchronization module and data output module has to be added, causes extra waste of FPGA resources undoubtedly.

A kind of SOPC implementation raised by this paper is based on Avalon bus and NIOS CPU system. By controlling internal data-flow and hardware modules accurately, it achieves a maximum multiplicity of internal modules and memories so finally meets system demands.

2. Research Method

2.1. System Architecture

Internal hardware structure of FPGA is showed in Figure 1. System operates under control of NIOS. In consideration of flexibility and transmission efficiency， the data transmissions between computing and memory units employ directly memory access to minimize time delay; others employ DMA method for the flexible transmission parameters control.

Received November 1, 2012; Revised January 23, 2013; Accepted February 3, 2013

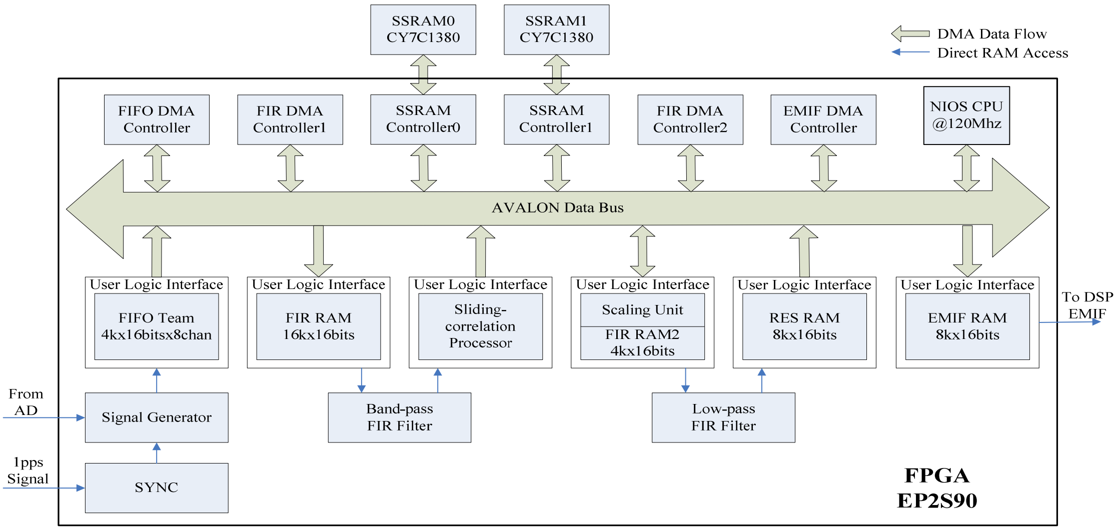

Figure 1. Hardware structure of FPGA

2.2. Processing Procedure

As mentioned above, the whole system is conducted by NIOS CPU, not only data transmission and calculation but aslo controlling of every computing units. The program flow of NIOS is illustrated in Figure 2.

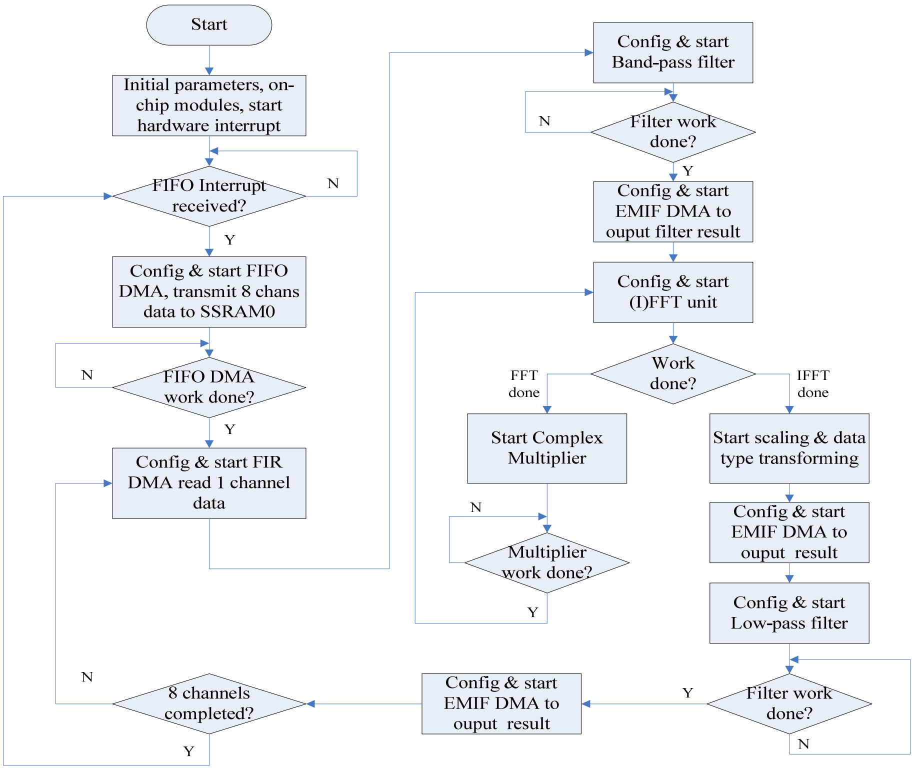
Start
Initial parameters, on-chip modules, start hardware interrupt
FIFO Interrupt received?
N
Y
Config & start FIFO DMA, transmit 8 chans data to SSRAM0
FIFO DMA work done?
N
Y
Config & start FIR DMA read 1 channel data
8 channels completed?
N
Y
Config & start Band-pass filter
Filter work done?
N
Y
Config & start EMIF DMA to ouput filter result
Config & start (I)FFT unit
Work done?
FFT done
IFFT done
Start Complex Multiplier
Multiplier work done?
N
Y
Start scaling & data type transforming
Config & start EMIF DMA to ouput result
Config & start Low-pass filter
Filter work done?
Y
N
Config & start EMIF DMA to ouput result

Figure 2. NIOS program flow diagram

Signal generator unit outputs 8 channels' acquisition data from AD. The output data is filled into 8 FIFO units, once a FIFO reaches its preset trigger condition, it will trigger NIOS' hardware interrupt. NIOS will config FIFO DMA controller, the latter then transmit every channels' data from FIFO to corresponding address in SSRAM0.

After 8 channels' raw data moved into SSRAM0, FIR DMA controller will read raw data of first channel in SSRAM0 and write it into FIR RAM which is the input data source of band pass FIR filter. NIOS then starts the filter and outputs the filter results to sliding correlation processor, which is demonstrated in Figure 3.

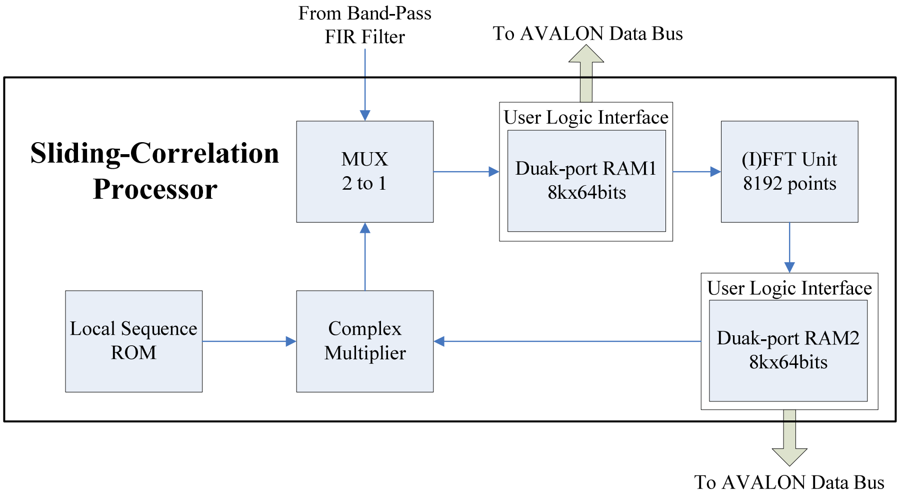

Figure 3. Structure of sliding correlation processor

NIOS open MUX unit, saves the filter's results to dual port RAM1. When the filter ends its computation, NIOS will be noticed by interrupt signal. It switches (I)FFT unit to FFT mode, leading the FFT results stored in dual port RAM2.

NIOS will get interrupt signal after FFT computation finished as well, it starts complex multiplier. This unit fetches the local signal complex data stored in ROM, and multiplies it with FFT results. This time, NIOS changes MUX's mode, saves the multiple results to dual port RAM1 again.

The (I)FFT module turned to IFFT mode after complex multiplication, the IFFT results (original unscaled sliding correlation results) will be stored into dual port RAM2. The dual port RAMs and (I)FFT unit are all time-multiplexed units.

Original sliding correlation results will be transmitted from dual port RAM2 to FIR RAM2 under FIR DMA Controller2's control. After scaled and transformed to integer, the low pass FIR filter starts to compute and the final results will be stored in RES RAM.

This design is required to output internal computed information, therefore the FIR RAMs dual port RAMs and RES RAM are all connected into Avalon data bus therefore can be accessed by EMIF DMA. Once any information from the RAMs is needed, the system can output corresponding results through changing the source address of EMIF DMA simply by NIOS.

2.4. Data Buffering

The processing procedure of FFT and IFFT are based on data block rather than data stream, so FIFOs and RAMs are needed to buffer the data. To simplify the sliding correlation processor, this design adopts overlap-save method [5], consequently the data posting and fetching should be carefully arranged.

The length of sliding correlation is 8192 points, and local sequence length is 5192. Previous 3000 points in SSRAM0 are updated from FIFO each time. The data access procedure in SSRAM0 is detailed in Figure 4.

As the Figure 4, FIFO DMA writes 3000 points to SSRAM0, the writing address added successively and circularly, hence the trigger threshold should be 3000. FIR DMA reads 8447 points of data from SSRAM0 each time, the former 255 points are used to initialize the band pass filter and the latter 8192 points are the valid data.

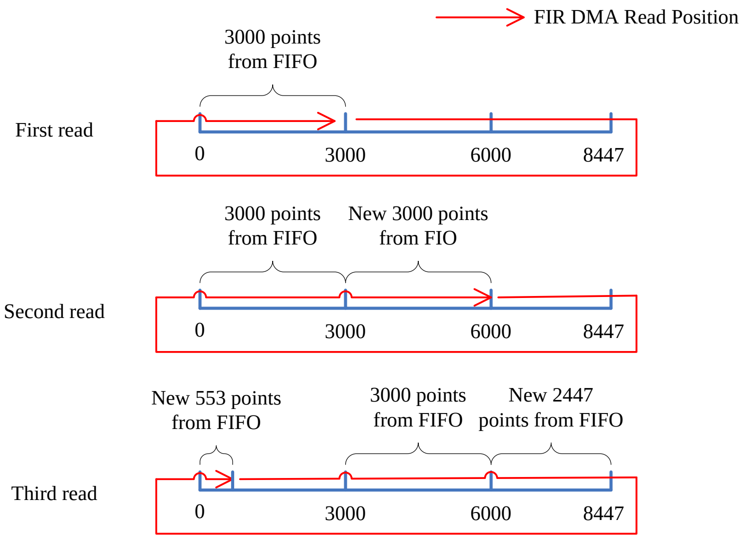

Figure 4. Address arrangement of overlap-save method

2.5. Sliding Correlation Processor

The modules in sliding correlation processor are (I)FFT IP Core with controller, complex multiplier and two dual port RAMs. (I)FFT IP Core is generated by customizing the parameters of the standard (I)FFT IP Core provided by Altera. The input data is complex signed 24bits, 8192 in length，and output data is 24bits+6bits exponents. This IP Core uses block-floating-point arithmetic internally to perform calculations and for minimizing the RAM usage, burst mode is chosen for I/O data flow.

Complex multiplier is used to multiply the data from FFT by local sequences stored in on chip ROM with 32bits width, 8192 in length. The output of complex multiplier is provided for IFFT computation. (I)FFT controller and complex-multiplier controller are written in Verilog.

2.6. Scaling Unit

(I)FFT IP Core outputs its result exponentially, therefore to improve the processing precision and reduce system resource consumed, exponents are buffered and did not take part in the previous computation. Sliding correlation processor will generate two exponents each time when computing FFT and IFFT. Scaling unit gets the final exponent by adding the two exponents together. It transforms sliding correlation result from exponential type to integral type. The integral result is scaled to reference level given by NIOS and stored in FIR RAM2 for low pass filter access conveniently.

2.7. Dual Port RAM

The dual port RAM in sliding correlation processor has a function to connect the computing unit with AVALON data bus. For directly access by either the computing unit or DMA controller or NIOS CPU on AVALON, a dual port RAM unit with AVALON interface logic is designed, the architecture is shown in Figure 5.

As is shown, NIOS CPU or DMA controller can read data through A port of the dual port RAM; the front side computing unit can write data through B port by enabling the WR_EN signal while the subsequent computing unit can read data by enabling the RD_EN signal. Compared with triple port RAM design, this plan saves a half of RAM usage, it is important especially when sliding correlation length is long (8192 points) and data width is wide (dual 24bits).

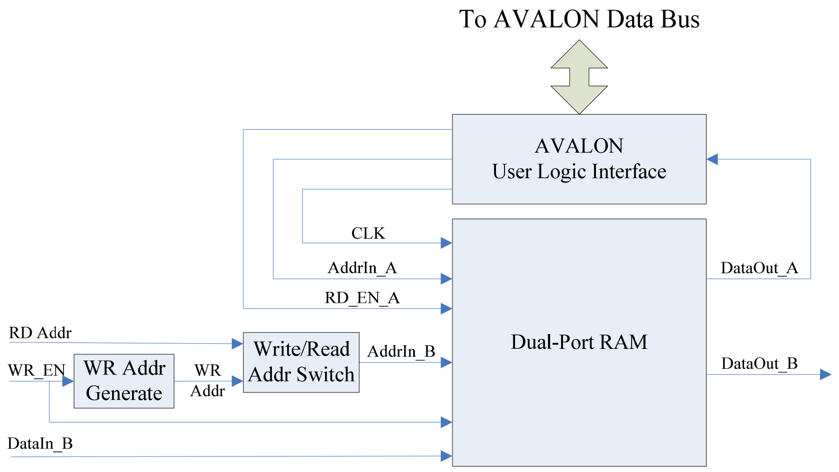

Figure 5. Inner structure of dual-port RAM

3. Results and Analysis

3.1. Resource Usage

Considerable FPGA resources is saved by time-multiplexed processing architecture and modules. Every channel's processiong and correlation processing calls a same sliding correlation processor, meanwhile the processor itself realized by time-multiplexed memory units and computing units too.

Table 1. Resources usage of key modules

EP2S90F780	Combinational ALUTS	Dedicated Logic Registers	Block Memory Bits	DSP Elements
Complex Multiplier	109	122	0	32
FIR IP Core1	12858	13933	233	0
NIOS CPU	5645	3936	2316800	0
Signal Generator	514	403	262144	0
(I)FFT8192 IP Core	2554	3894	442752	8
FIR IP Core2	10458	12714	224	0
Local Sequences ROM	34	2	262144	0
Total(%)	46	56	73	14

Table 1 shows that the RAM resources usage is larger while the logic resources and DSP elements usage is comparatively less. That means it has potential to implement more algorithm or functional logic. It has two reasons for large RAM occupancy, one is due to long sliding correlation length (8192 points), secondly the implement of 8 channels' FIFO on FPGA consumed lots of RAM resources. Improvement can be carried out in future to solve the RAM usage problem, so that the system can process more channels' data in real time.

3.2. Timing Analysis

The system is required to process 8 channels' acquisition data in real time. 3000 points of data are updated each time, while the sample rate is 200KSps, so the system has 15ms to accomplish 8 channels' sliding correlation process and data output, the time sequence is shown in Figure 6.

The system is triggered by 1PPS signal. After 15ms' acquisition data is obtained，the system starts to process. When the band pass filter of first channel finished its work, NIOS send Frame Sync pulse, followed by Channel Sync pulse, which means the filter result transmission towards DSP has begun (Phase A in Figure 6). Similarly, when sliding correlation processor and low pass filter finished their work, NIOS sends Channel Sync pulses and starts to transmit corresponding results (Phase B & C in Figure 6), then the system turn around to process next channel until 8 channels' data has all been processed and transmitted. 8 channel's process and transmission will spend the system 24.38ms-15ms=9.38ms, much less than 15ms, so this system meets its timing requirements and has potential to extend.

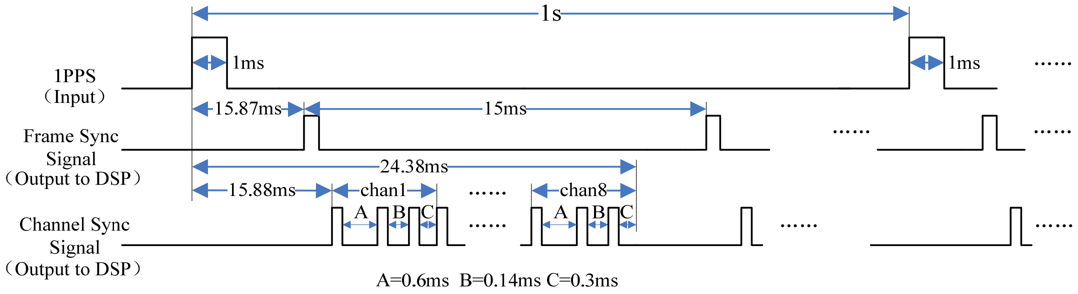

Figure 6. Time sequence waveform

3.3. Analysis of Real Tests

The system can be fully tested with LFM signal stored in ROM. Parameters of LFM is listed in Table 2.

Table 2. LFM Signal Parameters

Item	Signal Type	Band(kHz)	Length(mS)	SampleRate(kHz)	Amplitude
Value	LFM	9-14	25	200	±32767

The results obtained from FPGA of LFM signal passed through band pass filter, sliding correlation processor and low pass filter are shown in Figure 7 to Figure 9. And the comparison between FPGA results and MATLAB results is presented in Figure 10.

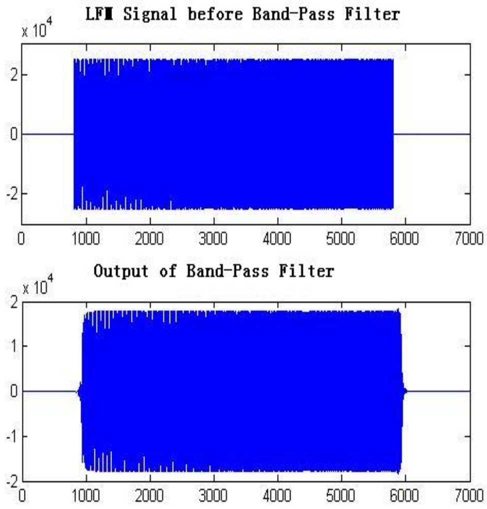

Figure 7. Original LFM and results after band pass filter

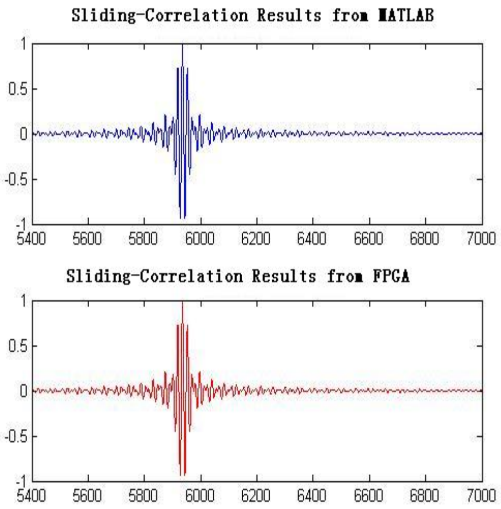

Figure 8. Sliding correlation results simulated by MATLAB and computed by FPGA

Figure 10 shows the totally system error in dB after sliding correlation and low pass filter compared with MATLAB. It can be seen that the maximum error is approximately -80dB after sliding correlation, and this error maintains the same level after low pass filter, so conclusion can be made that the precision of this system fully satisfies the demands.

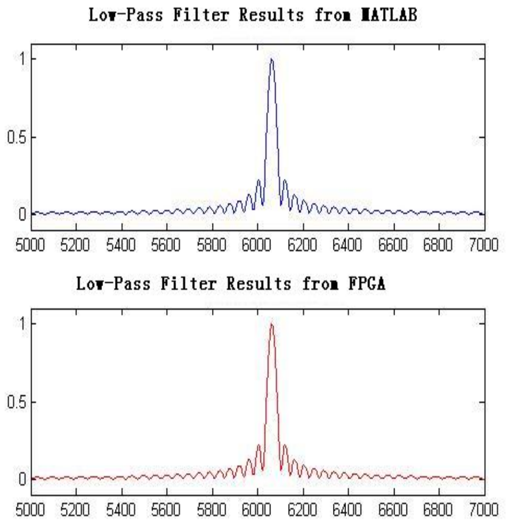

Figure 9. Low pass filter results simulated by MATLAB and computed by FPGA

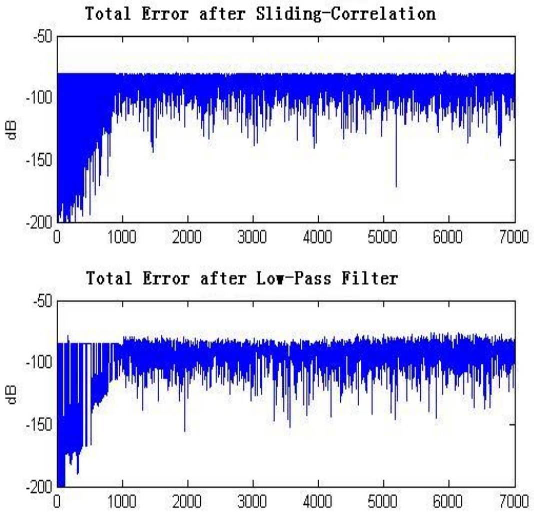

Figure 10. Total error after sliding correlation and low pass filter

4. Conclusion

The paper discussed implement of multi-channel real time correlation processing system in FPGA. The advantages of this design reflect in less resource usage, higher utilization of resources and system flexibility. A kind of 8 channel 200kSps real time correlation processing system based on this design is successfully implemented on FPGA chip type EP2S90 from Altera corp. The stability is proved by lake experiment.

Acknowledgements

The financial support of National Natural Science Foundation of China (Grant No.50909029), Science and Technology on Underwater Acoustic Laboratory Foundation 9140C200406110C2001 and 2010AA093901 are gratefully acknowledged.

References

[1] Wei Li, Jinghong GUO, Xiaohu You. The Research of Mobile Station Fast Cell Search in WCDMA System. *Jiangsu Communication Technology*. 2001; 17: 1-4.

[2] Dongjin Zhu. Design and Implementaion of PN Code Acquisition Based on FPGA. *Information Research.* 2010; 36(11): 48-50.

[3] Song Han, Xiaoli Niu. Real-Time Digital Pulse Compression of SAR. *Systems Engineering and Electronics.* 2002; 24(6): 57-59, 62.

[4] Yichun Yang, Chizhou Ma, Xiaodong Li, Jing Tian. Algorithm study of fast and accurate time-delay estimation with fine interpolation of correlation peak. *Acta Acustica.* 2003; 28(2): 159-166.

[5] Chengying Lu. Study of Technique of Fast Acquisitioin Pseudo Noise Code Based on Overlap-Save Method and FFT. *Modern Defence Technology.* 2007; 35(4): 95-99.