The Readout in BESIII DAQ framework

LEI Guangkun for the BESIII DAQ Group
IHEP, Beijing, CHINA

Abstract

The BESIII readout system implements an interface between DAQ framework and FEE (Front-End Electronics). As a part of the DAQ system, the readout system plays a very important role in the process of data acquisition. According to the design report of BESIII, the event rate will be about 4000Hz and the data rate will be up to 50Mbytes/sec after Level 1 trigger. The BESIII readout system consists of about 40 VMEbus crates which are divided into five branches based on sub-detectors. The principle functionality of Readout is to read, pack, buffer and forward the data coming from FEE modules in the VMEbus crates to Readout PC. The implementation of the BESIII readout is based on commercial components: VMEbus PowerPC based single board computer; the VxWorks real-time operating system. The readout crates work independently and concurrently. To synchronize the crates the state chart has been used in the readout software. In this paper, function modules, state chart and implementation methods of the BESIII readout software will be presented.

0 Overview of the DAQ (Data Acquisition) system for BES III

The event rate of Beijing Spectrometer III (commonly known as BES III) under construction is estimated to be 4000Hz. As an important part of BESIII, DAQ system must have enough performance to acquire data from the front-end electronics, process large numbers of data, and store data in limited time. To meet these requirements, several technologies are employed, such as large computer farm, high-speed Ethernet, multi-level buffering, parallel processing, and high-speed VME readout, etc.

The DAQ system of BES III can be roughly divided into two parts: the readout subsystem whose primary duty is to read the event data segments from the FEE modules and send them to readout PCs, and the online system which is in charge of collecting data, building events and data storage. The architecture of the DAQ system is illustrated in Fig.1. The bottom of figure 1 shows the readout subsystem, which consists of several readout branches. Each of the branches includes a certain number of VME crates. Above the readout subsystem is the online system, including PCs with multi CPUs and multi network cards (also known as readout PCs), file servers, PC farms, etc.

The communication between the readout subsystems and the online software is accomplished through Ethernet, with TCP/IP employed. Through 100M switches, the readout subsystem sends data to and receives commands from the readout PCs of the online system.

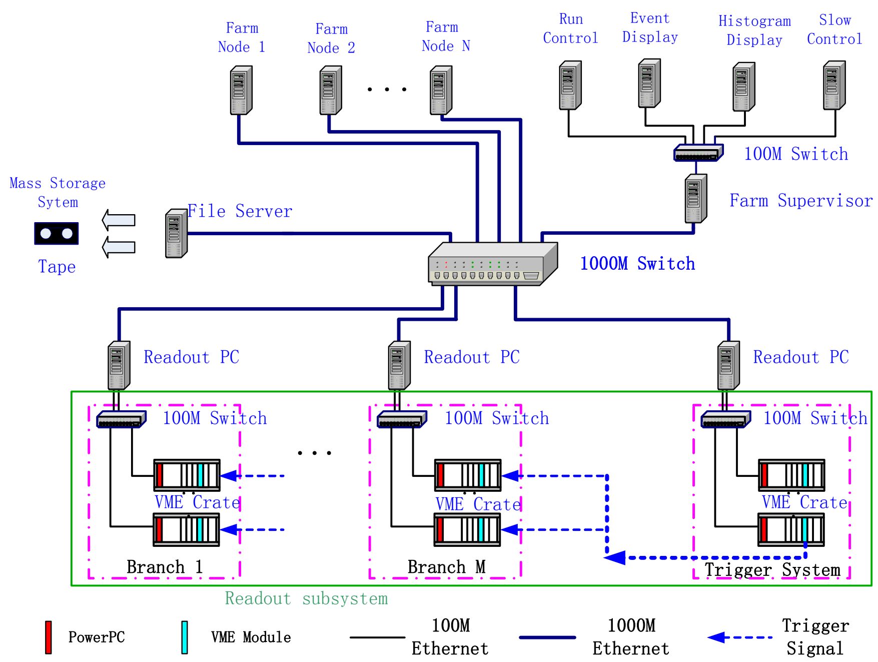

Fig. 1. Architecture of the BESIII DAQ system

1 Structure and functions of the readout subsystem

The readout subsystem is made up of several readout branches, each readout branch consists of several VME crates, the number of crates varies from one branch to another. In the DAQ system the readout subsystem is the only part that interacts with electronics modules of FEE systems and Trigger system. Not more than 16 FEE readout modules are mounted in one VME crate; some crates also have FEE control modules in. See Fig.2. Each VME crate adopts MVME5100, an embedded single board computer manufactured by Motorola Inc., as its system controller. The operating system running on MVME5100 is VxWorks (version 5.4), which is a real time operating system developed by WindRiver Inc. All the signals between the system controller and the FEE modules are transferred through VME bus. The analog signals from detectors are converted to digital signals through the FEE systems, and then the system controllers of VME crates read valid data by using CBLT (Chained BLock Transfer) method.

When the software of the readout subsystem is running, it maintains three tasks: read data from FEE module, pack these data into pre-defined data format for upload, forward packages to the readout PC, all the three tasks are parallel and have strategy for synchronization. After the system startup, MVME5100 must set the running mode, initialize FEE readout modules and control module first; it also needs collect the data from the FEE readout modules as quickly as possible.

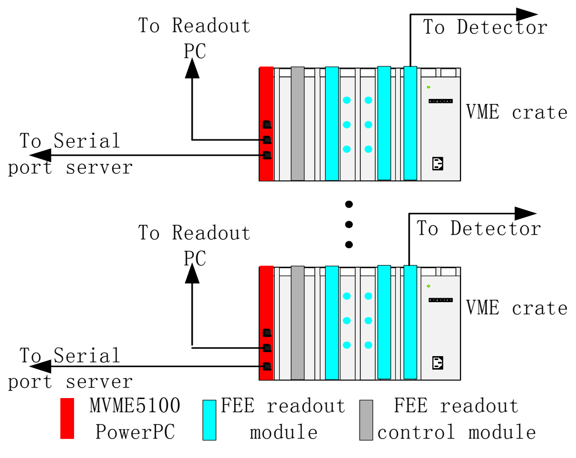

Fig.2. Structure of readout subsystem

2 Implementation of DAQ readout software

The readout subsystem is a complex system whose maintainability and robustness must be fully considered. The readout subsystem uses a state machine to control running.

2.1 State machine design

A state machine is designed for the readout subsystem of BESIII DAQ, because it facilitates the readout system to get commands from the online software and carry out corresponding actions. This state machine is Moore-like, which means that actions in it are separated from the state transition that can be regarded as instantaneous. A global variant is allocated to record the current state. After the readout system receives a command, the global variant is reevaluated to indicate the next state. The operation is just an integer assignment, so the transition can be regarded as instantaneous.

State machine of the readout system works as Fig.3. There are two big states: waiting and active. In the waiting state, readout software listens for the online software to connect to it, after online software connects to the readout software, the current state turns to be active. In fact, all the state transactions of readout system are controlled by online software.

According to the main tasks of the readout system, the active state is divided into sub-states: the Pre-TakingData state and the TakingData state. In the Pre-TakingData state the readout system sets up the system controller and the electronic modules and prepares all the system conditions ready for taking data. In the TakingData state, the data taking tasks begin to run. Pre-TakingData and TakingData sub states are also divided into more sub-sub states for system control and synchronization.

Totally, there are 12 types of commands defined in the online software: LOAD, UNLD, CONF, UNCF, PREP, STAR, SATR, SPTR, STOP, PAUS, RESU, and EXIT.

When transition occurs, operations related to this transition must be accomplished. And after carrying out these operations, an acknowledgement is sent to notify online software whether this transition success or not.

2.2 The interface between read out subsystem and online software of DAQ

The connection between Readout software and the online software is based on the network. They create three socket channels to communicate, one for receiving commands and sending acknowledgements, one for uploading system status and error information, another for forwarding the data read from the FEE modules.

2.2.1 Commands and acknowledgements socket

This socket is used to receive commands from readout PC since the readout system works according to the commands sent by online software. After receiving a certain command, the readout system turns to do the corresponding actions, and then send acknowledgement telling online software whether receive success or not.

At the pre-running state, readout system need to config the FEE hardware, so the information used to initialize and config the FEE modules also need to be downloaded through this socket. For configing and controlling the FEE modules more efficiently, uniform structures of hardware configuration are defined for readout and online system.

There are five FEE sub-systems in BES III: Electromagnetic Calorimeter (EMC), Main Drift Chamber (MDC), Muon Counter (MUC), Time of Flight (TOF) and Triger (TRG). They are partly independent, and have different working mode and control requirements to their FEE modules.

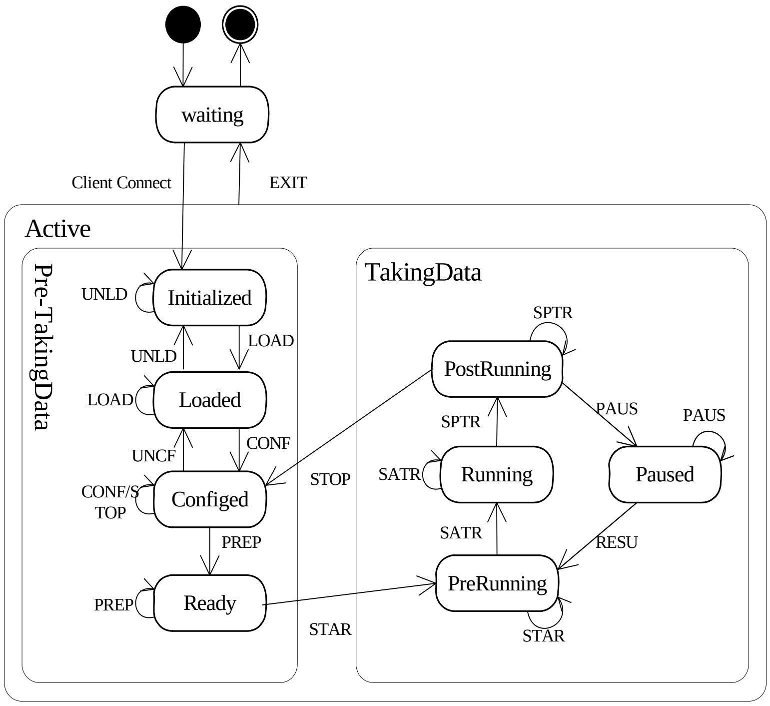

Fig. 3. The state chart of the read out subsystem

Therefore, readout system is designed to use a jumper on MVME5100 board to distinguish the FEE subsystems. After the system starts up, readout subsystem checks the jumper and confirms the subsystem first, then link the corresponding processing functions.

2.2.2 System status and error socket

If system encounters error or system status information needs to report, the package containing this information will be sent via this channel.

The detail of system status and error reporting will be discussed below.

2.2.3 Data sending socket

This socket is dedicated for the high-speed data transfer, no other information can be transferred through it. Thus, the data can be sent independently and immediately.

The purpose of using three sockets is to avoid the mutual interference and keep system more stability and efficiency.

2.3 The interface between data readout system and FEE

Readout subsystem accesses FEE modules via VME bus. Two types of information are exchanged. First, information to config and control the FEE modules. Second, event data that read out from FEE modules.

Module configuration and control is accomplished by writing well-defined data to the control registers of FEE modules via VME bus. System run mode and configuration parameters are obtained from online system.

Event data read out is based on interrupt. FEE modules get analog signals and make ADC convert, then save converted data to the module buffers. If satisfy the arranged interrupt conditions, generate interrupt to tell MVME5100 to read the data in their buffers. In order to speed up the data readout, DAQ readout system adopts CBLT（Chained BLock Transfer）mode to read data from the global buffers of FEE modules.

Each response, MVME5100 must readout same count of events, and the count is set before the read data task starts. The entire treating processes must keep consistency of the event number.

3 The data readout process

The relationship between readout system and online system and FEE system has been discussed above. Here will show the detail of control for data readout tasks.

For getting the best performance, many mechanics are used for the multi-task system. Semaphore and message queue are used for communication and synchronization between multi parallel tasks. Multi-level Ring buffer is employed to improve parallelization and stability. Here the Ring buffer is specified and is different from ring buffer provided by VxWorks library. In particular, the memory block used in system need to be continuous, besides, a mutex semaphore guarding mechanism is used in the ring buffer applications, in such a way, simultaneously accessing by multiple readers and writers would be interlocked.

As shown in the state machine, after STAR command is received, the readout subsystem does not take data immediately, it just spawns the readout task, data repack task, net upload task, and then waits there until the SATR command is received. Normally, these three tasks take most of the system resource while taking data. Readout system use two ring buffers, one for caching data before data processing, and one for caching data before net

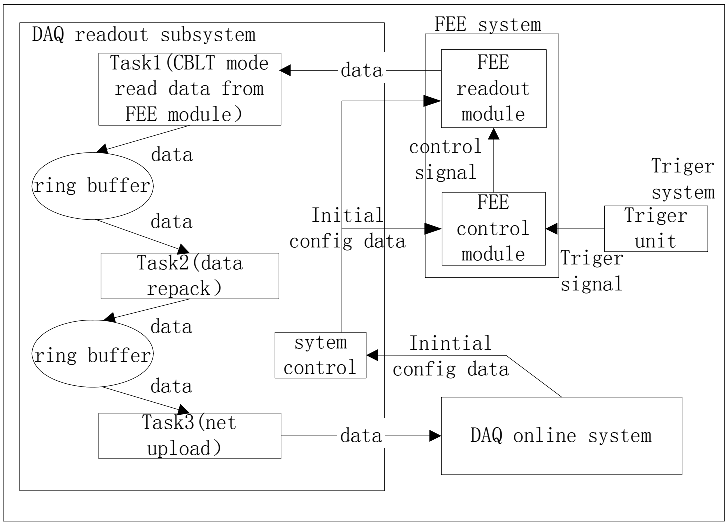

Fig.4. Data flow of readout subsystem and its interface with other correlative systems.

upload. See the Fig.4.

Operating system kernel of vxworks 5.4 has been customized and optimized in order for the readout software to get better performance.

4 Error and status report of readout system

For reporting system error and status information, a compatible structure is defined. It means that system error and status can be uploaded to online software by using similar implementation, only distinguished by a specific bit. This makes the system easy to maintain and reduces the complexity of transferring. Furthermore, the defined structure is length-alterable, it is convenient for the system to upload appending information in the debug phase.

Error report defines error level, error type and slot information for error FEE modules in the structure. If system encounters error, report function will fill in the corresponding error information quickly and send it through error and status socket.

Readout system set a timer to upload system status periodically. The status contains system memory using, CPU load, and net speed. When it's time to report status, all these information will be obtained and packed, then forwarded. These kinds of system information are gained by utilizing the SPY function of VxWorks kernel.

SNMP protocol also can do the job of reporting system error and status. Testing has got a satisfactory result.

5 Conclusions

The coding of readout system framework has been completed. Several functions corresponding to FEE modules have been partly accomplished. Preliminary test from MDC FEE sub system to readout system to online system shows that readout system and online software work well together, the system can reach designed performance. Some other optimization work and integrated test have been scheduled and deployed.

References

[1] BES collaboration, BESIII Technical Design Report, Ver1, IHEP 2001.

[2] MVME5100 Single Board Computer Programmer's Reference Guide, V5100A/PG3, July 2003 Edition.

[3] ZHU Haitao, The Implementation and Application of state charts in Real-time system in High Energy Physics, IEEE Real Time Conference 2005.

[4] American National Standard for VME64 Extension for Physics and Other Applications, ANSI/VITA, 23-1998.

[5] Particle Detector and DAQ, First Edition, Xie YiGang etc. China Science Press, 2003.

[6] The application of PowerPC/VxWorks to the read-out subsystem of the BESIII DAQ, Tao Ning, etc. CHEP04.