DECENTRALIZATION OF THE NAC CONTROL SYSTEM

G.F Burdzik, R.K. Fisch, W.L.J. Hendrikse, I.H. Kohler, H.W. Mostert, J.N.J Truter, K. Visser and H.F. Weehuizen
National Accelerator Centre, CSIR
P.O. Box 72, Faure, 7131, Republic of South Africa

Summary. The original NAC control system was conceptualized around a classical hardware topology: a number of mini-computers linked to the accelerator hardware by a CAMAC Executive Crate, a serial highway and several parallel branches. Since 1983 we have been progressively introducing low-cost microcomputers for the local control of accelerator sub-systems such as the rf systems, most of the stepping motor applications, and a number of beam-current measurement systems. The introduction of further decentralization was recently given an impetus as a result of a requirement to interlock the currents of a large number of magnet power supplies. This requirement has prompted the following developments: i) the decision to abandon CAMAC as the interfacing standard in favour of a microprocessor bus standard, and ii) the adoption of IBM PC compatibility as a standard for our microprocessors.

Introduction

Tha National Accelerator Centre's control system was designed in the late 1970's. It is based on a classical hardware topology:- a number of mini-computers linking to a CAMAC Executive or System Crate, to which is also connected a CAMAC serial highway and several CAMAC parallel branches. In our present configuration there are two mini-computers connected to the Executive Crate, which are used for the control of the accelerator. A third mini-computer, which is used for program development, is linked to this system only in that it shares a common hard-disk system with the two control computers. The serial highway has been interfaced to 'slow' equipment which includes all the magnet power supplies. On the other hand, although not all equipment interfaced via the parallel branches is 'fast', a fair proportion of it is characterized by requiring 'fast' access at times by the mini-computers (eg. during the acquisition of beam currents from a moving probe, or from a harp or rotating beam profile monitor), or is equipment which can set LAM interrupts. The control stations, of which there are four, are linked via the parallel branches: two of the control stations are part of the main control console, and the other two are mobile consoles.

An important feature of the above-mentioned topology is that every mini-computer connected to the Executive Crate has access to the entire CAMAC system (i.e. the system is fully multi-sourced). As Gurd et al[1] have pointed out, this allows one to allocate the workload on the mini-computers by tasks rather than by accelerator sub-systems. One of the control mini-computers is used for the servicing of the control consoles, i.e. for the servicing of interrupts (LAMs) originating at the consoles, and for the updating of status and actual-value information at the consoles, and for the programmatic setting up of many of the accelerator sub-systems (e.g. for programs which automatically condition the magnets of the cyclotrons to give reproducible magnetic fields). The other is used for the fast acquisition and display of data from harps, scanners and probes at the main control console.

The Control System Mini-computer Software

The existing control system software on the mini-computers mirrors the topology of the hardware, in that it is very centralized. Firstly, all the mini-computer programs make use of centralized data tables in mini-computer memory, which contain a description (identifying code, CAMAC addresses, calibration constants, etc.) of all the external devices to be controlled and monitored. Secondly, apart from equipment which is already being controlled by local microcomputers, the control and monitoring of accelerator devices are accomplished directly by mini-computer programs, e.g. all of the approximately 180 magnet power supplies are under direct control of the mini-computers. Thirdly, data acquisition for monitoring purposes is accomplished by a sequential accessing of individual devices by the mini-computers rather than by a block transfer of data, e.g. the program UPDAT which periodically updates the actual values of variables displayed on the control console colour monitors, sequentially schedules, for each of the variables displayed on the console, an equipment-specific program to read the actual value at the equipment and waits until the program returns the actual value.

Development of Microprocessor Capabilities at NAC

It was our intention from the outset to decrease the workload on the mini-computers by offloading tasks of a dedicated or single-task nature onto microprocessors. Our first efforts in this direction were the inclusion of embedded processors in some of the control console instruments (a unit with which an operator can set a reference value or setpoint, and an analogue meter which accepts digital data via CAMAC). This trend has continued, and most of our recently-developed I/O cards include embedded processors.

The decision to re-distribute tasks also included the decision to employ microprocessors as auxiliary crate controllers at the CAMAC crate level. However, our choice of a microcomputer bus on which to standardize was a controversial one. At the time, a number of interested laboratories and private companies in South Africa proposed the development of a low-cost South African bus standard (the SABUS standard), and we decided to support its development[2], rather than use microprocessors based on internationally accepted standards such as S-100, Multibus, etc. An obvious drawback of that decision was that we have had to do much of the development of the microprocessor and interface cards ourselves, the initial development effort turning out to be rather expensive. On the positive side is the fact that we now have staff members who know this system intimately, and who have the ability to design and develop or modify fairly rapidly any card which is required. The real benefit at this stage, however, is that the cost of fabricating the electronics in-house is much lower in our case than buying imported hardware, even if one takes development costs into account.

Until recently the development of the SABUS microcomputer at NAC was restricted to 8-bit systems, because i) the 8-bit systems are inexpensive and have adequately performed the tasks required of them as auxiliary crate controllers, and ii) the upgrade path to 16-bit systems was unclear: some members of the division felt that in upgrading we should abandon the SABUS standard in favour of one like Multibus.

A typical SABUS 8-bit system, used as an auxiliary crate controller, comprises a Z80-based CPU card with 64k bytes of memory and an RS-232 serial port to which is attached a video terminal, a 128k byte bubble memory card, and a CAMAC interface card. The last-mentioned card is a suitable interface for either of the following two CAMAC modules: (i) a commercially-available auxiliary crate controller CAMAC module conforming to the ANSI/IEEE 675-1979 standard, or (ii) a mailbox CAMAC module developed at NAC. The operating system CP/M 2.2 has been configured to run on this hardware, allowing booting from a floppy disk, or if a floppy disk is not present, the bubble memory card. Application programs are written in a popular implementation of Pascal. During program development a floppy disk controller and 8-inch floppy disk drive are added to the system. Such a system is very convenient for the development of local-control applications: Its convenience is due to three factors. Firstly, the development and the target systems are one and the same, so that development, debugging, etc. can all be done on the final hardware and <u>in situ</u>. Secondly, the particular implementation of Pascal is suited to rapid prototyping and application development. Thirdly, once a program has been developed and debugged, it is easily loaded into the bubble memory from where it can be booted by means of an auto-booting procedure. Any modification of a program in a locally-controlling SABUS system is similarly easy.

An interesting spin-off of the SABUS development has been the use of the SABUS microcomputer in stand-alone applications which do not have anything directly to do with the control system. One such application is the development of control systems for accelerator vacuum sub-systems which forms the subject of a separate contribution to this conference[3]. Further applications are the use of SABUS microcomputers in the NAC Safety Interlock System, in an interlock system which prevents the movable devices in the central region of the solid-pole cyclotron from colliding with one another, and in a control system for a cooling system required for isotope production.

Mircocomputer Applications in the NAC Control System

SABUS microcomputers have been used for distributed control in three major application areas, viz. for the control of stepping motors, rf systems, and in beam-current measurement systems. In all cases the use of the microprocessors has been similar. In the first two applications the local microprocessor is given access to a given group of CAMAC crates by means of an auxiliary crate controller module in every one of the group of crates. In all applications communication between the control mini-computers and the local microcomputer is accomplished by the inclusion of a CAMAC mailbox memory module in a crate to which the mini-computers and the local microcomputer have access. Control of the accelerator hardware, and checking of the status of, and consistency of the data read from, the hardware, are left entirely to the local micro. Status and actual-value data are written at regular intervals by the micro to the mailbox, to be read by the mini-computer at its convenience. When a mini-computer wishes to control some equipment under the local control of a microcomputer, it sends a high-level command to the mailbox together with any required data. The local micro then interprets the command and executes it. The local micro is also required at regular intervals to retrigger a watchdog in the mailbox, so that if any micro should fail, such a failure would manifest itself by the setting of a LAM interrupt in the mini-computer.

Stepping Motor Control

Initially we had relied too heavily on the assumption that stepping motor hardware, which in principle should have worked reliably, would work without error, e.g. that absolute-encoder data were error-free, etc. When it became apparent that closer error monitoring was required, neither the input/output capabilities of the mini-computers nor the indeterministic nature of tasks running under a real-time system lent themselves to using the mini-computers for this purpose. The primary reason for putting our stepping motor applications under local microprocessor control, therefore, was to increase the exception-handling capabilities of the control system. Secondly, it has been possible to hide the details of several different hardware implementations of stepping motor control from the mini-computers. Thirdly, it has also become easier to provide beam-related rather than motor-related variables at the control consoles: e.g. instead of requiring an operator to control the four individual motors of the ion source of the solid-pole cyclotron, in order to set the ion source - puller gap, ion source tilt, etc., the operator deals directly at mini-computer level with four beam-related variables, viz. ion source - puller gap, ion source displacement parallel to puller face, vertical displacement and tilt. The locally-controlling microcomputer then transforms these variables into motor-related variables.

Rf System Control

The rf division was originally given a SABUS system, so that control strategies for the rf system could be developed without having to rely on mini-computer availability. The decision was a good one, and the programs developed on the SABUS system include a stand-alone program with which the rf systems can fully be controlled from the keyboard of the SABUS terminal, as well as programs for diagnosing problems. When the rf system is controlled from the mini-computer the local SABUS microprocessor takes its instructions from a mailbox, as mentioned earlier, and on a regular basis writes back status and actual-value data to the mailbox. As is the case for stepping motor control, provision has been made for beam-related rf variables at the control consoles, the rf SABUS system being responsible for translating them to rf-equipment related variables. In addition a command stack has been implemented in the mailbox, enabling an operator to switch the rf system off-line to the mini-computers, and later switching it to mini-computer control again, without loss of commands from the mini-computer system.

Beam-Current Measurement Systems

Several auto-ranging current measurement systems, which make use of SABUS microprocessors and communications modules based on the RS-485 specifications and a variation of the SDLC communications protocol, have been manufactured. Proprietary remote controller (slave) and communications controller (master) boards were bought, and an interface to SABUS implemented, enabling the SABUS microcomputer to communicate with the master modules and through the communications link with the slaves.

The off-loading of tasks onto microcomputers has had several very beneficial effects on the control system. Firstly, the control programs in the mini-computers have been reduced in complexity and number, and the mini-computer tasks are less critically dependent on the mini-computer I/O structure and on having to be real-time. Instead of having to cater for several different hardware implementations of stepping motor control on the minis, the mini-computer programs now communicate in a uniform way with the mailboxes associated with stepping motor control. Secondly, the microcomputers are at all times available for maintenance, diagnosis and development of the sub-systems they control, reducing considerably the requirement of mini availability for these purposes. Thirdly, the provision of beam-related rather than equipment-related variables has been a natural result of their use. However, as Goodwin and Shea have pointed out[4], the overall architecture of a centralized control system is not changed significantly by the addition of the locally-controlling micros, and the NAC control system still has some rather annoying shortcomings.

Problems with the Present Control System

Shortcomings in the Hardware

The most troublesome problem from the operational point of view is probably the lack in certain equipment (mostly power supplies) of setpoint or reference value registers which can be read by the control system. The lack of readable setpoint registers prevents the recovery by the control system of setpoint values (i) when equipment is set offline to the control system and is then switched to computer control, or (ii) when a control mini- or microcomputer fails and is restarted during a run. We have considered a number of solutions, including the writing at regular intervals of all the reference values to disk, or attempting to recover the setpoint values from the actual values, but none of these methods is sufficiently general to cover all eventualities. The only satisfactory solution is to provide hardware 'setpoint' registers which are used and are readable by both the local and control system control as the sole external 'locations' of reference values.

A further shortcoming is the centralized nature of our CAMAC interfacing link, and the fact that the mini-computers are always involved in any transfer of data from one CAMAC crate to another when the crates are separated by more than a few meters. The problem is compounded by the mini-computers' operating system and I/O structure which are not particularly suited to the handling of high interrupt and data rates, e.g. it takes about three milliseconds for the operating system to schedule a LAM-servicing routine, after a LAM source has been detected and identified.

A third problem is the tightly-coupled nature of the CAMAC parallel branches: the control system is not 'gracefully degradable' when a CAMAC module needs to be exchanged or when it is desired to expand or change the control system configuration. Instead, the whole control system has to be stopped and the branch(es) in question taken off-line.

Fourthly, from an operational point of view there is a requirement for more local control stations, particularly for maintenance and the diagnosis of accelerator hardware problems, e.g. it would be most useful if a local control station could be made available for maintenance purposes in the vaults of the cyclotrons. We do have mobile control consoles available, but they are not as mobile as one would like, as they link to the control system via CAMAC parallel branches.

Fifthly, the memory and speed limitations of our 8-bit microprocessor systems have become apparent, and a need has arisen to upgrade the SABUS to run modern 16-bit microprocessors.

Shortcomings in the Control System Software

Apart from some small but annoying shortcomings in the structure of the mini-computer software, which will be rectified shortly, the problems with the control software arise mainly as a result of the centralized nature of the control and monitoring processes in the mini-computers. Firstly, the centralized device tables oblige us to store device-specific details which should more logically be stored near the devices themselves. Secondly, most of the mini-computer equipment-specific programs have to cater for a variety of hardware implementations of basically similar equipment: the magnet power supplies are a case in point. As a result, programs either become difficult to maintain if one attempts to accommodate all the hardware differences in one set of programs, or the programs multiply in number and cannot all remain memory resident. It would be better to provide the mini-computers with a uniform interface to all power supplies, hiding the equipment-specific details at the equipment. Thirdly, the sequential accessing of equipment by the mini-computers for updating of information on the consoles is inefficient and very time-consuming. We find that when pages displaying variables are selected on more than two consoles, the console responses to operator requests become sluggish and unpredictable. Basically the acquisition of data at the equipment level should be decoupled from the data transfer process to the mini-computer and the display of those data at the consoles, so that each of those processes can proceed unhindered by one another.

An Interlock Requirement and its Impact on the Control System Hardware

Recently the group at NAC responsible for beam development specified that many of the accelerator magnet power supplies be interlocked with respect to actual currents flowing through the magnet coils. The interlock hardware must monitor the actual current for each magnet and provide a hard-wired status signal to the Central Safety System indicating whether or not the current is within a given 'window'. The existing CAMAC modules which interface the power supplies to the control system do not satisfy the interlock specifications (A/D conversion times are far too slow, no status signals are available for the safety system, etc.), and therefore the interlock hardware requires duplicating some of the functions of existing CAMAC modules (e.g. A/D conversion). Moreover, the interlock hardware will also be required to interface to the control system, as the 'window' specifications for monitoring the actual currents are energy- and particle-dependent and must be downloaded to the interlock hardware from the control system.

The requirement of providing the interlocking hardware had a cathartic effect on our thinking. If the development of the interlocking electronics was unavoidable, and the specifications were better than for our present CAMAC modules, then such hardware should also be designed to be suitable for any future interfacing of power supplies to the control system. There was the further requirement to rectify shortcomings in the present system: in the case of the power supplies, to provide readable hardware setpoint registers at the power supplies, local control stations in the power supply area, to increase the tempo at which the actual values of power supplies could be monitored, and to simplify for maintenance purposes the mini-computer power supply programs. We wished also to take our longer-term plans into account, namely, the installation of a local area network at some future data, in order to resolve the bottle-neck resulting from the star-like CAMAC interfacing link.

CAMAC equipment has become very expensive in this country (imported CAMAC modules cost two to three times that of the equivalent locally-produced SABUS modules, even if SABUS development costs are taken into account). Moreover, the costs of developing CAMAC modules in-house are inherently greater than for SABUS modules. There is also no compelling advantage in staying with the CAMAC standard: on the contrary, there are drawbacks. The other microprocessor-based busses such as Multibus, VME and S-100 are similarly expensive. (If we had originally decided to standardize on one of the previously-mentioned busses and participated locally in its development, the cost of locally-made equipment would no doubt have been much cheaper). In view of the experience and success we have had with the SABUS system, we have therefore decided to develop all our future requirements based on it.

The development of SABUS replacement modules for CAMAC has thusfar been confined to a stepping motor controller, a power supply controller and a 16-bit input gate/output register. In each case the module is functionally equivalent to and connector compatible with the CAMAC module it replaces. A window monitor which will be used for the interlocking of power supply currents is also under development. The stepping motor controller is ready to go into production, and the other modules are in the proto-typing stage. We intend also to start with the development of the remaining CAMAC replacement modules within the next six months.

Upgrading of the SABUS to a 16-bit System

The choice of an upgrade path to a 16-bit system for SABUS has been a rather frustrating one, in that many of the users of SABUS systems were familiar with the IBM PC standard (its cheapness, user-friendliness, free availability of hardware and software from a multiplicity of sources, availability of Local Area Network hardware, etc.), and were seeking similar features in an upgrade of SABUS. Two years ago we began toying with the idea of using IBM compatible personal computers to control the SABUS backplane. The IBM PC I/O expansion bus differs little from the SABUS backplane, so the task of connecting a PC as a controller for SABUS was trivial. However, the available PC-compatible motherboards were not suitable for mounting in the SABUS card-cage. More recently, however, several manufacturers started marketing IBM PC XT and AT compatible microcomputers in the form of PC expansion cards. The cards are intended to be used for industrial control, and slot into a passive backplane which is compatible with the PC expansion bus. It has similarly been easy to design a SABUS card which is a passive PC expansion bus (to interconnect the aforementioned microcomputers and I/O cards such as display drivers, floppy and hard disk controllers, etc.) and which simultaneously connects to the SABUS backplane.

The virtue of this approach is that it allows us to use all our previously-developed SABUS cards, while giving us the means, via the PC expansion bus, to augment the system with any of a large choice of inexpensive commercially-available cards. Three features of the PC-based system are particularly attractive: Firstly, any software written in Pascal for our CP/M Z80 systems is easily transferable to the PC version, by a recompilation of the source code on the PC version. Secondly, the commercially-available development software already available for the PC, and increasing daily, such as windowing and graphics software, diagnostic and maintenance tools, newly-developed compilers, etc. will allow us to cut down considerably on our software development effort. Thirdly, the memory-mapped display of the PC system will allow for far greater flexibility in the displays at the local control stations. The use of standard software implies the use of the MS DOS operating system, which is not a real-time multi-tasking system. We foresee that it might become necessary to install a real-time operating system on a number of the larger systems. There should not be a problem in doing so, as there are several real-time operating systems available for PC compatibles.

Future Control System Plans

We plan to install the hardware for the interlocking of the magnet currents in the power supply area in about six months time. The hardware is SABUS-based and the microprocessors will link to the existing control system via CAMAC crates. Initially the control of power supplies will remain in the mini-computers, and the SABUS systems will only be used for the current interlocking function. Later, however, we plan to use the SABUS systems as local control stations for the power supplies. This requires much of the power supply control software to migrate to the locally-controlling microprocessors. From our previous implementations of local microcomputer control we have learnt that a local control station need consist of no more than a microcomputer with keyboard and monitor. With such control stations we hope to eliminate many of the deficiencies of the NAC control system mentioned earlier. Eventually we plan to provide microprocessor-based control stations throughout our facility, including mobile stations for use in the cyclotron vaults and near the accelerator hardware. However, this will only become a reality when we have a convenient way of linking such stations to the rest of the control system, i.e. after the installation of a local area network.

References

1. D.P. Gurd, D.R. Heywood and J.V. Cresswell, Developments in the TRIUMF Control System, Proc. Ninth Int. Conf. on Cyclotrons and their Applications, p.565.

2. G.F. Burdzik et al., Recent Developments on the Control System of the South African NAC Accelerator, Proc. Tenth Int. Conf. on Cyclotrons and their Applications, p.385.

3. K. Visser and H. Mostert, Vacuum Control Systems at the NAC Cyclotrons, this conference.

4. R.W. Goodwin and M.F. Shea, Modern Control Techniques for Accelerators, Proc. Tenth Int. Conf. on Cyclotrons and their Applications, p.547.