The 6th Symp. on Accelerator Science and Technology, 1987, Tokyo, Japan (© Ionics, Tokyo, 1987)

ONLINE USAGE OF GENERAL PURPOSE COMPUTERS IN ACCLERATOR OPERATION

E. Kikutani, H. Koiso and S. Kurokawa
National Laboratory for High Energy Physics,
Oho-machi, Tsukuba-gun, Ibaraki-ken, 305 Japan

ABSTRACT

The TRISTAN accelerator control system uses twenty-five minicomputers and large general purpose computers. The minicomputers control hardware equipment and serve for man-machine communication. The general purpose computers are used for computations which overload the minicomputers. In 1986, new network system, TRNET, was constructed to connect the minicomputers and the general purpose computers. Hardware as well as software system of TRNET is described.

INTRODUCTION

Recently the sizes of high energy accelerators have grown up to "km" order. The TRISTAN electron-positron collider facility[1] is one of typical examples of these machines. Large size means not only geometrically large but the large number of hardware components to be accessed. Then we can not avoid using a systematic computer control system to operate such large scale machines.

In the TRISTAN accelerator control system[2], minicomputers and large scale general purpose computers participate in operation of the accelerators. The minicomputers (the TRISTAN computers) are used for actually accessing the hardware components of the accelerators and for man-machine communication. The general purpose computers which are the resources of KEK Central Computer System (the central computers) are used to carry out the large scale computations which overload the minicomputers.

For the fast communication between the TRISTAN computers and the central computers, a high speed network is necessary. Until 1986, KEKNET[3] was used for this purpose[4]. But KEKNET has become old-styled and the support by the KEK Central Computer System has become poor. Then in 1986 we constructed a new network system, named "TRNET", for the exclusive use of operation of the TRISTAN accelerators. This paper describes the hardware and the software of TRNET.

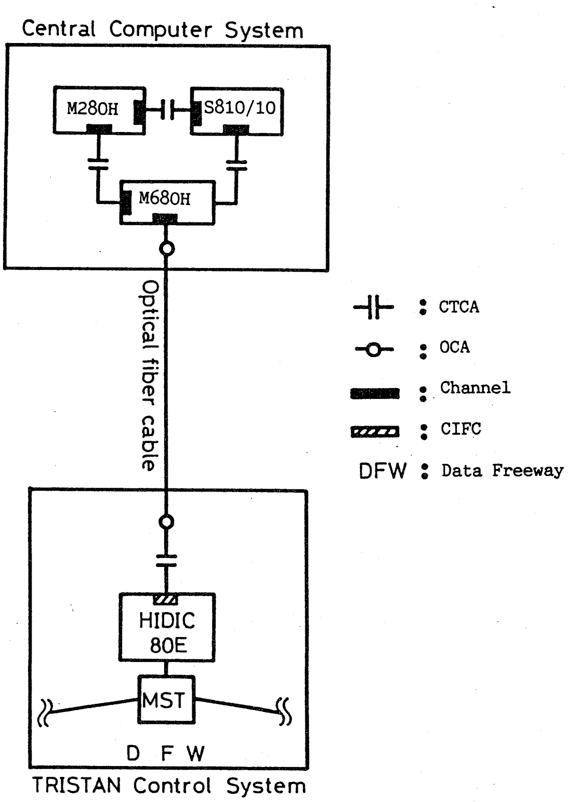

Fig. 1 The hardware structure of TRNET

HARDWARE SYSTEM

Computers The TRISTAN accelerator control system uses twenty-five 16-bit minicomputers with 1-Mips computing power, HITACHI HIDIC 80/E's and 80/M's. They make a token-ring network linked by 10-Mbps optical fiber cables. On the other hand, the KEK Central Computer System uses three large computers, HITACHI HITAC M-280H, M-680H and S-810/10; the former two computers are scalar processors while the last one is a vector processor for special usage.

One of the TRISTAN computers (named LBO) is connected to HITAC M-680H in the KEK Central Computer System. Fig. 1 illustrates the computer connection.

CTCA General purpose computers make I/O processes through a *channel* which is a kind of simple computers. Channels share memory with CPU and run on order from *channel programs*. In the KEK Central Computer System, communication among general purpose computers are carried out with *channel-to-channel adapters* (CTCA) as shown in Fig. 1. In the TRNET system, connection between M-680H and LBO is also done with CTCA which is installed in TRISTAN control building.

CIFC CIFC (Controller of InterFace Change) is a hardware which is installed in HIDIC series computers and it simulates the action of channel of general purpose computer. It was originally developed for connecting peripherals of general purpose computers to HIDIC 80's. In TRNET system CIFC is installed in LBO and connected to CTCA mentioned in previous section.

OCA and optical fiber cables In a classical way, the communication between the CPU and long-distanced peripherals are done with the aid of MODEM. But within half a decade, the channel cables can be extended to remote stations by optical fiber cables. In this method conversion between electronic and optical signals is done by Optical Channel Adapter (OCA).

In the TRNET system, a pair of OCA's are used; one is installed in the building of the Central Computer System and the other is in the TRISTAN Control Building. The distance between these buildings is about 1 km. The transmission speed is 20 Mbps on the optical cables which can be extended up to 3 km without repeaters.

SOFTWARE SYSTEM

Design principle The TRNET system is implemented in KEK NODAL system[2], which is the framework of TRISTAN accelerator control system. At the stage of designing TRNET, KEKNET had been used for communicating with the central computers and many user programs had been already developed. Thus the first guiding principle is that we must not change the syntax for end-users.

The second one is that the computer code must be written by KEK as can as possible. Under KEKNET, elementary parts of programs were written by the HITACHI software factory. The response of the software factory is not fast enough when some software troubles happened.

Software in the TRISTAN computers The software system of TRNET has two-layerd structure; the NODAL interpreter task and the CIFC handler task. In the NODAL system, a NODAL program is data to be processed by the interpreter which runs as a task under a multi-tasking operating system, PMS. There are four kinds of NODAL interpreter tasks; terminal tasks, remote execution tasks and real time tasks.

The communication with the central computer is done by calling *network functions* in NODAL programs. There are twenty network functions which are written in PCL (a FORTRAN-like compiler language for HIDIC 80). They are used to transfer data from/to a disk dataset on the central computer, submit a batch job, check status of a submitted job.

The flow-chart of network functions is shown in the left most part of Fig. 2. In a network function, a pair of subroutines KWREQ and KRREQ are used. These subroutines communicate with the CIFC handler task for sending and receiving a message.

The network functions are used only by the remote execution interpreter task, EXEC-P task, that has the highest execution priority among the NODAL interpreter tasks. In this way, it can be avoided to disturb the sequence of the KWREQ and KRREQ in a network function by the execution of a network function by other tasks.

Software in the central computer In the central computer, one job, TRNT, is always running on M-680H. This jobs is monitoring the attention from LBO. On the request by a operator, TRNT can submit another job to M-280H. The submitted job carries out calculations necessary to accelerator operation. Both TRNT and the submitted jobs run with a special priority higher than ordinary user jobs.

TRNT is a multi-task job and it consists of three tasks; the main task AOMP (Accelerator Operation Monitor Program), the channel communication subtask NIOP (Network I/O Program), and the disk I/O subtask DIOP (Disk I/O Program). Fig. 3 outlines the time flow of these tasks.

AOMP attaches subtasks at the start of TRNT and supervises the flow of these subtasks. The main program of AOMP is coded in FORTRAN and it calls subroutines in assembler to communicate with the subtasks. The synchronization among these tasks is accomplished by using WAIT and POST macros. Additionally this task has functions of submitting job, checking status of a submitted job and other auxiliary functions.

The subtask, NIOP, takes charge of handling channel I/O's. It is coded in assembler and includes channel programs which consist of 8-byte CCW's (Channel Command Word). By calling EXCP macro, NIOP executes actual I/O for CTCA directed by the channel program.

The subtask, DIOP, works as a disk handler and it has four functions; open datasets, close datasets, read data from datasets and write data into datasets. This task is also coded in assembler. If some error occurs, DIOP reports it to AOMP through the POST code.

PROTOCOL OF TRNET

Format of a message frame A message frame which is exchanged on TRNET consists of three parts; 16-byte system header, 8-byte user header and data. The maximum length of data is 8168 bytes, which is restricted by the size of I/O buffer size on LBO. The user header contains the following parameters; 1) the length of data, 2) the request code which distinguishes the NODAL network functions, 3) the character code which identifies the code of data in the frame such as binary, ASCII, EBCDIC, etc, 4) the sequence number and 5) the return code.

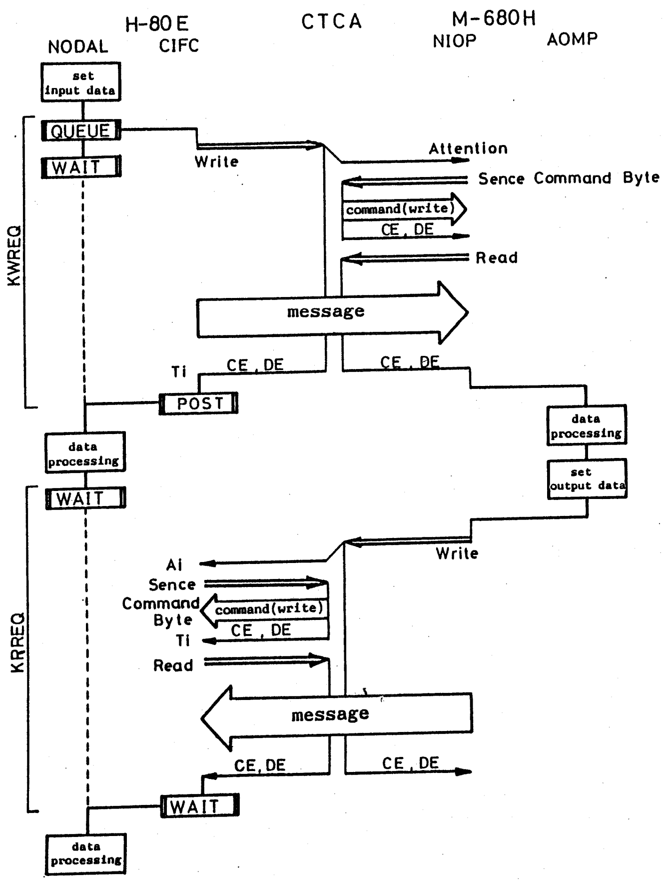

Fig. 2 Data exchange procedure in TRNET

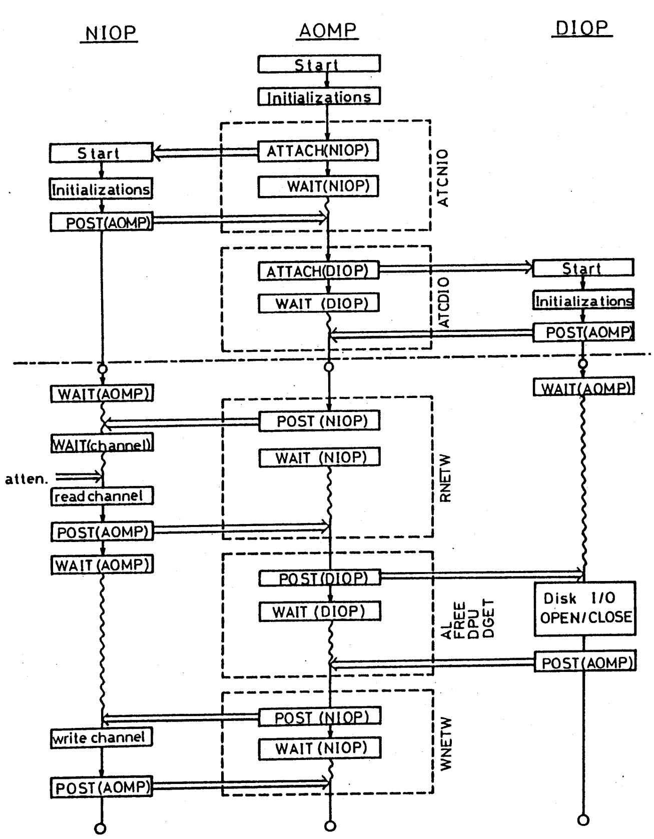

Fig. 3 Time chart of job TRNT

Message Exchange Procedure A series of data exchange is initiated by LBO. Fig. 2 shows a time chart of the message exchange procedure. At first the CIFC handler task issues the *write* command to CTCA, then CTCA gives attention to NIOP to inform that some command is sent from CIFC. NIOP checks as if the received command is *write* and it issues *read* command to receive a message. After data processing by AOMP, NIOP send a reply message and the CIFC handler receives it by using also *write* and *read* commands.

PERFORMANCE OF TRNET

Transmission speed The transmission speed on TRNET was measured with the most simple network function which contains only the sending and receiving action. In this measurement, a simple main program which receive a message and immediately sends it back was running on the central computer. Fig. 4 shows the time for sending/receiving data as a function of data length. The highest transfer speed of 105 kbytes/sec is obtained at the data length of 8168 bytes and the asymptotic transmission speed determined by the gradient of the curve is 121 kbytes/sec.

Operation example A typical example of operation with TRNET is the correction of closed orbit. Generally, the closed orbit in an accelerator is different among the optics. Therefore if some new optics is examined, orbit correction must be carried out.

In the TRISTAN Main Ring, beam orbit is measured with 392 position monitors and the correction is done with 520 steering magnets. The calculation to find suitable kick angle of each steering magnet is carried out on the central computer by the program PETROK which is a KEK version of PETROS[6]. The size of input data amounts to 9.4 kbytes, the necessary CPU time is ~40 sec and the typical elapsed time is a few minutes.

ACKNOWLEDGMENTS

We wish to thank Professors H. Baba, G. Horikoshi, Y. Kimura, and S. Ozaki for their support during this work. We also thank the other members of the TRISTAN control group and the accelerator performance developing group. Thanks are also given to system engineers of HITACHI Co. ltd.

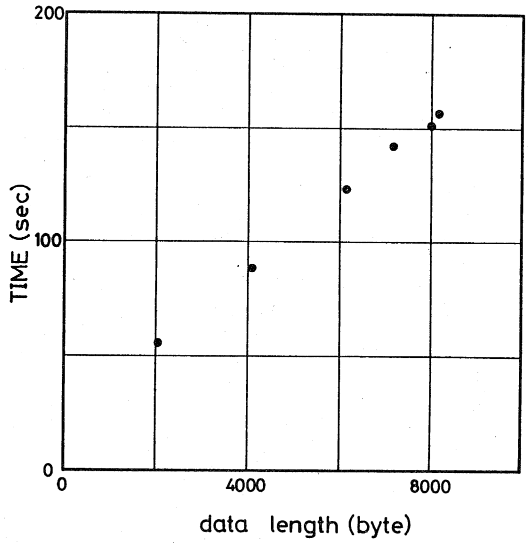

Fig. 4 Time for data transmission on TRNET

REFERENCES

1) G. Horikoshi and Y. Kimura, "Status of TRISTAN", Invited Talk at the 1987 Particle Accelerator Conference, Washington, KEK preprint 87-23.

2) S. Kurokawa et al., "THE TRISTAN CONTROL SYSTEM", Nucl. Instrum. Methods A247 (1986) 29.

3) Y. Asano et al., "A High Speed Online Network for High Energy Physics", Nucl. Instrum. Methods 159 (1979) 7.

4) S. Kurokawa et al., "CHARACTERISTICS OF TRISTAN COMPUTER CONTROL NETWORK", Nucl. Instrum. Methods A247 (1986) 202.

5) M.C. Crowley-Milling and G.C. Shering, The NODAL system for the SPS, CERN 78/07

6) K. Steffen and J. Kewisch, DESY 76/07