The Implementation of 100MHz Data Acquisition Based on FPGA

Tao Lin
Southwest China Research Institute of Electronic, Equipment, P. O. Box 429
E-mail: tonylin123w@hotmai.com

Zhou Zhengou
College of Electronic Engineering, University of Electronic Science & Technology of China,
E-mail: zzo@mail.uestc.edu.cn

Abstract

A high-speed data acquisition based on FPGA and implemented in VHDL is presented in this paper. According to the requirement of a new radar system, several new technologies are adopted in the design and implementation such as Time Compression Storage and Memory Rewriting. As a result, the system performs well with low dissipation of power, simple circuit layout and high efficient utilization of memory. The acquisition system comprises four parts: ADC circuit, Data Package and Interface, Sampling Data Memory and data Flag Memory. To implement large circuit, FPGA is adopted in this data acquisition system with reconfigurable ability and constant delay feature [1].

Index Terms—Compression Sampling, field programmable gate array (FPGA), VHDL, flag, memory, Top-Down

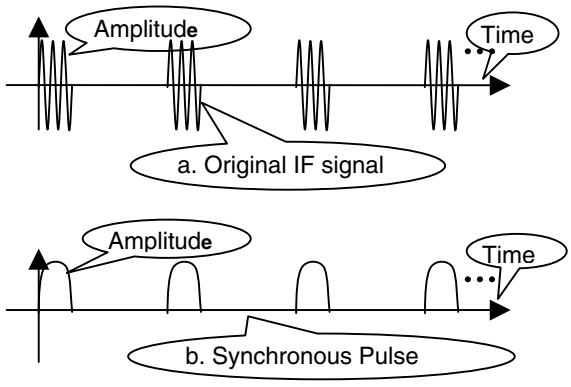
1. Introduction

In resent years, with the development of radar systems with novel technology, it is hard to receive and analyze the signal radiated by such radars in detail, through traditional means. New challenges bring the requirements that we need to know more parameter about these radars. According to the challenge, the technology of Radar Signal Fine Grain Characteristic Analysis is introduced in this area. The technology is focused on analyzing the detailed feature of intrapulse magnitude and phase of short duration pulses [2]. So, this technology makes it possible that the whole pulse of the radar signal can be analyzed within 3 areas including time domain, frequency domain and modulating domain. The analyzing process is comparable to what we called "Fingerprinting". Capturing, analyzing and parameters picking of signal of special radar can help identify the radiation source and platform correctly, swiftly and uniquely in highly dense and complex electromagnetic circumstance. So, it is applicable to store the radar signal and construct the radar signal library. But, as for the considerably high frequency of radar signal, normally GHz, it is very difficult and uneconomical in practice to analyze the radar signal directly. According to the general principle, it is possible to analyze these signals by transferring the original ones to their Intermediate Frequency for convenience of sampling, storing and analyzing.

The remainder of the paper is focused on the construction and key elements of a circuit to conform to the requirements mentioned above, a reliable, reconfigurable and well performed data acquisition system.

2. Time compression storage

In previous design, the working module of the Data Acquisition System is based on Single Shot: Once the trigger pulse comes, the system commences sampling process, regardless whether there is a signal. But comes to the low PRF, maybe there is only one pulse's information stored and even none in the worst situation. Most of the data stored in the memory are useless and those useful are lost. In order to improve the efficiency of memory, we adopt the Time Compression Storage method to reduce the storage of useless date. When the Synchronous Pulse is sent to the system, the memory begins to store the sampling data transmitted from ADC. Once the Synchronous Pulse ends, the address of the memory stops ascending along with the sampling clock. Till new Synchronous Pulse is received again, the address of memory continue to accrue and new sampling data will be stored. However, we do not know when the pulse signal arrive and where the sampling data are stored in the memory. So, it is necessary to induce the technology of Flag storage including Time Flag and Address Flag, with which we can rebuilt the signal sequence undistorted.



Proceedings of The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications ISBN 0-7695-1929-6/03 $17.00 © 2003 IEEE

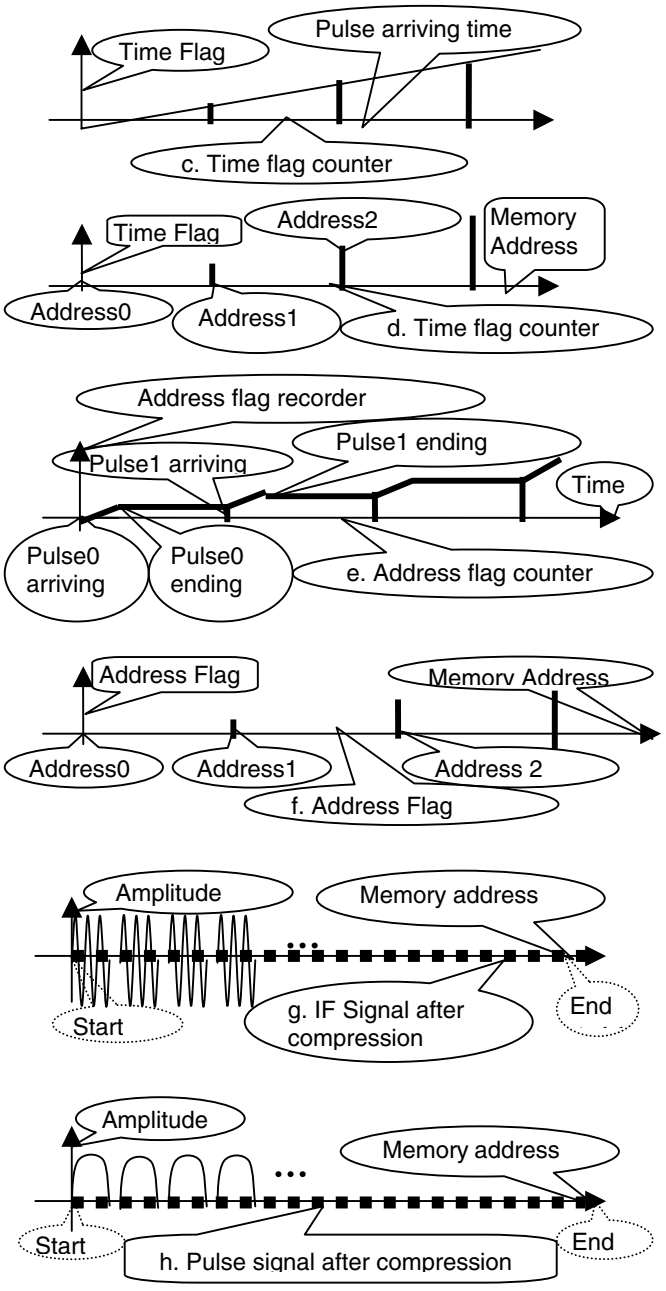


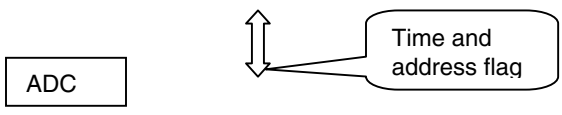
Figure 1. Time compression storage

Figure1 illustrates the patterns of Time Compression Storage. The main object of Time Compression Storage is to resolve the limitation of memory depth. As for Continuous Wave (CW) signal, there is almost no difference between Time Compression Storage and Continuous Storage. But according to pulse signal, the efficiency of the memory will vary along with the Duty Ratio if adopting Continuous Storage. In Time Compression Storage, the storage of data is available only under the control of Synchronous Pulse. But without Flag Storage, the strict relation of timing sequence will not exist, only the precedence order available. On the other hand, the exact addresses of each pulse in the memory are lost. In the interest of acquiring the specific information of the signal and between pulses, in practice, we use additional memory as the Flag Recorder. In Figure 1, the patterns show how the information is recorded. There are 2 counters and 2 recorders used for storage of the information of the pulses. Counter 1, as the time counter, begins to take count of the clock when sampling commences. The time recorder, recorder1, takes record of the output from time counter at the moment of arriving and ending of the pulse. Thus, we can rebuild the Time Interval between pulses and the width of each pulse. Counter 2, as the data memory address counter, keeps on counting the clock during the period of the Synchronous Pulse and stops counting when Synchronous Pulse ends. The output of the Address Counter, which is also used as the Data Memory address, will be noted at the beginning and the ending of the Synchronous Pulse in the Address Recorder, called Recorder 2. Therefore, the address of the Data Memory occupied by each pulse can be calculated through subtracting beginning address from the ending address.

3. Circuit structure and design

A block diagram of the whole system is shown in Figure 2. First, the Superheterodyne Receiver downconverts the preferred radar signal to intermediate frequency (IF). The IF signal is sent to data acquisition system and the particular superheterodyne receiver transmits the external video signal (RTF 10 ~ 40MHz and PW 5 ~ 800us), which is detected from the corresponding IF signal. The video pulse is used as timing and synchronous signal for the data acquisition system, which will be discussed next.

Given that the upper limit frequency of IF signal is 40MHz, it is required to sample the signal at the rate of at lest 80MHz according to Nyquist frequency. In this circuit, we use a 100MHz crystal oscillator as the sampling clock. At the front end of the data acquisition system is the A/D Converter-AD9012 manufactured by Analog Devices, Ltd. with high speed up to 100MSPS Encode Rate and 8-bit resolution. Because of the complexity of the system and the size requirement of the circuit layout, it is nearly impossible to design this system with ordinary TTL gates. The availability of CPLDs and FPGAs has made it possible to design even large high-performance systems in relatively short time and small teams of designers [3]. In this design we integrate the Control and Interface Block into a single FPGA, Virtex [4] manufactured by Xilinx Ltd.



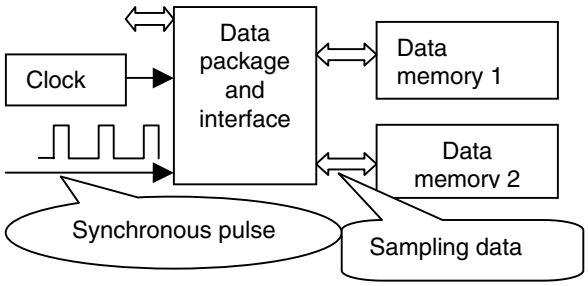


Figure 2. System Structure

In high-speed data acquisition system, the memory bandwidth is the obstacle to speed up the whole system. According to the design, the data are sampled with the clock of 100 MHz. To compensate the technical problem of sampling rate and memory bandwidth, we introduce the Multi-body-storage architecture to enhance the system memory bandwidth. The structure of Multi-body-storage is showed in Figure 2. The ADC transfers the data stream of 100MSPS to the Control and Interface Block, in which one data stream is divided into two data stream of 50MSPS in tow clock cycle synchronously. Then the divided 2 data streams are sent to different data memory respectively. In order to reduce delay, the RAMs are directly connected to FPGA [5].

In this design, another memory is adopted to store the specific information of the timing and synchronous signal for the data sampling. In order to enhance the efficiency of the data memory, we adopt Compression Storage technology in this design and we call it Flag memory. During the sampling period, the specific information of the pulse is recorded and stored in the Flag memory consequently. When sampling period ends and transmitting period begins, the flag data are read out in counter direction and packed with the related sampling data.

4. Design and implementation of FPGA

Figure 3 illustrates the time sequence of the overall data acquisition system. When the whole system is ready, data acquisition system receives the START pulse from the control system outside. The pulse advancing edge of START trigger the response the data acquisition system in which the pulse of DAQ_EN alter from ready state (‘0’) to working state (‘1’). The sampling process begins with two conditions depending on 2 signals (MODE and M_PULSE) to restrict the storage of sampling data. When the memory is full or the Data Acquisition System is idle for a certain time without input, the system will automatically generate a signal to trigger the data transmitting process, in which the Sampling Data and Flag Data will be picked out from the memory and integrated into a Data Package. After transmission, DAQ_EN is set to ‘1’ demonstrating one cycle of Sampling and Transmission coming to an end and waiting for another cycle.

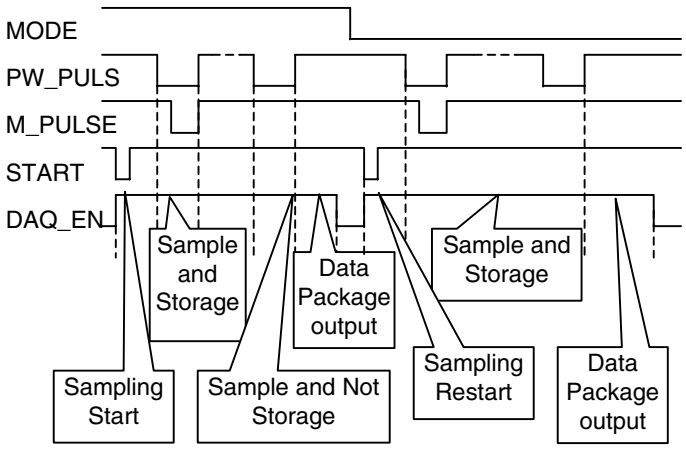


Figure 3 Data Acquisition Time Sequence

4.1. Hierarchical Methodology

The whole design mentioned above is integrated and implemented into Virtex 300 FPGA. A novel technology of Top-Down is adopted in this design. Under this method, we contrive the design through 3 layers in sequence: First, Behavior Description; Second, Register Transmission Description; Third, Logic Synthesis. In this paper, we only illustrate the Behavior Description, and others are implemented by using computer-aided design (CAD) tools. In the design of FPGA, both speed and density of FPGA are affected by the structure of the logic blocks and the interconnection between them [6]. One way to do increase the performance of FPGA is to choose logic blocks with high functionality so that the number of logic block levels in the critical path is minimized [7]. In this design, we divide the system into several function blocks with considerable independent purpose and deduce the interconnection related.

The traditional method of hardware design is focused on Graphic Design. But, it is difficult to read the original graphic file and to deduce the theory of operation. However, by using VHDL, we do not need much hardware technology and experience to read source program and analyze the design. Further more, by using VHDL, we have more flexibility using FPGAs regardless their manufacturer. In this design, from Top layer to Bottom layer, we use VHDL to describe the whole system and each function block.

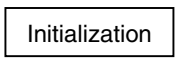
Proceedings of The 3rd IEEE International Workshop on System-on-Chip for Real-Time Applications ISBN 0-7695-1929-6/03 $17.00 © 2003 IEEE

1) Clock Generation Block: High quality, low skew programmable clocks are required in this design, as well as employ the external clock. The Virtex™ FPGA series offers up to eight fully digital dedicated on-chip Delay-Locked Loop (DLL) circuits providing zero propagation delay, low clock skew between output clock signals distributed throughout the device, and advanced clock domain control. The VHDL description is illustrated in [4].

2) Data Distributing Block: According to the hardware structure, we use tow Data Memory to store the data stream of 100 MHz converted by ADC. Therefore, the data stream should be divided into 2 streams down to 50 MHz. We use VHDL descript the progress: the single data stream is driven by Global Clock of 100 MHz and separated into 2 data stream by Global Clock of 50 MHz. Moreover, because of the delay of Synchronous Pulse after detection of IF signal, we will lose the useful information of the front-edge of the pulse signal if the storage begins right after the Synchronous Pulse. Therefore we should record the data in advance. In this design, we us a FIFO of 8×8 bits to pre-store the data of front-edge of pulse signal.

3) Address counting Block: As we mentioned above, in Time Compression Storage, we need to record the address at the beginning and the ending of the Synchronous Pulse. But whether the address should be recorded is depending on Flag Pulse (M_PULSE) and Working Module (MODE). However, M_PULSE is delayed for 100 ns than Synchronous pulse (PW_PULSE), which is showed in Figure 3. In this design, we contrive an Address Pointer (AP) to resolve this problem.

Figure 4 illustrates the process of Address Counting. When data sampling begins, the address counter keeps on waiting until the PW_PULSE comes. At first, Address Pointer (AP) records the Start Address of current PW_PULSE signal, which equals to the End Address of previous PW_PULSE signal, and transfers it to the address counter. The Address Counter continues to ascend and the sampling data are stored according to the corresponding data address until PW_PULSE ends. When PW_PULSE comes to an end, the validity of the current PW_PULSE is considered under the limitation of M_PULSE and MODE, which is mentioned above. If PW_PULSE is valid, AP is valued with the End Address of current PW_PULSE signal. At next cycle of PW_PULSE, the Address Counter will ascend basing on the End Address of current PW_PULSE. While PW_PULSE is invalid, AP is load with the Start Address of current PW_PULSE. During next cycle of PW_PULSE, current sampling data will recover the stored data of previous PW_PULSE.



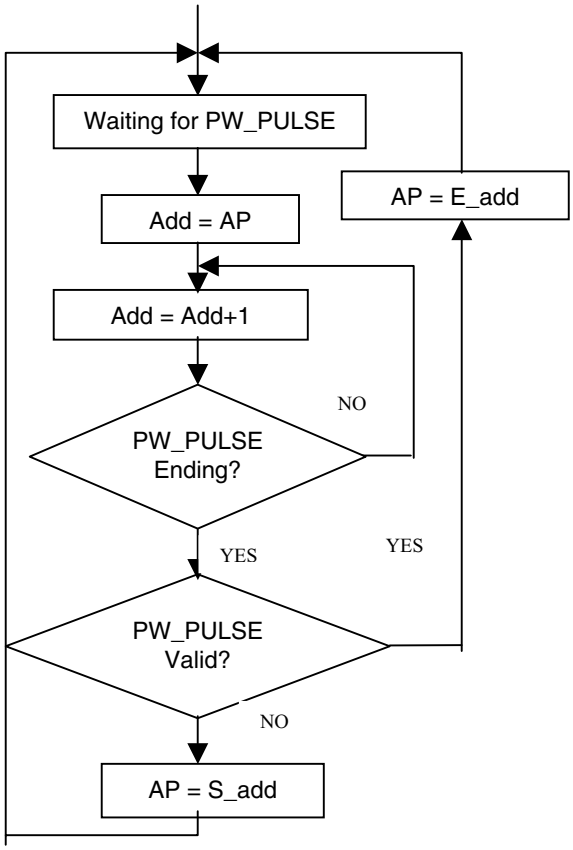


Figure 4. Address counting flowing chart

4.2. Data package

Table 1. Data Package

Data Acquisition Start Flag
Pulse Sampling Number
Sampling Data End Address
Pulse 1 Advancing Edge & Falling Edge Time Flag
Pulse 1 Advancing Edge & Falling Edge Address Flag
……..
Pulse N Advancing Edge & Falling Edge Time Flag
Pulse N Advancing Edge & Falling Edge Address Flag
Sampling Data
……..

Elicited from the Segment Mode of operation [8] for the sampling data, Table 1 shows the construction of the Data Package of output stream. The first part is the head of whole Data Package. The second part of the Data Package shows the number of pulses that have been sampled and how much space of the Data Memory occupied. Third part is the Flag Data that describe the characteristics of the pulse and pulse sequence, which is followed by Sampling Data.

5. Experimental results

In general, the comprehensive performance is valued through Signal Noise Ration (SNR) and Effective Number of Bits (ENOB). Under the condition of Over Sampling and Full Scale Rang, the formula is illustrated below:

$$SNR = 6.02N + 1.67dB + 10\log_{10}(f_s / 2f_a) \quad (1)$$

N — ADC number of bit; fs — Frequence of Sampling Clock;

fa — Frequence of Input Signal;

$$SNR_r = 20\log_{10}(SRMS / NRMS) \quad (2)$$

SNRr — Measured SNR; SRMS — Signal SNR Level;

NRMS — Noise SNR Level;

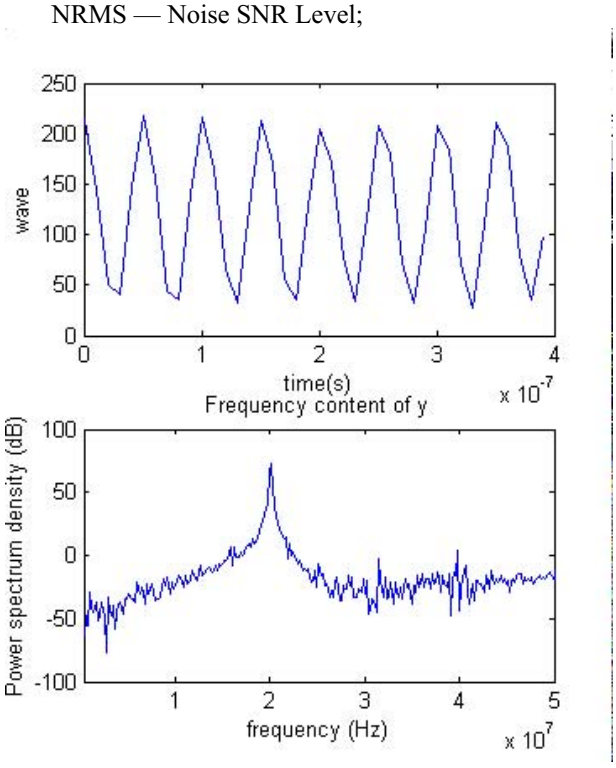


Figure 5. Wave and power spectrum density

Figure 5 illustrates the outcome of the Data Acquisition System after restoration and processing. The income is Sine-wave of 20 MHz created by Swept Signal Generator (HP 83620).

$$ENOB = \frac{SNR_r - 1.67dB - 10\log_{10}(f_s / 2f_a)}{6.02} \quad (3)$$

According to formula (1), (2) and (3), the theoretical value of SNR is 53.9 dB. In this design, ENOB is 7.47 bit.

6. Conclusion

In this paper we have described the circuit design and FPGA implementation used in the design of 100MHz Data Acquisition system. The major feature of the system is the novel Time Compression Storage, which enhances the efficiency of the Data Memory by deleting the useless information of the pulse sequence. At present, we have built and test one PCB and it is well applied in a novel system. However, there are some space left in the chip, which illustrates the explicit possibility of integrating more data processing function and bus interface between personal computer. What we do next is to contrive the PCI bus interface on chip and improve the performance of the Data Acquisition system.

Reference:

[1] Zvonko G. Vranesic, "Fundamentals of Digital Logic with VHDL Design," McGraw Hill, Hardcover, Published July 1999, 840 pages, fnd_dig_log, ISBN 0072355964

[2] Robert L. Trapp, "Automated Intrapulse RF Data Acquisition," Southeastcon '81. Conference Proceedings, Page(s): 508 -512

[3] Z.G. Vranesic et al., "The NUMAchine Multi-processor," Technical Report CSI-324, Computer Systems Research Inst., Univ. of Toronto, Canada, 1995

[4] Xilinx Inc. Xilinx Data Book 2002

[5] David M. Lewis et al., "The Transmogrifier-2: A 1 Million Gate Rapid-Prototyping System," IEEE, Transactions on Very Large Scale Integration Systems, VOL.6, NO. 2, JUNE 1998

[6] Zeljko Zilic and Zvonko G. Vranesic, "Multiple-Valued Logic in FPGA," IEEE Trans. on Computers, Vol. 44, No. 8, Aug. 1995, pp. 1012-1020.

[7] Satwant Singh, Jonathan Rose, David Lewis, Kevin Chung, Paul Chow, "Optimization of Field-Programmable Gate Array Logic Block Architecture for Speed," Custom Integrated Circuits Conference, Proceedings of the IEEE 1991, Page(s): 6.1/1 -6.1/6

[8] Custódio F. M. Loureiro and Carlos M. B. A. Correia, "Innovative High-Speed Data Acquisition Architecture," Nuclear Science Symposium Conference Record, 2000 IEEE , Volume: 2 , 2000, Page(s): 12/194 -12/196 vol.2