Timing system of HIRFL-CSR[*]

DONG Jin-Mei(董金梅)[1)]　　YUAN You-Jin(原有进)　　QIAO Wei-Min(乔卫民)

JING Lan(敬岚)　　ZHANG Wei(张玮)

(Institute of Modern Physics, Chinese Academy of Sciences, Lanzhou 730000, China)

Abstract　The national science project HIRFL-CSR has recently been officially accepted. As a cyclotron and synchrotron complex, it puts some particularly high demands on the control system. There are hundreds of pieces of equipment that need to be synchronized. An integrated timing control system is built to meet these demands. The output rate and the accuracy of the controller are 16 bit/μs. The accuracy of the time delay reaches 40 ns. The timing control system is based on a typical event distribution system, which adopts the new event generation and the distribution scheme. The scheme of the timing control system with innovation points, the architecture and the implemented method are presented in the paper.

Key words　HIRFL-CSR, timing control system, synchronization, event distribution

PACS　89.20.Kk

1　Introduction

The HIRFL-CSR project has been finished recently. It is an upgrade to the Heavy Ion Research Facility in Lanzhou (HIRFL). The HIRFL-CSR is a multi-purpose accelerator system that consists of a main ring (CSRm), an experimental ring (CSRe), and a radioactive beam line (RIBLL II) to connect the two rings[1]. The HIRFL-CSR is a synchrotron complex using a cyclotron as its injector; the relevant devices must be accurately synchronized to accomplish the task of injection, accumulation, acceleration and extraction of the beam bunches by the timing system. The timing system organizes the event series and triggers all parts of equipment participating in the process of beam handling to do the right things at the right time.

We adopt a virtual-accelerators concept in the control system which organizes the data for different accelerator cycles. A virtual acceleration is a complete set of data stored in the front-end equipment controllers necessary to execute a successful synchrotron cycle. When activated by the timing system, it becomes a real accelerating cycle[2].

After investigating the timing systems of other laboratories, we improved the scheme and design of the timing system of the CSR.

2　The timing scheme

The timing system of the CSR is based on an event distribution system that broadcasts the timing information to all the components to be synchronized. An event code is a 32 bit unambiguous time mark in the virtual accelerator. The timing system scheme is shown in Fig. 1. The event system is used to generate and distribute the sequence of events to all the relevant components. Practically the whole operation cycle is generated within the EVent Generator (EVG). The EVG sends 32 bit event codes over an optical net; the bit-stream is distributed to several branches with fan-out modules. At each destination DSP (Digital Signal Processor) card in the VME (VersaModule Eurocard) crate there is an EVent Receiver (EVR) that decodes the received event codes and performs the appropriate actions that are programmed for each event of interest to the controlled device[3]. As compared with other timing systems, there is no master timing source in the timing system. The RF system runs independently and the synchrotron phase is de-

fined automatically by the frequency and voltage pattern. In addition, the distributed signals are trigger events instead of trigger pulse signals in the timing system.
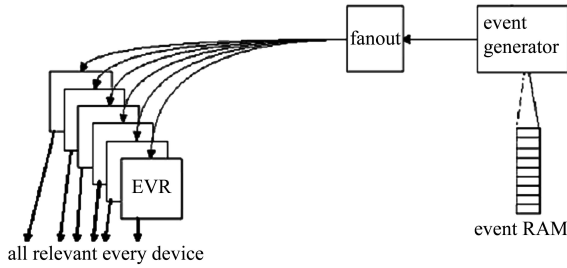

Fig. 1. Scheme of the timing system.

The RF frequency and voltage, the magnetic fields of the chopper, bumper, dipoles, quadrupoles, sextupoles, and so on, must be kept in a well-defined time relationship to keep the beam stable inside the vacuum chamber during a CSR cycle. Fig. 2 shows the timing relationship of the RF, the chopper and the bumper during the CSRm injection and capture procedure. Our design uses 2 event codes here, one is 0Xc002001 to trigger the chopper and bumpers, and another is 0Xc004001 to trigger the RF system. The delay between the bumpers and the chopper is realized by the DSP.

The wave data and event table of the devices are loaded to the front-end executable component (DSP card) in advance. If the device event table in the DSP contains an identical event code as the broadcasted event code, the DAC will start the output to the device after a time delay which is defined in the event table.

There are two merits in the timing system scheme. Firstly, the timing system is very stable and the cost of the project is low because the transmission of fiducial RF signals is not needed. Secondly, synchronization of the timing is convenient because it is independent of any other sources.
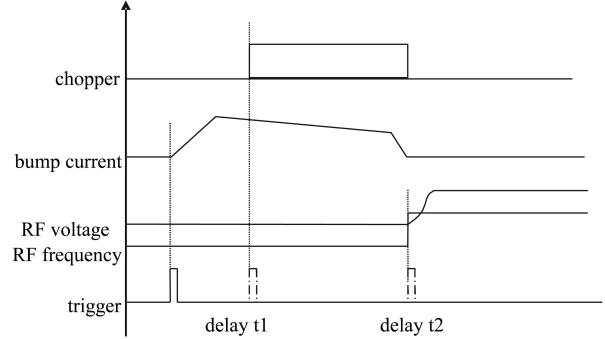

Fig. 2. Timing relationship among the devices.

3 The timing control system

The timing control system of the CSR follows a distributed architecture model[4] based on the fieldbus of the Ethernet as shown in Fig. 3. The I/O controller and the DSP card is a VME 3U-sized module inserted in a dedicated VME-crate with $+15/-15$V voltage inputs from the backplane. The I/O controller is based on an AT91RM9200 processor with FPGA (Field Programmable Gate Array), SDRAM (Synchronous Dynamic Random Access Memory), FLASH, display chip and USB interface. The I/O controller receives the data file via the net port and transfers the data via the HPI port of the DSP. The FLASH of the I/O controller is used for permanently storing the u-boot program, the Linux kernel and
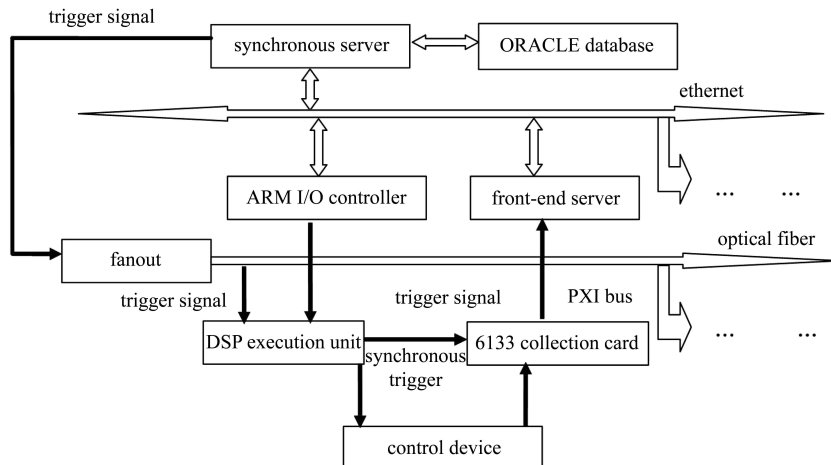

Fig. 3. Distributed control hardware architecture of CSR.

the ram-disk program. The DSP card is based on a TMS320C6713 processor with SDRAM and FLASH. The DSP processor receives the device wave and event table data from the I/O controller and stores the data in the SDRAM. The FLASH of the DSP card is used for permanently storing the digital signal processing program and the booting program. There are one event code receiving port, two DAC channels and two ADC channels in the DSP card. If the DSP card is successfully triggered by the event code, the DSP not only starts the output of two DAC channels but also from two optical channels. We designed and manufactured our own I/O controller and DSP card for a high performance-to-price ratio.

4   The synchronous server

The synchronous server is responsibility for the timing signal generation and the virtual acceleration of the data organization. The synchronous server is a standard PCI industry control computer with an installed Windows 2000 operating system. There is virtual accelerator device data calculating software and the organizing software of the CSR operation running data (OSCORD) on a synchronous server. The OSCORD is the top layer in the synchronous control system and kernel software for managing CSR operation. It takes charge of not only organizing and implementing the data of the CSR operation devices but also organizing, scheduling and implementing synchronous events for the CSR devices. It is based on synchronous hardware for every subsystem.

The data of the CSR synchronous control system include three data files: the device wave data (device process data) file, the event table file and the command table file. The calculated device wave data file, the device event table file and the command table file are stored in an oracle database via the OSCORD organizing. The device event table file is usually organized by device types; for example, with a stored beam, the ramping of magnets to steer the beam has to be done synchronously to avoid beam loss[5], so all the quadrupole magnets must response to the same device event table data. When modulating the beam, a complete set of data (device process data file, the device event table file and the command table file) is obtained from the Oracle database according to the experimental plan, and then the synchronous server transfers the data synchronously to the ftp of the I/O controller. In this way it is possible that many complete sets of data can be transferred to the I/O controller. There is a net watching process on the I/O controller, which receives the filename of the command table file from the synchronously operating server. The filename indicates which command table file has to be analyzed. Subsequently the net watching process reads the data and analyzes the content of the command table file. It gets the relevant information showing which event table file and which device process data is loaded to the DSP card. Finally the appointed event table file and device process data are loaded to the DSP SDRAM with the HPI (Host Port Interface) through the VME crate backplane. Fig. 4 shows the synchronous process control data flow. When the synchronous server completes all data transfer and data load, it can send a managed event sequence to start an accelerator cycle (all controlled devices will then work in time).
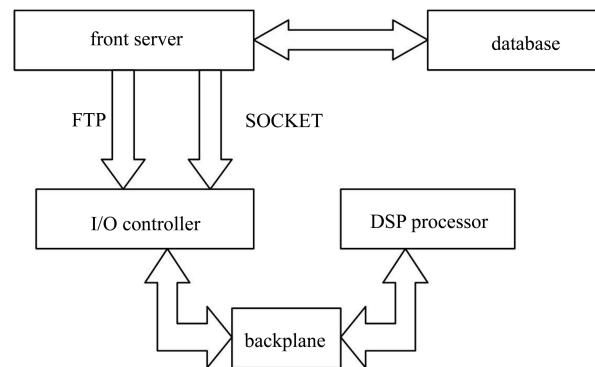

Fig. 4.   Data flow of the synchronous process control.

The structure of the device process data file is: output data sum; output data 1; output data 2; $\cdots$; output data n. Both the output data sum and the output data are unsigned 32 bit integer data; however, the range of the output data lies between 0 and 65535. The output data are sorted by time. The device process data can be calculated by characteristics of the device parameters and the beam properties defined by the experimental plan. At present, the time interval[6] of the output data of the DSP is 1 microsecond. The device process data filenames are characterized by a naming criterion, and the form of the filename is wave fn, where n is a natural number greater than 1.

The structure of the event table file is: event sum; event 1; delay 1; event 2; delay 2 $\cdots$; event n; delay n; output trigger pulse width. The event table file contains the event code and the corresponding event delay. All data are given in hexadecimal 32 bit data. The time unit of the event delay value and the DSP output trigger pulse width is 40 ns, which means that the beam timing delay adjustment precisely reaches a value of 40 ns. The event table filenames are also

characterized by a naming criterion and the form of the filename is event fn, where n is a natural number greater than 1. The value of n corresponds to the n value of the device process data filename. It shows to which device process data file the event table file corresponds.

5 Event generator

The Event Generator (EVG) is a PCI bus card inserted in the synchronous server. The firmware for the card is the VHDL (Very-High-Speed Integrated Circuit Hardware Description Language) code. Being VHDL, it is independent of the particular type of chip (FPGA) used in the hardware implementation[3]. The essential features of the event generator firmware are event RAM sequence handling, event priority determination and the possibility to send events from the software by writing into a register. There are two channels with optical signal output, used to send the event code signal. We have also created the driver program of the event generator. It can store hundreds of virtual accelerator event cycles, the start cycle to send the event code on timing and the break off cycle. The timing precision is 40 ns. Fig. 5 shows the event cycle sequence organizing GUI (Graphical User Interface). It is a constituent part of OSCORD. For physical events the GUI is used to write the event sequence cycle to the RAM on the EVG, read the event sequence cycle from RAM and start the event cycle. Moreover, the GUI can exchange data with the Oracle database.
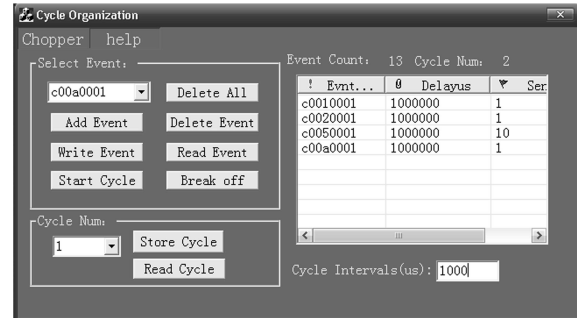

Fig. 5. Timing cycle event sequence organization GUI.

6 Summary

The timing system of the CSR has well integrated most of the required timing functions into a single subsystem. The system is layered and the hardware's functionality is captured into a device independent language (VHDL), which allows a decoupling of the functionality from the underlying technology and provides a smooth next-generation upgrade path. This can greatly enhance the capabilities and operation of a complex system. The timing system of the CSR has been constructed and its designed performance has been confirmed during the later operation; for example, by using this timing system for regulation and control, C^{6+}ion beam ramping in the CSRm has been achieved under the beam modulation scheme of stripping injection: more than 10^8 particles were accumulated, then accelerated to 1 GeV/u, and finally extracted. This indicates that the timing scheme is correct and effective.

References

1 HIRFL-CSR Project Synopsis. Lanzhou: Institute of Modern Physics, 2000. 1
2 Steiner R. Synchronisation and Timing in the GSI Control System, Internal documentation. 2002, 1
3 Korhonen T, Heiniger M. Timing System of the Swiss Light Source. ICALEPCS. San Jose: CA, 2001. 638
4 DONG Jin-Mei. Organizing Software System of CSR Operation Running Data. Lanzhou: Institute of Modern Physics, 2007. 15
5 Korhonen T. Review of Accelerator Timing System. ICALEPCS. Trieste: Italy, 1999. 167
6 YUAN You-Jin. CSRm Manual. Internal documentation. 2004