SLAC – PUB – 3618
April 1985
(A)

A PORTABLE DATABASE DRIVEN CONTROL SYSTEM FOR SPEAR*

S. HOWRY, T. GROMME, A. KING, M. SULLENBERGER

Stanford Linear Accelerator Center
Stanford University, Stanford, California, 94305

ABSTRACT

The new computer control system software for SPEAR is presented as a *transfer* from the PEP system. Features of the target ring (SPEAR) such as symmetries, magnet groupings, etc., are all contained in a design file which is read by both people and computer. People use it as documentation; a program reads it to generate the database structure, which becomes the center of communication for all the software. Geometric information, such as element positions and lengths, and CAMAC I/O routing information is entered into the database as it is developed. Since application processes refer only to the database and since they do so only in generic terms, almost all of this software (representing more then fifteen man years) is transferred with few changes. Operator console menus (touchpanels) are also transferred with only superficial changes for the same reasons. The system is modular: the CAMAC I/O software is all in one process; the menu control software is a process; the ring optics model and the orbit model are separate processes, each of which runs concurrently with about 15 others in the multiprogramming environment of the VAX/VMS operating system.

Introduction

This paper describes software for the new I&C computer system at SPEAR. The characteristics of this venerable storage ring are wellknown and documented[1,2] and its I&C history has been nicely described.[3] Recently, the lattice was modified[4] to provide a smaller beta at the MARK3 detector. Wigglers and undulators affecting the lattice have been inserted to create photon beams of high intensity synchrotron radiation. The number of photon beamlines has grown to 6 with 4 more coming soon. An interface to a fast feedback correction system[5] for the photon beams must be ready by fall 1985.

To cope with these diverse and changing requirements the SPEAR I&C was upgraded as follows:

- A VAX-750 computer (remaindered from another project) was acquired.
- Old interfaces were replaced by modern SLC-style CAMAC modules; a VAX CAMAC channel, the VCC,[6] provides low overhead (DMA) I/O between the VAX and the CAMAC world; PEP-style touchpanels, knobs, and graphic displays have replaced the old operator switches, thumbwheels, and crt.
- Independent trim supplies were added to provide orbit corrections every 90° of tune advance, and to create bumps at the photon beam source points.
- Completely new BPM data acquistion electronics was installed.[7]
- Software from the PEP computer control system,[8] modified to enhance portability, was transferred.

* Work supported by the Department of Energy, contract AC03-76SF00515

The rest of this paper discusses the last item above. It is interesting because of the degree to which it is driven by its database. In principle at least, as it is carried from one ring to another, *all you have to do is change the database.*

The Database

The starting point on any discussion of a database is its schema description design file. This design file is primarily a common place where all of the functional concepts are associated with database records. Each record has a unique generic name, in a notation that tends to group together similar functions with similar names. For example, a set of records associated with trim supplies should be specified as:

```
T = trim supplies in regions = east, west

Each T   has 2 orientations, horz, vert (TQ)
Each TQ  has 12 supplies (TQC)
Each TQC has:
                  class AM1  readback
                  class AC1  setpoint
                  class DI1  status
                  class D01  on-off control
```

So the readback of the fourth horizontal corrector on the west side has the unique name T2Q1C4/AM1 and the entire collection of 2*2*12*1 = 48 readbacks has the generic form TQC/AM. Software in this system processes generic forms according to their functional descriptions. For example, the power supply display program contains in its source code character strings of generic forms of power supplies. The program that calibrates DAC pedastal constants contains a generic form of all signals with both readback and setpoint classes. These functions are common to most control systems. The dimensions within the generic forms may vary from ring to ring (for example, PEP has 6 regions; SPEAR has 2), but the letters used need not be changed. The closer you can make the design files, the more complete will be the software transfer. Of course, truly different functions or signals may require new names and software; and if a function becomes irrelevant some software may be dropped.

Each record in this database is a fixed length block of 256 bytes. A header takes up the first 40 bytes; the remaining area is for attribute fields, whose (2 letter) names, sizes, and positions are specified at the very end of the design file. The resulting identical structure within all records of the database is somewhat circumvented by specifying overlapping fields. For *hardware* records like those in the example above, attribute fields for CAMAC routing, display name, pedastal constants, limits, etc., are utilized. For *software* (or class XX) records, one 216 byte data field is specified, overlapping all of the other fields.

Contributed to the 1985 Particle Accelerator Conference, Vancouver, B.C., Canada, May 13-16, 1985

Processes

Software processes are ordinary programs submitted to the batch queue of the VAX. The interrelationships are roughly shown in figure 1. The entire database structure, a VAX global shared section, is part of the image of each process. Each software process is a collection of subroutines called subtasks. A subtask can be scheduled for execution, either once or at a selected frequency. When a subtask is running none of the other subtasks of the parent process can begin. Overall throughput is achieved by organizing the collections so that conflicting subtasks (those that must run concurrently) are in different processes, and therefore get multiprogrammed by VAX/VMS. On the other hand, nonconflicting subtasks are grouped together in order to reduce the total number of VAX processes.

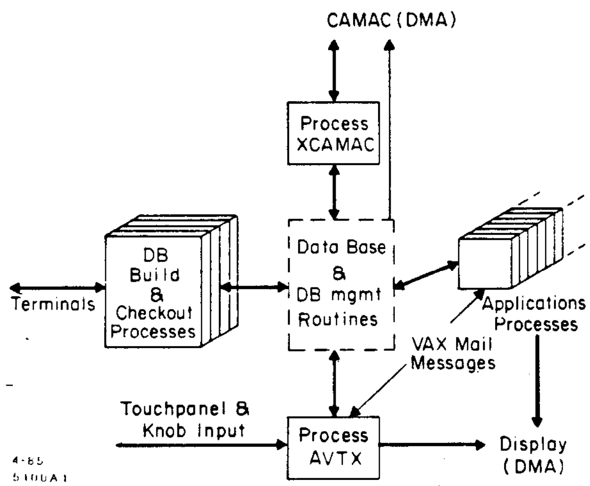

Figure 1. Database Control System

Two software processes play a special role in the system:

- XCAMAC continuously refreshes to VAX memory all database input values. A single VAX QIO command sends to the VCC a train of some 300 packets, precompiled from data in the attribute fields of all *hardware* database signals. The VCC controls this transfer at the leisurely rate of 1 word per 10-100μsec. This allows the CPU and the other DMA device, the graphics controller, enough memory cycles for smooth operation. Database output setpoints are refreshed to CAMAC in a like manner. A request to change one or more setpoints results in the entire setpoint train being sent. Ramps are accomplished by issuing this refresh at a selectable rate up to 10Hz. When not being changed, the setpoints are continuously verified by comparing CAMAC values with those in the database.
- AVTX receives and interprets all operator touchpanel button pushes and knobturns. A button push may send one or more VAX mail messages to one or more subtasks. Messages can schedule subtasks, and can deposit into or read from FORTRAN variables. Messages, in the form of FORTH sentences (see below), also can be received and interpreted by AVTX. A knobturn usually results in a weighted incrementing of a selected set of database setpoints. A knob can be also be programmed to execute a selected FORTH sentence when turned.

Touchpanel Software

Touchpanel source files describe the positions, labels, and functions of the panel buttons and knobs, and can display text and limited graphics. The files are written in a variant of the FORTH language. They can be run interpretively, or in compiled form for faster response. FORTH has proved to be a controversial choice. It is an extremely versatile language for panel design; new buttons are quickly coined from definitions in existing vocabularies, and the result is a very rich set of panels at every level of operational detail. Trivial functions, such as display of database values, are made without any FORTRAN support. On the other hand, the world seems to have a significant number of (otherwise intelligent) people with FORTH-phobia, and for these, panel design is almost impossible.

Model Processes

The beam transport model and orbit correction program run directly from defined *software* records in the database. All information as to magnet type, position, characteristics, etc. are kept in and retrieved from the database. The lattice information is loaded into the database with a separate routine that provides detailed printouts for updates and verification. After the lattice is loaded into the database, the model and orbit routines are ready to run. There is a general extraction routine that provides the lattice information from the database in a format that is usable for separate model programs. In principle, any number of model routines can be attached to the control system in this manner. All that needs to be written are *adapter* subroutines that match respective input and output formats to the database structure.

At SPEAR the model routine COMFORT[9] was adopted. This transport model routine is used to create general configurations for SPEAR, which are then loaded onto the power supplies and optimized by the operator. The model is also used to calculate dynamic modifications to machine parameters. An example is to change the quadrupoles for a 1% higher vertical tune NUY. This is often done and is quite successful. The model program is also used to compute weights for ganged knobs, permitting the operator to knob in machine-parameter space.

The orbit correction program[10] is used in a slightly different manner. It is viewed as a perturbation of the configuration that was calculated by COMFORT. The orbits are measured with the BPM system and from the database the machine functions at each element of the ring as calculated by the model are known, so that extensive correction schemes can be systematically studied with the program, these include least squares, eigenvector and harmonic analysis. An appropriate correction, change in corrector dipole magnets, is selected and a new orbit is measured and compared to the predicted change. Through several iterations of this procedure an optimized *golden* orbit is obtained. With the ability to check actual orbit changes with predicted orbit changes, comes the capability to spot magnet position errors, magnet field strength errors, etc. The program showed its value recently on the PEP storage ring, where the above technique was used to isolate a magnet field error of a single quadrupole (due to a shorted coil connection) without leaving the control room. The alternative would have been to individually test a thousand powered magnets, a dangerous and time consuming prospect.

Plan for the Transfer of this Package

This section loosely describes steps that might be used to transfer the software from SPEAR to another storage ring. Although effort is obviously required (there is no free lunch), it is only a fraction of the 15-20 man-years expended on the original projects. It is assumed in the example that both rings have similar I&C hardware. The VAX is essential because of the heavy dependence on VAX/VMS. Non-CAMAC (or non-VCC) environments could be accomodated by rewriting process XCAMAC using different database attribute fields.

- edit the database design file for the new ring.
- run package routine DBGEN to build the new database records.
- enter display names, routing data, nominal ADC, DAC pedastals, etc.
- calibrate transductors and enter the resulting ADC pedastal data.
- run package software to calibrate DAC pedastals.

This is level 0 of operation; at this point there is package software to run setpoint and readback displays; to create any selected combination of ganged knobs; to save, get, ramp or touchup *hardware* configurations (where only the setpoint values values are known). Note that this level of operation can be transferred to any control system, not necessarily a storage ring.

- enter geometric data, and description of the physical lattice; use package software to get an expanded ring layout for verification.
- enter magnetic measurement calibration data, such as B vs I curves, effective lengths, etc.
- write the custom software program to compute the conversions between the *strength* and *hardware* configurations.
- modify the standardize package program for the new magnets.
- use a package routine to enter a seed *strength* configuration, i.e. - one that has been computed and simulated externally by some model program.
- run the package model from the touchpanel to verify that the configuration agrees with simulation; save this configuration; ramp magnets to it and inject beam into it.

This is level 1 of operation; at this point there is package software to print detailed information about machine lattice parameters, and to dynamically change them; to save, get, ramp or touchup *hardware*, *strength*, or *model* configurations.

- modify the two touchpanels that patch correctors, BPM's into the orbit program
- enter BPM calibration data and modify the BPMSCAN program.
- make BPM scans from the touchpanels and look at orbits.

This is level 2 of operation; at this point all of the software should be usable; it is invoked from various touchpanel displays.

REFERENCES

1. M. J. Lee, et al. "Nov7 SPEAR Lattice design" SPEAR Note 102 May, 1971
2. A. Boyarski, et al. "Automatic Control Program for SPEAR" *IEEE Transactions on Nuclear Science* NS–20 No. 3, (1973)
3. R. E. Melen "Centralized Digital Control of Accelerators" Lecture notes in Physics, #215, Proceedings Berlin 1983 Springer-Verlag.
4. K. Wille "A Proposed Mini Beta Insertion for SPEAR" SLAC/AP–009 March,1984
5. R. O. Hettel "Beam Steering at the Stanford Synchrotron Radiation Laboratory" *IEEE Transactions on Nuclear Science* NS–30 No. 4, (August 1983)
6. D. Nelson, M. Briedenbach et al . "The Vax Camac Channel" *IEEE Transactions on Nuclear Science* NS–28 (1981)
7. See J. L. Pellegrin, this conference.
8. R. E. Melen "The PEP Instrumentation & Control System" *Proceedings of the 11th International Conference on High Energy Accelerators*, pp 408–420 (1980)
9. M. D. Woodley, M.J.Lee et al. "Control of Machine Functions or Transport Systems", *IEEE Transactions on Nuclear Science*, NS–30 No. 4, (1983)
10. M. H. R. Donald et al. "Some Schemes for Online Correction of the Closed Orbit, Dispersion and Beta Functions in PEP" *IEEE Transactions on Nuclear Science* NS–28 No. 3, (1981)