IEEE Transactions on Nuclear Science, Vol. NS-28, No. 3, June 1981

Design of the Control System of TRISTAN

H.Ikeda, K.Ishi, T.Katoh, T.Kamei, Y.Kimura, S.Kurokawa, S.Shibata, M.Takasaki, S.Takeda and K.Uchino

KEK National Laboratory for High Energy Physics
Oho-machi, Tsukuba-gun, Ibaraki-ken, 305 JAPAN

Introduction

TRISTAN is an electron-positron-proton colliding beam facility of KEK (National Laboratory for High Energy Physics) [1]. The construction starts this spring and it takes five years before the commissioning of the electron-positron collider of 30 x 30 GeV. Then we will add a superconducting proton ring with a maximum energy of 300 GeV for electron-proton or positron-proton colliding. Its completion is expected to be in late 80s.

The electron-positron collider consists of three major parts: a 2.5 GeV linear accelerator, a 6 GeV accumulator ring and a 30 GeV main ring. In this report we present the design of the computer control system of the accumulator and the main ring of TRISTAN.

Basic Design Concepts

1. Distributed Control TRISTAN is a very complicated and large machine. It consists of a number of sub-accelerators and storage rings, which cover a large area. Large number of equipments are distributed around it. Complex operations are necessary in order to fully utilize its ability. These complexity and size of TRISTAN make it reasonable for us to adopt a scheme of the distributed computer control system. We can point out the following advantages of the distributed control system:

(1) The system is flexible, since we can modify the subsystems without affecting the rest of the system.

(2) Failures in one or more subsystems do not cause the entire system down.

(3) Cost of cabling is reduced due to short distances between computers and controlled devices.

(4) Number of transactions between subsystems are reduced, since preprocessed data are transmitted.

2. Languages A complex accelerator such as TRISTAN necessitates great software efforts, since elaborate operation of it and thorough investigation of its nature are possible only by means of good control programs. Therefore, it is advisable that the device designers and the operators, who know the algorithm better than the other people, write control programs. There lies a barrier in front of us, namely 'software barrier' [2]. The difficulties in making application programs are the main cause of preventing people from being familiar with computers. In the field of the distributed accelerator control this circumstances are more severe, since there are many problems inherent to the distribution of intelligences. For example, we must take into account the synchronization of two or more processes that run on different computers. This is a difficult task, if we write programs using a compiler language such as FORTRAN.

We have decided to adopt NODAL interpreter [3], which was devised and has been successfully used in CERN SPS, to overcome the above mentioned 'software barrier'. The reasons for adopting NODAL are:

(1) We can easily debug programs written in NODAL, since it is an interpreter.

(2) NODAL has the clear idea as to incorporate device handlers as data modules.

(3) NODAL has powerful string handling capabilities, which are very useful when we write programs that handle man-machine interfaces.

(4) In NODAL the lines are divided into groups; separate groups of a program can be executed as subprograms. This enables us to write structured and readable programs.

(5) NODAL is a multi-computer language; it allows a program to be expressed as a number of separate tasks which can be executed on different computers. This greatly simplifies programs which run on a multi-computer environment.

(6) NODAL has powerful real-time facilities.

3. Interface Standard We adopt serial CAMAC as the interface standard for the control of TRISTAN. The reasons are:

(1) It is the only commercially available data way which connects devices to computers over long distances with a sufficient transmission speed.

(2) Bypass and loop colapse functions of it are useful for the system maintenance.

(3) The test of prototype devices is possible before the commissioning of the control system, since CAMAC can be connected to a computer that is different from those used in the TRISTAN control.

Network

There are various techniques to link mini-computers to form a local network. For example, in SPS they use message handling computers based on the store-and-forward technique [2]. Though it is a very simple method, it has a drawback that the transmission speed is relatively low. Moreover, if the number of computers connected to the network is increased, we must use more than two message handling computers. This increases the complexity of the system and makes the response slow.

We adopt the method which has become popular in the industrial control such as large steel manufacturing plants. All nodes are connected with optical-fiber cables to form a ring network. The transmission speed on the optical-fiber cables is 10 Mbps. When a node wants to transmit a message to another node, it senses whether it can transmit the message or not. If the line is free, it transmits the message to the destination node. If the line is occupied by other nodes, it must wait untill the line becomes free. The function of message switching is distributed among the receiving nodes; each node recognizes and accepts the message addressed to it. There is no central node for switching. The transmission scheme is ,therefore, that of an N-to-N. This method ensures the very effective transmission. In our case overall transmission capacity is estimated to be approximately 300 kwords/sec; this speed is approximately 10 times faster than that of SPS. Another advantage of this method is that addition of a new node is easy and simple. We need only to cut the cable and insert a new node. This is very convenient for us, since the number of the nodes will be increased as the development of the accelerator.

The network for the TRISTAN control system is illustrated in Fig. Twenty-five similar 16-bit mini-computers are linked to two loops. Four local computers for the accumulator ring hardwares are connected to the small loop, whereas fourteen computers for main ring hardwares are connected to the large one. Seven central computers, which are located in the control center, are common to the both loops. There exists only one control center for TRISTAN; the facilities of central computers are commonly used for the accumulator and the main ring controls.

Software System

We adopt the ideas of SPS concerning the software system. These are:

1. Use of an interpreter language NODAL as a control language.
2. Use of a distributed data base.
3. Use of data modules for device handlers.
4. Assignment of linkmen as interfaces between the control group and hardware groups.

We intend to speed up NODAL interpreter by the following methods: (1) We make such NODAL interpreter that is similar to NODILER in SPS, that is to say, an incremental compiler [4]. (2) We use the microprogramming technique to fix frequently used portions of NODAL into firmwares.

Data modules are closely related to the hardwares driven by them. Therefore, it is convenient to develop data modules at the place where the hardwares exist. To this end, every local computer is equipped with a disk and a CRT terminal and can be used as a stand-alone program developer. Using it, we can make and test data modules by examining the action of the hardwares.

We develop NODAL compiler as a tool for making data modules. It must generate efficient codes to minimize the memory size and execution time, since they are important factors for data modules. By the use of a set of interpreter and compiler of the same language specification, the efficiency of making data modules will be greatly increased. We use the interpreter first to utilize the easiness of debugging programs. After debugging, we compile the programs to get object codes.

Process Interfaces

We can distinguish three levels in our system. The first level is the mini-computers and the links between them. In our system, main computing powers are concentrated in the mini-computers. The main transactions among mini-computers are exchanges of NODAL source codes. The average size of the messages is estimated to be around 200-300 bytes. The second level consists of CAMAC serial highways and CAMAC modules. From each local computer a CAMAC serial highway is extended over the devices. Bit serial CAMAC highway is adopted to minimize the cable cost and to simplify the connections. The transmission speed of it is 2 Mbps. The intelligence in the second level is CAMAC auxiliary crate controllers. The main purpose of them is to buffer data and adjust the difference of speed between computers and devices. The third one is the device level. If a closed feedback loop is necessary, it must be incorporated in the device. Transactions between mini-computers and devices are, therefore, simple commands and data.

To summarize, the first level communication can be regarded as the mean of communication between intelligences, whereas the nature of the second level is the extension of computer input-output buses. This point is essential for the choice of serial CAMAC as the process interface standard for TRISTAN. For example, there is criticism that CAMAC is not a suitable interface standard, because it is not devised to be a mean to communicate between intelligent devices. However, recent progress of micro-processors and LSIs enbles us to make intelligent devices. Control algorithms are fixed in the actual devices in the form of codes in PROMs; complex intelligences in CAMAC crates are not necessary. The nature of CAMAC as an extension of the computer bus is adequate for this kind of applications.

Control Center

TRISTAN should be controlled and monitored from a single control center. The functions necessary to the control center are divided into four subfunctions which are managed by functionally specified mini-computers. They are:

Operators Console Function,
Alarm and Logging Function,
Program Library and Data Base Function, and
Program Development Function.

Man-machine interfaces are very essential to the accelerator controls, since they are the means that facilitate the communication between operators and the accelerator. The basic philosophy of our man-machine interfaces is to have general purpose consoles. The improvements of the functions is achieved by increasing the number of the general purpose consoles and refining the control softwares. If we allow the use of a special console, the motivation of developing softwares to improve the controllability will be suppressed. At the beginning the special purpose console may be convenient, because it is made up for the special use. When we want the more elaborate control of the accelerator, this speciality limits the expansibility of the control program.

Program library and data base are managed by the file computer. It is similar to other computers except that it is equipped with large volume disks. All application programs, data modules are filed in the disks. At the start up of the system, the necessary files are sent from the file computer to the others and are stored on local disks; it helps to reduce the number of transactions on the network. Then data modules are loaded from the local disk into the memory. This reduces the overhead necessary for data movement from the disk to the memory.

There are a few cases which cannot be manipulated by the mini-computers on the network. Examples of them are execution of the large size programs for accelerator simulation and management of large data base of the accelerator history. For this purpose we use a back-end medium scale computer which is connected to the network through the program development computer. Also the central computers of our laboratory, Hitachi M200Hs, are linked to the network by means of a 48 kbps synchronous communication line. From any computer in the network we can communicate with them. Any terminal of the mini-computer can be regarded as a TSS terminal of the M200Hs.

Summary

We summarize here the outline of our system.

(1) Distributed computer control scheme is adopted.

(2) Twenty-five 16-bit mini-computers are connected by 10 Mbps optical fiber cables to form an N-to-N ring network.

(3) NODAL interpreter and compiler are used throughout the program development of the system.

(4) CAMAC serial highways are used as the means of communication between the mini-computers and the devices.

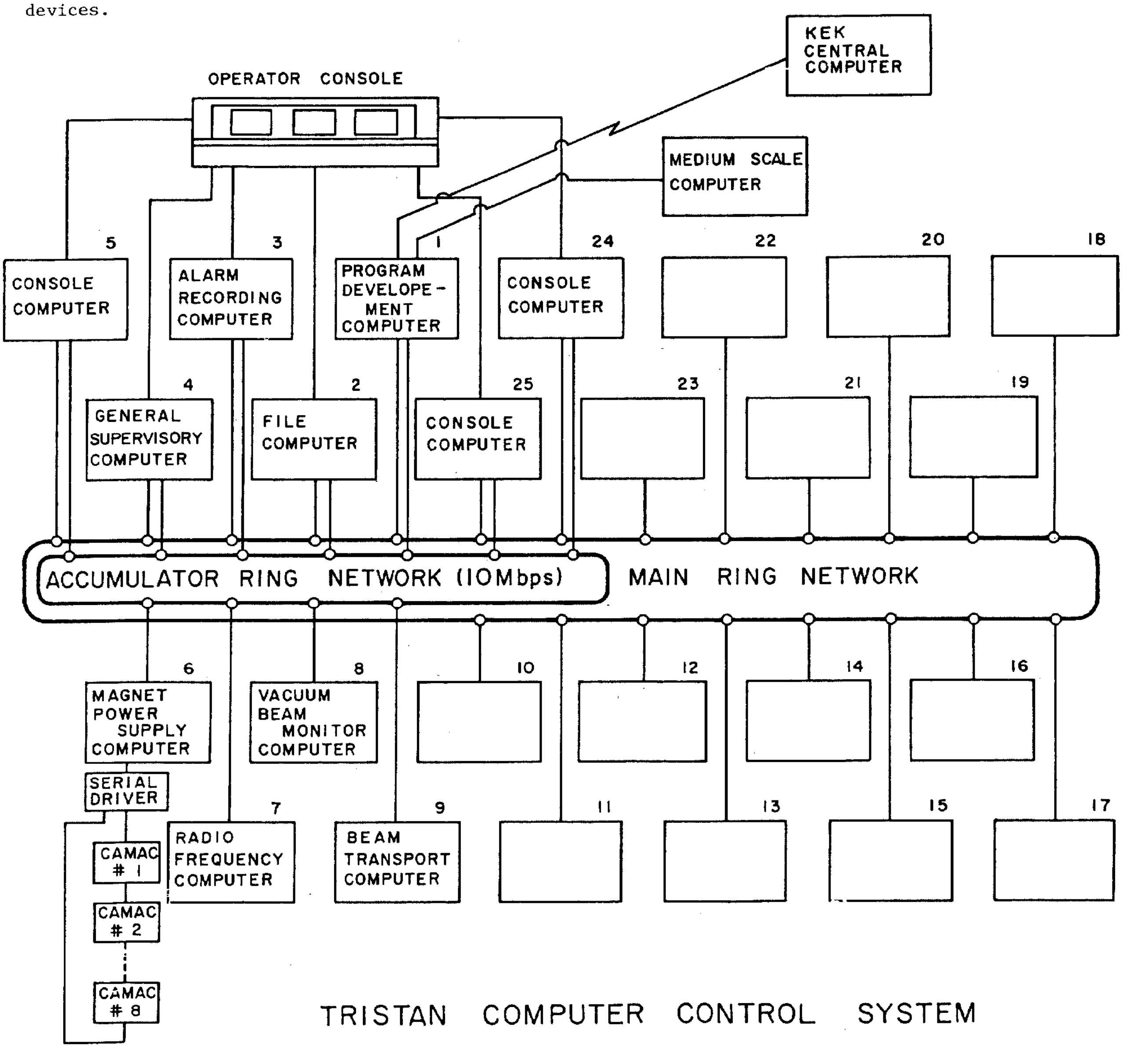
(5) TRISTAN system is controlled from a single control center.

Acknowledgements

We wish to thank Professors T.Nishikawa, S.Ozaki and H.Takahashi for their valuable discussions.

References

1. T.Kamei and Y.Kimura in this proceedings.
2. M.C.Crowley-Milling CERN 75-20 and CERN 78-09.
3. M.C.Crowley-Milling and G.C.Shering CERN 78-07.
4. J.Altaber and P.D.Van de Stok CERN SPS/ACC/PVdS/Rep. 79-4.



Figure