CONF - 791218--2

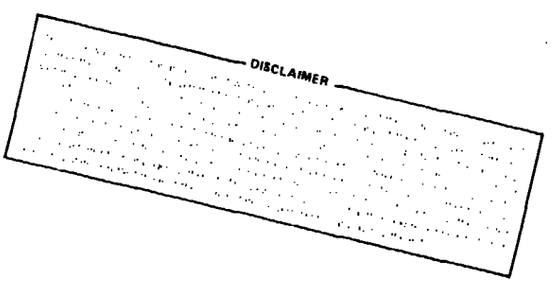
LA-UR - 79-3276

TITLE: BUILDING CONTROL FOR NUCLEAR MATERIALS R&D FACILITY

AUTHOR(S): Orval Hart

SUBMITTED TO: Data General Users Group Annual Meeting
New Orleans, Louisiana
December 4-7, 1979

MASTER



University of California

LOS ALAMOS SCIENTIFIC LABORATORY
Post Office Box 1663 Los Alamos, New Mexico 87545
An Affirmative Action/Equal Opportunity Employer

Form No. 836 R3
St. No. 2629
12/78

UNITED STATES
DEPARTMENT OF ENERGY
CONTRACT W-7405-ENG. 36

BUILDING CONTROL FOR NUCLEAR MATERIALS R&D FACILITY*

Orval Hart
Los Alamos Scientific Laboratory
Los Alamos, New Mexico 87545

ABSTRACT

The new plutonium research and development facility at LASL was the first facility to be completed in the United States under the new environmental requirements. To insure that these new requirements are met, a redundant computer system is used to monitor and control the building. This paper describes the supervisory control and data acquisition system that was implemented to perform that function.

Background

In the mid-60's, it was recognized by Los Alamos Scientific Laboratory (LASL) personnel that the old nuclear materials research and development facilities were becoming inadequate. Fluor Engineering, Inc. was chosen to design and build a new $75 million facility to meet all the recent environmental requirements for our area--earthquakes and tornados. After the design, LASL decided to take over management of the project and implement the original Fluor design.

The original design called for the computer to control the building, then to act only as a data logger, and then again to control the building. The system grew from about 1000 logging points to over 3000 channels of monitoring and control points to include the fire, radiation, switch gear (power distribution), and heating and ventilating systems. Due to the stringent environmental requirements, no vendor wanted to bid on the entire system, so the system was acquired in three parts: computer, display generator and data multiplexer, and software. All equipment had to be seismically qualified. The computer and multiplexer were placed on shake tables and subjected to design base earthquake conditions. The computer was actually running EMORT Long during the test (one recoverable disk error encountered).

*This work was performed under the auspices of the US Department of Energy.

The engineers with whom we had to work had no prior experience with computer-controlled systems, so we had to educate them and take our best stab at what we thought would satisfy them. At about this time, LASL received an unsolicited proposal from Rockwell International. Inc. to supply the software from some work they were doing for power utility systems. We could not afford their completed package, but decided to buy their prototype version (it was working but not completely debugged). After much haggling between legal representatives, we were finally able to get the sources on tape. Within 4 weeks we had it cycling on our system. The initial areas of change were in the data-acquisition and display generation portions, as our hardware was not the same as theirs. Eventually, the whole system was overhauled considerably, the only thing that remained unchanged was the outward appearance of the system, that is, the man-machine interface.

Configuration

The system as originally ordered consisted of two S/200s with 48-k memory, two pairs of Diablo front-loader disks (device codes 33 and 73) with the dual port option, foreground teletype I/F to back up the IPB should it fail, and on a bus switch a tape unit, card reader, and Versatec printer/plotter. Added to this were our own custom interfaces for the display generator (one slot) and multiplex system (two slots), all of which were DMA interfaces. The memory grew quickly to 96-k words, and recently we acquired two 2MB fixed-head disks (device codes 26 and 66) with dual port option and two Dasher LP-2s with DMA I/F (device code 57), one for each machine (see Fig. 1.0).

The system is currently running under RDOS, Rev. 6.45, with each system running off of one of the fixed-head disks as its primary device, with logging going to one of the Diablo disks. The remaining disks are used for program development, data reduction, etc. Hardcopy logging is performed on the LP-2s (original logging went to 30 CPS TI-733 terminals, sometimes causing a considerable backlog).

There are two Aydin dual ported display generators, each with two CRTs. Either computer can drive the displays, but only one at

a time is advised. High-resolution color CRTs are used to display the current status of the building. The display generators receive data from the computer through the DMA interface at 400-k bits/second, giving instantaneous screen change capability.

Data Input

Data acquisition for the system is performed through two high-speed (600-k bits/second) multiplexer controllers that are ping-ponged after each scan of the data points. Each multiplexer controlier is connected to 23 remote multiplexers by alternate paths. Each multi plexer may contain a variety of function cards based on the data it has to acquire. The separate paths are to offer an alternate path for acquiring the data should one of the paths be damaged due to accident, etc. and redundancy in general.

Multiplexer interrogation is performed by address ROMs stored in each controller and the data is passed to the computer, if requested, in the same order. The computer also has no control on the rate at which the controllers ping-pong or will return data to the computer. Also, it is difficult to retrieve good data if both computers are attempting to acquire data at the same time.

The type of data acquired by the system includes analog status input (12 bits, bipolar), pulse accumulator input (12 bits, 4096 counts), contact status input (12 bits), and contact control output (12 bits). There are a total of 336 ASI channels, 280 PAI channels, 1524 CSI channels, and 888 CCO channels, which are used for such things as PDTs, temperatures, alpha radiation counters, and breaker/switch status and control (see Fig. 1.1). Exception checking is performed on the data and where data have changed, the appropriate application programs are operated.

The complete data base is core resident, residing in extended memory. Directories for the analog and status data are kept in lower memory. The multiplexer address is used to locate a card entry in the directory, which in turn points to the card data in extended memory. The data are then mapped into the virtual window for access by the system (see Fig. 1.2).

Redundancy

All major systems are dual ported for redundancy, although they are not used actively to any great extent. For example, we do not normally access the on-line disk from the backup computer, mainly due to the extra overhead this creates, but also due to the problem that if the backup computer releases the on-line disk, it is possible that all of the primary machine's system devices will disappear.

The custom hardware devices are also dual ported so that either computer may access the appropriate interface. However, only the primary computer is allowed to access a given device, due to the confusion generated. As mentioned earlier, the primary computer operates off of one of the fixed disks (for example, DS0) while the other operates off of the second fixed disk (DS4). Data archival is then performed on one of the Diablos (for example, DP4). Currently, both primary and backup computers log to the same disk, but in the future this will be changed so the backup computer will log on a separate disk (for example, DP1), mainly because it may have failed over to the backup when the primary logging disk went down. Each computer has its own hard-copy logging printer operated as $LPT1 in the DMA mode. The Versatec is operated in the PIO mode through the bus switch by the backup machine, and used mainly for program development, data reduction, etc.

The watch-dog timer in the dual port option is used to determine if the primary computer has halted (actually not very likely). In the backup computer, a foreground program is monitoring the watch-dog timer to see that the primary computer is keeping it updated. If the watch-dog timer is not updated at least once a second, an interrupt is received by the backup computer indicating such. The monitor then kills whatever activity is going on in the background, and returns to the command line interpreter in such a way that a CLI command file is executed. This command file releases the memory previously assigned to the foreground task, tries to move certain files from the primary to the backup disk, and executes the supervisory control and data-acquisition (SCADA) system in the background of the backup

computer. Unfortunately, the probability of this happening is very slim. The major problem is with SCADA locking up in some sort of loop or reaching some illogical result. In this case, the user has no control over the watch-dog timer being updated, so some alternative scheme must be used. This involves implementing a software watch-dog timer with the backup computer through the full duplex link between the computers that is provided as part of the dual port option. The primary computer then sends an "I'm here" message to the backup computer once a second. If the backup computer does not receive an "I'm here" message for a period of 10 seconds, then the backup assumes that the primary has died, and takes over just as it the primary computer had halted.

After-Hours Operation

When the multiplex system goes bad (for example, a multiplexer starts returning bad data--a card has gone bad), it is sensed by the data processors, and notification is passed to the Configuration Status Program (CSP). If the CSP determines that too much bad data are being received for the system to make meaningful use of it, SCADA will retire to the "monitor mode" and not try to actively control the building. The criteria for going to the monitor mode is the receipt of bad data from a card or point eight consecutive times. To get out of the monitor mode, one complete pass of good data must be received.

Even though active control of the building is discontinued while in the monitor mode, the system is still passively maintaining the good part of the data base. Should any fire or criticality alarms be received, SCADA will attempt to send the alarms, etc.. regardless of the monitor mode state.

RDOS Extensions

The multitasking nature of RDOS is exploited to its fullest, with the system using some 40 tasks. Due to certain shortcomings in the task handling area. however, modifications had to be made to the task handlers. If you are doing window mapping under RDOS, the state of the window is not maintained between tasks. To insure that

the window did not change when interrupted by a higher priority task, the remap call was modified to save the map address in an unused word in the task control block. Then when a reschedule occurs, the window is restored to its state upon interruption.

The second problem involves the floating point registers, which are also not saved in a multitasking environment. However, in this case, there was no room available in the task control block. To get around this, most tasks were given task identifications (ID). Each task has an entry in a table (ordered by task ID) describing its priority, task ID, and a possible floating point save-area address. If a task is interrupted that contains a floating point save-area address, then the floating point registers are saved. Upon reschedule they are restored.

Another extension to the multitasking capability was the addition of dynamic queue processors. This allows tasks to be transaction oriented by front ending specific tasks with queues. Each main task is controlled by a queue task controller (also a task) that accepts queueing from other tasks, pushes the data onto its queue, and if the main task is not running, queues it. The queues are either one or two words each. A task operates on its queue by popping off a one- or two-word transaction, processing it completely and popping off the next transaction. When the queue becomes empty, the task kills itself until activated again by its queue task controller.

Man-Machine Interface

Control of the SCADA system is through the color CRT consoles. There are four CRTs in the system, two in the control room and two in remote offices. The remote CRTs are not allowed to perform control functions, for example, control a breaker, etc.

The CRT screen is divided into four regions (see Fig. 1.3 for example):

(a) Two-line header--common to all display formats.

(b) Forty-two-line body--available to user for display format definition and display.

(c) Two-line function register--used to execute functions applying to points that have been poked in the body, for example, close/open breaker, acknowledge/delete alarms. etc.

(d) Two-line alarm register--contains the two most recent outstanding alarms.

Control of the system is accomplished through "poke points," that is, cursor-sensitive coordinates on the CRT screen. These poke points may either be explicit (a specific symbol) or implicit (a breaker symbol that is known to be controllable). Poking a point is the process of positioning the cursor at a given x,y coordinate on the screen (for example, a breaker) and pressing the transmit cursor key. The computer then reads the cursor coordinates and determines from an internal table of coordinates the function to be performed.

There are normally no typing functions involved with controlling the SCADA system, the exception is real-time alarm limit changes and manual control through the computer console.

Displays are organized in a tree fashion where the top node of the tree is the master index display. The next level consists of nodes for each of the applications areas, that is, fire, radiation, heating and ventilation, and miscellaneous (switchgear). Each application area has two subnodes, schematic and tabular, containing the appropriate displays. The master index display can be accessed from any display by "poking" INDEX. From there the operator may select the specific display desired, or in some cases, the first display of a major category, for example, a series of ladder diagrams for a particular fan combination.

Building Control

The system is responsible for four application areas:

(a) Fire--monitoring heat detectors, smoke detectors, etc. and maintaining integrity of the fire system by periodically supervising (controlling) the fire signals to see that they work and sending alarms when appropriate.

(b) Radiation--monitoring alpha and gamma radiation sensors and sending alarms when appropriate.

(c) Heating and ventilation--monitoring the pressure zones within the building and through control of fans, maintaining the appropriate pressures, and generating alarms when appropriate.

(d) Miscellaneous--mainly concerned with monitoring power distribution within the building, and switching to alternate power sources when necessary, including the backup.

Most of this control is maintained through the use of ladder diagrams that describe the control logic and action involved with the different areas. These diagrams have been translated into tables that are processed by a special program whenever something changes in the data base for the specific application areas.

Ladder diagrams are a pictorial form of Boolean algebra that is used by mechanical engineers to describe control logic for a function. All control logic for heating and ventilation and switchgear was defined in this form, so a special table format was created to accommodate this form. Special "macros" are used to create these tables and a special program (Circuit Description Compiler) is used to print them in ladder diagram form (for error checking).

A ladder diagram is made up of any number of circuit types, each of which may have any number of circuit descriptions. The circuit type describes the logic "map" and the circuit descriptions furnish the multiplexer address to go with the points in the equation. As this was already such a large part of the system, almost all other control logic was converted to this form, for example, the fire system.

Conclusions

The system has taken 4 years to refine and bring to a reasonable state of performance. In fact, we are just now getting to the state where we are not receiving hourly change requests from the engineers.

The major problem, besides engineering change requests, has been anticipating the problems that can be caused by unexpected responses from the custom hardware devices, for example, a whole multiplexer going bad.

The users are relatively pleased with the results so far. In fact, one of the engineers has gone so far as to say that "when the system is operating right, it is far better than a hard-wired system would have been" (obviously one of the engineers that had to live through the teething period). The system gives the users a considerable amount of flexibility and currently more capability than they know how to take advantage of. Currently, the system is heavily loaded due to the number of high-speed input/output devices attached to the system. A considerable amount of this load could have been avoided by the use of intelligent multiplexers, which were just coming on the market when the system was being built.