Research on Overall Design of Pulse Radar Signal Replay System

Yang Liu, Jing He
China Satellite Maritime Tracking and Control Department, Jiangyin, 214431, China

Abstract. In order to better meet the needs of monitoring and control tasks, this paper studies the overall design of the pulse radar IF signal playback system. It mainly introduces the hardware and software platform of the system from the overall perspective. This article provides the basis for the post-mortem analysis of test tasks and daily training.

1. Introduction
With the rapid development of the performance of spacecraft such as strategic missiles and large-scale carrier rockets, people are increasingly demanding the distance and effect accuracy of the tracking radar. In order to better meet the needs of measurement and control tasks, according to the status of the pulsed radar in the target range, it is necessary to record and play back the signals received by the pulsed radar antenna to provide basis for the experimental task and post-mortem analysis of daily training.

The system is divided into two parts: the system hardware platform and the system software, as shown in Figure 1. This article will discuss their structure and working principle. In addition, this article will also focus on the core unit of the system - pulse radar playback board structure design.

2. Hardware platform structure
In order to maintain consistency with the style of the radar system equipment, the system selected industrial control computer as the hardware design platform. There are many types of industrial
control computers, but the most widely used ones are industrial computers based on VEM bus and CPCI technology. Both of these industrial computers use the assembly of European-style cards. In all respects, the performance of the PCI local bus is much better than that of the VEM bus. The industrial control computer based on the CPCI assembly not only has the same electrical specifications as the PCI local bus, but is also more robust in structure than the PCI device. Therefore, in order to allow the system to transmit large amounts of radar data at high speed and stability, and considering that it is easy to implement a solid system structure and reliable and complete system functions, the system uses a CPCI assembled industrial computer to design the system hardware platform structure.

The system hardware platform is mainly composed of pulse radar playback boards, CPU boards, SCSI cards, disk arrays, and other peripheral modules. The data exchange medium between the modules is the PCI bus. The data transmission of the entire system is mainly divided into two parts: the first part is the disk read of data, and the other part is the PCI transmission of data. The disk read of data is mainly the computer reads the radar data of SCSI array through DMA operation of SCSI card, and send the radar data to systematic memory of CPU board through PCI bus; And the PCI transmission of the data is mainly the pulse radar board The PCI interface controller reads the cached radar data in the system memory through DMA operations, as shown in Figure 2.

![Playback system structure and data flow diagram](image)

**Figure 2.** Playback system structure and data flow diagram

Below, analyze the main components of the system hardware platform:

2.1. **Central processing unit - CPU board**

The system uses the CPU board developed by Advantech as the central processing unit of the system. It is mainly composed of a CPU, a DDR2 memory, a Bios, a graphics card, and a sound card. Devices such as SCSI controller cards, system memory, system hard disks, Ethernet, and monitors are all mounted on the CPU board. The CPU central processor controls and coordinates mutual data communications between these devices. The system uses the PCI bus as a data transmission carrier to implement communication between the pulse radar IF playback board, the CPU board, and the SCSI disk array. The pulse radar playback board and the SCSI disk array can work independently of the CPU controller.

2.2. **Disk Array.**

As we all know, the bottleneck of data transmission in almost all record playback systems is the bus data transfer rate and the disk read/write rate. RAID technology—a disk array composed of multiple SCSI hard disks is responsible for data access tasks, which can significantly increase the data read/write speed and redundancy of the system disk, and can, to a certain extent, solve the problem of the bus and disk read/write speeds. Matching problem. Because RAID0 can achieve the required performance of the system, and considering its ease of use, data read and write faster, so we use RAID0 to build a hard disk array.
Due to the limitation of the structure of the CPU board itself, the insertion of four SCSI hard disks into the RAID array of this system basically reaches the maximum data access rate of the disk. The system uses an Ultra320 SCSI hard disk, and its theoretical data rate is 80 MB/s. Therefore, the maximum theoretical data rate of 4 SCSI hard disks is 320 MB/s. After testing, the actual disk read and writes speed in the disk array is about 180MB/s.

3. **Software platform structure**

The functional design of the system requires that the radar data stored on the disk be continuously read in real time, and the radar signal can be played back in real time after processing the radar data. In this way, before designing the system control software, it is necessary to demonstrate the real-time performance of the operating system. Although the response time of the real-time system to the peripherals and the communication delay between the tasks can be short, the interaction performance is poor, the development cost is high, and the user-oriented objects are not wide enough. Windows non-real-time operating system not only has a good user interface and human-computer interaction, but also because of preemptive multi-thread scheduling strategy of Windows 2000/XP, it also has certain real-time processing capabilities. Therefore, through comprehensive consideration of the cost of development, the human-computer interaction of the system and the real-time nature of the system, this system select Windows 2000 as the software platform for real-time processing software development.

System software can be divided into two parts: user application program and driver program. They work in user mode and kernel mode respectively, as shown in Figure 3. The user application refers to the top-level user program directly to the user. It mainly provides the user with interface operations, read and write of disk data, program calls, and system control. In addition, Microsoft guarantees the security and integrity of the platform. If the user wants to access the system hardware, such as starting DMA transfer, interrupting operations, and accessing the hardware physical address, IO port, etc., it must be implemented by the management program calling the corresponding device driver. Since PCI bus data transmission is a key technology of this system, this will inevitably involve the development of PCI interface driver issues.

![Figure 3. System Software Architecture](image)

The system control part of the application program adopts multi-thread technology to realize a total of three threads in the design: main thread, interrupt thread, and file read thread. After the system playback starts, the program first starts the file reading thread. At this time, the file reading thread reads the radar data from the SCSI disk and writes it into the system memory according to the data amount of a DMA, and the file pointer identifies the reading position of the current file. The software then initiates the first DMA transfer in the main thread. At this point, the hardware side - the PCI interface controller on the playback board receives the "DMA start" command, immediately applies for control of the bus and transmits a burst of radar data written into the system memory to the cache.
by the file reader thread. In the module, waiting for the follow-up circuit to complete the data
extraction and playback; and software - the interface driver will start the ISR, and establish interrupt
threads, waiting for the arrival of the interrupt. When the interrupt thread detects the arrival of the
interrupt, first determine the type of interrupt, if the "DMA interrupt", the system will start the file
read thread, read the file pointer from the last file position to read the data again to the system buffer If
the "FIFO empty" interrupt, the software will start the next DMA. Figure 4 shows the control flow
chart of the system.

![System control flow chart](image)

**Figure 4.** System control flow chart

This article is to develop the PCI interface driver using JunDrive's complete toolkit for developing
drivers, WinDriver6.22, which is actually a repackaging of the DDK (Driver Development Kit). Below,
brieedly describe its advantages:

a). Developers do not need DDK and core state driver development experience. They only need to
use the driver interface functions generated by Windriver to complete complex tasks such as interrupts,
DMA, and memory access.

b). Windriver-specific "KernelPlugIn" can create high-performance drivers.

c). Mainstream PCI interface chips and IP vendors (such as AMCC, PLX, V3 series, Altera, PLDA,
etc.) all provide good support, and can use common software development platforms (Visual C++,
Borland C++, Java, etc.) for driver development. ;

d). Supports I/O, DMA, interrupt handling, and plug-and-play functions for various devices such as
PCI, ISA, and USB.

e). Users can run on multiple system platforms without changing the program code. Windriver6.22
supports Windows 95/98/2K/NT/XP/CE, Linux, Solaris, and VxWorks.

4. Summary

This paper first discusses the hardware platform structure of the playback system and the system data
transmission principle, and introduces the RAID array assembly method of the CPU board and the
system. Finally, the structure of the system control software and the system control principle are
analyzed.

References

[1] Integrated Device Technology. IDT72T36135M datasheet [R], 2015.9
[2] PLD Aplications. PCIX & PCI Core User's Guide(Version7.0.5) [R], 2015.11
[3] Jungo Ltd. WinDriver v6.22 User’s Guide [R], 2014.6
[4] Altera Corporation. Quartus II Version 7.0 Handbook [R], 2007.3