COMPUTER CONTROL SYSTEM OF TRISTAN

H. Koiso, A. Akiyama, T. Katoh, E. Kikutani, S. Kurokawa and K. Oide

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

Abstract

The Accumulation Ring (AR) and the Main Ring of TRISTAN is controlled by a distributed computer control system, in which twenty-four minicomputers are linked to a token-passing ring network by optical fiber cables with 10 Mbps transmission speed. The software system is based on the NODAL interpreter developed at CERN SPS. The KEK NODAL has some improvements such as the fast execution speed due to the compiler-interpreter method etc. Eleven minicomputers have been connected to the network and are successfully working on the operation of AR. The performance of the present system has been studied.

Introduction

An electron-positron colliding beam facility, TRISTAN, is now under construction at KEK[1]. It consists of three accelerators: a 2.5 GeV linac, an 8 GeV accumulation ring (AR) and a 30 GeV main ring (MR). AR and MR are controlled by a single highly-computerized control system.

On designing the TRISTAN control system we have considered the following requirements:

(1)--It can control hardware equipment distributed over the circumference of 3 km.
(2)--It must be flexible to be adapted to various changes in machine design and operation style.
(3)--To assure the flexibility, the software system must make programming easy for machine operators who are not expert programmers.
(4)--To do more complicated calculations such as corrections for closed orbit distortions etc., it must be able to utilize a general-purpose computer with high-speed calculation capability and large memory.

System architecture

Network and minicomputers The comlexity and the size of TRISTAN compel us to adopt a distributed computer control system. Twenty-four 16-bit minicomputers (Hitachi HIDIC 80-E's and HIDIC 80-M's) are distributed around the accelerators. These computers are linked together by optical fiber cables to form an N-to-N token-passing ring network[2]. Each minicomputer serves as a node for this ring network.

The message transfer between the nodes is done as follows: (1) While any node has no transfer request, a free token is circulating along the network. (2) If a node wants to transmit a message, it must catch the free token. Then it sends a busy token followed by a message packet with a header, in which the address of a destination node is written. (3) Each node reads the header and passes the packet to the next node. Only the destination node copies the message. (4) The busy token travels again to the source node, where the message packet is destroyed and the busy token is replaced with a free token and sent again. The transmission speed on the optical fiber cable is 10 Mbps. Fig.1 shows the schematic layout of the TRISTAN control system.

The minicomputers are classified into two groups: the system computers and the device-control computers. The nine system computers are located in a central control room. Each of them processes one of the control center functions such as the operator's console (OP0-OP4), the alarm alerting (AL0-AL1), the program library (LB0), and the program development (DV0). Each console computer has one or two color graphic displays and two color touch panels. The fifteen device-control computers

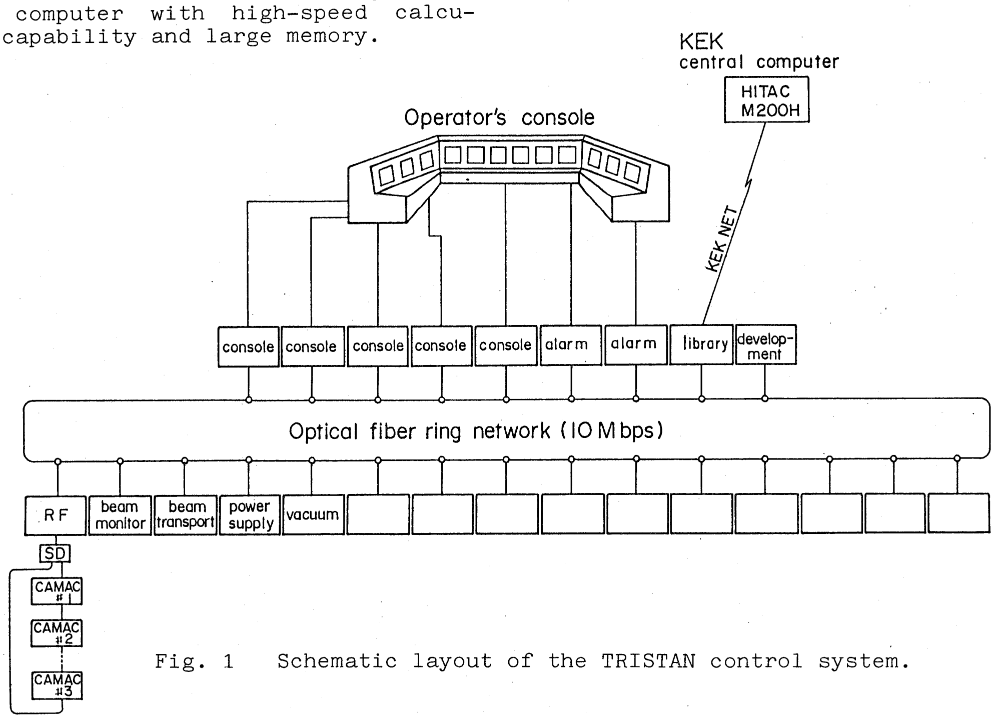

Fig. 1 Schematic layout of the TRISTAN control system.

control hardware equipment such as magnets, power supplies, radio-frequency systems, etc. These computers are installed in the site buildings around the accelerators.

<u>CAMAC serial highway</u> From each computer, a 2.5 Mbps bit-serial CAMAC serial highway extends to hardware equipment. The number of CAMAC crates is 40 for AR and 140 for MR. The advantages of CAMAC and CAMAC serial highway are: (1) CAMAC serial highway is one of the most powerful standards for the long-distance process control data-highway. Its bypass and loop-collapse functions are useful tools for maintenance and diagnosis, (2) many kinds of CAMAC modules are commercially available from many vendors, and (3) we can easily make a test bench with a low-cost personal computer, because CAMAC is a computer independent standard.

<u>Linkage with KEK central computer</u> To manage tasks which overload minicomputers, the library computer LBO is connected to the KEK central computer (Hitachi M-200H) by KEKNET, an in-house high-speed network at KEK[3].

NODAL system

<u>Multi-computer programming facility of NODAL</u> The software system of the TRISTAN control is based on NODAL originally developed at CERN SPS[4]. This language is an interpreter and enables us to develop programs interactively. The multi-computer programming facility of NODAL, which is the most important feature, effectively unifies a distributed computer system. An example below illustrates how the multi-computer facility works:

```
1.1 DIM A(10)
1.2 EXEC<MGO> 2 A ; WAIT<MGO>
1.3 FOR I=1,10 ; TYPE A(I)!
1.4 END

2.1 FOR I=1,10 ; SET A(I)=BACURR(I,'CUR')
2.2 REMIT A
```

When a computer, for example OPO, interprets this program from the line 1.1 and encounters the line 1.2 including the EXEC command, OPO sends the program of group 2 (that is the lines 2.1 and 2.2) and an array A to the computer MGO. The NODAL interpreter on MGO then interprets these lines and sends back the array A to OPO by the REMIT command. Then OPO types out the value of A on its terminal. As this example shows, the multi-programming facility enables us to unify subtasks to be executed on different computers.

<u>Data module and distributed data base</u> Another important feature of the NODAL system is the existence of data modules, which are device handlers. In the above program, BACURR(I,'CUR') is an example of a data module, where BACURR is the data module name, I is the unit number, and 'CUR' is the property. The property distinguishes operations on the hardware equipment. In this example, the data module reads the current of the I-th magnet and sets it to the I-th element of the array A.

Each data module has a 2-dimensional control table which stores various parameters such as addresses of CAMAC modules. A data module and a data table are loaded only on the computers equipped with the hardware on which the data module operates. In this environment, the multi-computer facility of NODAL effectively works as a distributed data base handler.

Improvements of KEK NODAL

The KEK version of NODAL has the following improvements over the SPS NODAL: (1) the fast execution speed due to the compiler-interpreter method, (2) the extended multi-computer file system, (3) the dynamic linkage of data modules and functions, and (4) the screen editing facility.

<u>Fast execution speed</u> To overcome slow execution speed due to the interpreting scheme, we adopt the compiler-interpreter method in KEK NODAL. When a programmer write one line of a source code and press the return key, the compiler part of NODAL translates the source code to an intermediate code. The source code is stored on a magnetic disk, and the intermediate code on memory. The interpreter part of NODAL interprets the intermediate code at run time. This method can save time for changing a constant in ASCII string form into a floating point form, time for rearrangement of the order of operators and operands, etc. at the interpreting phase. As a result, two to three times speedup is achieved compared with the direct interpreter scheme.

<u>Extended multi-computer file system</u> KEK NODAL has a extended file system, under which we can access any NODAL program files, data files and I/O devices, such as a line printer or a magnetic tape etc., attached to different computers. When we want to print out the program file TEST on the computer OPO to the DVO's line printer, we can do it as follows:

```
OLD OPO/TEST
OPEN(11,'W','DVO/LP:')
LIST<11>
CLOSE(11)
```

<u>Dynamic linkage</u> The KEK NODAL system has various data modules and functions written in PCL, a FORTRAN-like compiler language for process control programming on HIDIC 80's. To make the system flexible, these subroutines must be easily linked. In KEK NODAL, the information for each subroutine such as the address, the number of arguments, the argument type, etc. is stored in a table. At run time, the NODAL interpreter links subroutines dynamically according to this table. Therefore, re-linking a data module or a function to the NODAL interpreter is not necessary when a new one is added.

Evaluation of system performance

<u>Present status</u> Eleven minicomputers (HIDIC 80-E's with 192 kword memory and 8.6 Mword magnetic disks) are now connected to the network. This computer system has been successfully working on the AR operation since October 1983. About 500 application programs are stored in each of three console computers and two-fifths of them are actually used in day-to-day operation and machine study. These programs have been coded only for one year, which shows the productivity of the KEK NODAL system. The completion of the control system is scheduled in 1986.

<u>Performance of the network</u> The message transfer on the network is managed by a dual system of Control Stations (CST's). The CST monitors the message flow and traces its status on 2KW RAM. We can easily read the RAM

with NODAL user functions. This on-line tracing facility is a powerful tool for diagnosis of the network.

The performance of a network is evaluated by the throughput and the response. The throughput is determined by three parameters: the number of transmission words (M), the loop length (L) and the number of nodes including CST's (N). Fig. 2 shows the throughput versus M in the present network where L is 1.5 km and N is 13. The maximum message length to be sent as a packet is 256 words and a longer message than this limit is divided into packets.

The channel utilization of the network is represented by a ratio,

$$m \times Tt \; / \; (m \times Tt + n \times Tn)$$

where Tt is the averaged time for one transmission cycle, m is the number of transmitted packets, Tn is the averaged time for one turn of a free token and n is the number of turns of a free token. Fig. 3 shows this ratio on AR ordinary operation, where the maximum utilization is 2.0 %. At that time, the averaged word length of a packet was 48 words and the transmission rate was 5 kword/s. The utilization of the complete system is estimated to be 10-13 %, assuming that L is about 10 km, N is 26 and the transmission rate is 25 kword/s.

The EXEC responses of two typical types were measured:(1) transmit only the END line and (2) transmit an array of 1 kwords and remit the array immediately as:

```
1.1 DIM-I A(1024)
1.2 EXEC<LBO> 2 A ; WAIT<LBO>
2.1 REMIT A
```

We obtained 92 ms for (1) and 175 ms for (2) as a result. The response was linearly dependent on the size of the array and became slower by 20 ms when the array was enlarged by 256 words. Since most of EXEC transactions transmit data (a program and variables) of less than 1 kwords (three-fourths of them transmit data of less than 256 words), the EXEC response of the present system is typically 100-150 ms, which is shorter than the standard hardware response of the accelerators.

An EXEC or a REMIT transaction causes transmissions of at least six message packets as shown in Fig. 4. There is enough unoccupied time between packets. This implies that the network maintains the good response even if EXEC transactions between more than two pairs of computers occur at the same time. Actually we measured also 175 ms response, when three simultaneous EXEC transactions of the type (2) between three pairs of computers were processed. In this case the utilization of the network was 12.5%.

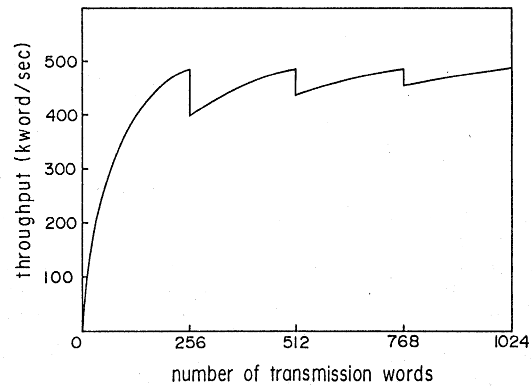

Fig. 2 Throughput of the present network.

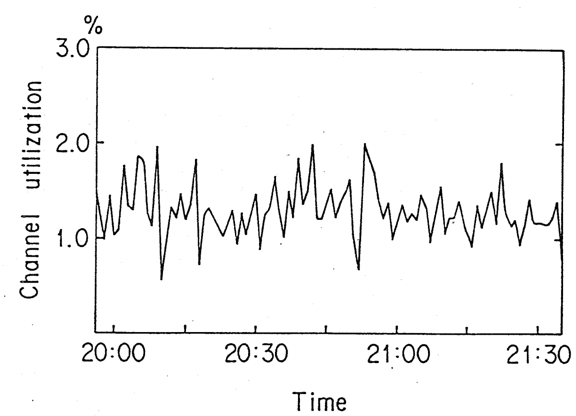

Fig. 3 Example of the channel utilization on AR operation.

Acknowledgements

We wish to thank Professors Y. Kimura and G. Horikoshi for their encouragement and support during this work. We also thank the members of the TRISTAN control group and the TRISTAN operation group for their useful discussion and advice.

References

1) T. Nishikawa and G. Horikoshi: Status of KEK TRISTAN Project, IEEE Trans. Nucl. Sci. 30 (1983) 1983.
2) Hitachi,Ltd.: H-7485C Optical Data Free Way Hardware Manual, HIDIC 80 series 80-4-090.
3) E. Kikutani et al.: Accelerator Operation with General-purpose Computer, contributed paper to this conference.
4) M.C. Crowley-Milling and G.C. Shering: The NODAL System for the SPS, CERN 78-07 (1978).

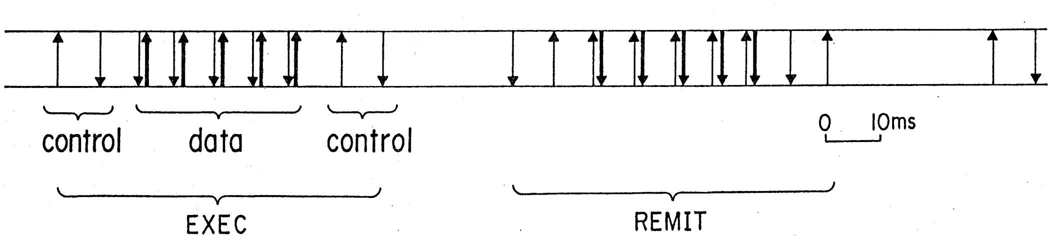

Fig. 4 Exchange of packets on the EXEC and REMIT transactions. An arrow means the transmission of a packet.