The Distributed Control System for the Fermilab 200 MeV Linac*

R.W. Goodwin and M.F. Shea

Fermi National Accelerator Laboratory
P.O. Box 500
Batavia, IL 60510

Summary

A new MC68000-based distributed control system is currently being assembled and installed for use on the Fermilab 200 MeV Linac. Sixteen local Secondary stations are interconnected by a 1-MHz, fiber optic, serial (SDLC Loop) communications link to a Primary station. This Primary station interfaces to the Host computer and collects data from the Secondary stations on behalf of either the Host or other Secondaries. IEEE-P796 (Multibus†) compatible hardware is used throughout. The system features synchronous 15-Hz response, extensive integral local control capability and provision for display and control of remote parameters from any Secondary console.

Introduction

The preliminary design of the new distributed control system for the Fermilab 200 MeV Linac has been described in Ref. 1. Although much of the system remains the same, the processor type has been upgraded from an 8-bit MC6809 to a 16-bit MC68000. This paper describes the hardware and software implementation of the system along with some measured performance characteristics and the present status of the project.

General Description

The new Linac control system is a distributed network of sixteen Multibus-based local Secondary stations linked to a Primary station using an SDLC Loop protocol. Each Secondary station contains an MC68000 processor card, the I/O interface cards to read and control all the equipment in a local area of the accelerator, and a small keyboard/video console to allow operator interface to the local station and the network. The Primary station controls the communication on the Loop, collecting data from the network and returning it to the requestor. The requestor may be either a Secondary or the Host computer that supports a console in the Main Control Room. Two MC68000 processors are used in the Primary station; one drives the SDLC link and the other interfaces with the Host computer.

To operate as a stand-alone control system each Secondary must maintain its own portion of the distributed data base that includes device names, analog settings and calibration constants, and digital control and reading characteristics. This data base resides in non-volatile core memory at each Secondary and can be modified locally or downloaded from the network. To participate in network activity the stand-alone Secondary is simply inserted into the communications loop. Programs for Secondary stations are stored in PROM memory so that a Secondary will recover from a power outage in an operating state independent of the communication link.

The systems described here all operate synchronously with the Linac 15-Hz repetition rate. Each Secondary acquires all its data each cycle and scans the data for out-of-tolerance values or improper status conditions. New alarm messages that result from this monitor scan are returned to the Primary during the alarms poll. From any Secondary console an operator can simultaneously display and control a selection of parameters from the local and several remote Secondaries. Careful attention has been given to preserving the fast response and interactive "feel" of the system. Data are returned to the requestors at 15-Hz and settings are sent to the appropriate Secondary at 15-Hz so that the effects of an adjustment can be noticed on the next cycle. The 1 MHz SDLC link and its supporting software drivers easily support more than 100 messages each fifteenth of a second.

Hardware

Secondary stations are complete stand-alone local control systems that include the computer, I/O and a small console. The following is a brief description of the hardware components.

Each station contains an Intel ICS80 Multibus chassis that houses the processor card, a 32K byte core memory card, the communication card, and binary and analog I/O cards. The processor card is a Fermilab design because no MC68000-based Multibus board was commercially available until recently. The card contains two serial RS232 ports, four bytes of parallel I/O, a three channel timer and eight pairs of byte-wide memory sockets for RAM and ROM.

Commercially available Multibus-compatible D-A converters are used where 12-bit bipolar D-A control is needed. External 16-channel unipolar 10-bit D-A chassis are used to control the 171 drift tube quadrupole power supplies.

Nearly all Linac analog signals are pulsed, so a modular 16-channel sample-and-hold/A-D chassis was designed. In a typical station four

*Operated by the Universities Research Association, Inc. under contract with the U.S. Department of Energy.

†Multibus is a trademark of the Intel Corp.

of these chassis are daisy-chain connected to a binary I/O card in the Multibus chassis by a single 50-pin ribbon cable.

All the binary I/O requirements for this system are met using a single binary I/O card design. This card provides nine bytes of data organized as three groups of three bytes. Each group interfaces to a 50-pin card edge ribbon cable connector. Individual bytes can be input or output; output bytes can be read back, so they appear as memory to the processor. The pinout of the card edge connector is compatible with Opto22 style isolated I/O modules used to replace some of the relay-based controls of the RF systems. The characteristics of each bit are controlled by the software. This allows a bit to be dc or pulsed, active high or low, and short (≈20 μs) or long (nx66 ms) pulse. Short pulsed outputs are used for stepping motor controllers; long pulses may operate large contactors.

A communications card interfaces the computer with both the console and the SDLC link. A 16 line by 32 character video RAM display generator is included on this card to drive the 5" CRT on the local console. This provides a much faster interface for supporting page-style updating displays (equivalent to a 640K baud terminal). A serial interface outputs two bytes of data to the console to control the lighted pushbuttons. Data input from the console are two bytes of switch status, one byte of data from the ASCII encoded keyboard and one byte from an up-down counter connected to the console shaft encoder knob.

A Motorola MC6854 LSI controller performs much of the protocol logic needed to receive and transmit messages on the SDLC loop. Link data are read into and transmitted out from the 32K core memory board under DMA control. Because the Multibus may not be immediately available to the DMA controller, a 16-byte FIFO is used, in both the transmit and receive channels, to provide about 125 μs of temporary storage. A 26-pin card edge connector interfaces the communication card with the link repeater chassis.

A fiber optic light link interconnects Secondary and Primary stations. A separately powered link repeater chassis is used so the computer can be powered-down without interfering with other activity on the link. As a result of using a fiber optic link to interconnect the Secondaries, the two stations for the pre-accelerator high voltage domes can be the same as all other Secondaries. Control of the ion source equipment can be accomplished from any Secondary console.

Selected Data Scheme

Central to the design of the new Linac Control System software is the plan devised to provide lists of selected data items to many requestors at 15-Hz. Such a request must be sent ahead of the time the data is needed in order to prime the system to deliver the list of requested data efficiently. It is clear that using loop communications could cause the Primary to become a bottleneck to efficient message transfer since every request must pass through that system. To alleviate this problem the Primary, without taking time to analyze each item requested, broadcasts the request to all Secondaries. During that same cycle, each Secondary scans the entire request, noting those items which belong to its own system, and builds an index list of relative pointer values into the answer list which will ultimately be returned by the Primary to the requestor. The following cycle the Primary polls for index lists for each newly-created list and then polls for answer lists for all active lists to be polled according to each list request's update frequency. The index lists are saved and used by the Primary to place the final answer list in the original request order. The answer list is then delivered to the requestor, whether it be the Host, the Primary Console or a Secondary.

Secondary Software

Secondary station software is divided into two kinds of modules - interrupt routines and tasks as shown in Figure 1a. Interrupt routines are scheduled by the hardware priority interrupt system to support SDLC communications (highest priority), serial console communications at 4800 baud, stepping motor pulsing at 150-Hz and Linac data acquisition at 15-Hz. Eight tasks are sequenced by a simple non-preemptive round-robin multitasking scheduler. Each task has a separate data region for its stack and local variables which may be shared by as many as eight subtasks which are priority scheduled by the task itself.

Interrupt Routines

The SDLC Driver interrupt routine operates at the highest priority level to receive messages from the Primary into the input queue under DMA control. If the message is a poll message the driver sends the response frame immediately. Poll messages are requests for answer lists, index lists, commands to request new data lists or control devices, alarms and a diagnostic test. Usually poll messages are broadcast to all Secondaries. The responses from each Secondary are sent to the Primary in physical link order according to SDLC loop protocol. If the message is not a poll message then the Link Command Task is triggered to process it further.

The Timer interrupt routine initiates serial console communication by sending control commands to update the console switch lights. The response data triggers the Console Link interrupt. Another timer channel is used to generate 150 - Hz interrupts used to send pulses to any active stepping motors.

The 15-Hz routine acquires all the analog and binary data in that Secondary. The exact parameters of the data acquisition are specified in a Data Access Table to facilitate making device changes and to help make it possible for all the Secondaries to have the same software. Five tasks are triggered as indicated in Figure 1a. The

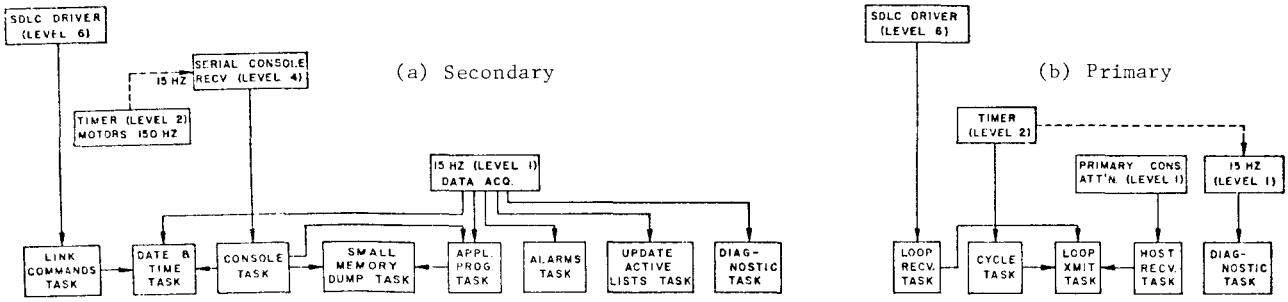

Figure 1. System Interrupts and Tasks

15-Hz routine executes in 3 ms for a system which has 64 analog channels and 24 bytes of binary status.

Tasks

The eight tasks used in one Secondary are: Link Command, Alarms, Console, Application Program, Small Memory Dump, Date & Time, Update active data requests, and Diagnostics.

The Link Command Task is triggered by the SDLC interrupt driver upon receipt of a non-poll message. Link commands from the Primary include requests for repetitive data lists or for device control on behalf of either the Host computer or another Secondary. Most other messages are answer lists to a Secondary's active data requests.

The Alarms Task scans all active analog channels and binary status bits each cycle for changes in alarm conditions. The time to scan 128 analog channels and 192 binary status bits is 4.2 ms with 40 active channels and 40 active bits.

The Console Task processes keyboard input characters, cursor activity, knob counter accumulation, switches and lights logic, display page selection, and page title changes. The console switch lights are under software control so that only switches which are used by a given page will light up when pressed.

The Small Memory Dump Task, when active, is used to display eight bytes of local Secondary memory on the bottom display line as a diagnostic aid. Applications programs do not use the bottom display line, keeping it free for system use.

The Application Program Task simply provides a task environment for the application program which is currently active. Application programs are written in PASCAL, compiled, linked, downloaded from the Motorola EXORmacs development system and programmed into PROMs. At present there are four application programs. The Parameter Page program displays an arbitrary list of up to 14 analog channels from any set of Secondaries in the network. Console switches select readings, settings, nominals, tolerances, channel numbers, or settings upon entry to page, to be displayed in engineering, volts, or hex units. Analog control may be effected with keyboard entry or by raise/lower switches or knob control with 15-Hz updating. Optional analog-associated digital status and control may also be supported for each channel. Normal updating displays averages of readings on beam pulses over the last 13 cycles. These characteristic features have evolved over 12 years of experience with Fermilab accelerator control systems. In fact, the Secondary console switch options were designed around the requirements of the Parameter Page. The Analog and Binary Descriptor Pages allow one to display, list, or change any Secondary's data base. The Memory Dump Page is a diagnostic tool which displays 64 bytes of memory organized as eight lines of four words. Each line may separately refer to any eight contiguous bytes of data anywhere in the network. The display is in hexadecimal or volts units and updated at 15-Hz.

The Date and Time Task keeps track of the time of day by counting 15-Hz triggers and updates the display on the top line each minute. The Update Task is triggered each cycle after local data has been read to update answers for all active Primary data request lists. The Diagnostic Task optionally displays local alarm messages on the bottom display line.

Primary Software

Primary Link Driver software is also divided into interrupt routines and tasks as in Figure 1b. Interrupt routines support SDLC loop communications and provide for scheduling poll activity for specific portions of the 15-Hz cycle. Five tasks are sequenced by the same multitasking scheduler as used in the Secondaries.

Interrupt Routines

The SDLC Link interrupt routine is used for loop communications. Messages from the Loop Xmit Queue are sent around the loop. Responses received from poll messages are collected under DMA control into the Loop Receive Queue, and the Loop Receive Task is triggered.

The Timer chip on the CPU board is used to schedule loop communications within each 66 ms cycle. The 15-Hz interrupt occurs about 1 ms after Linac beam time and establishes a reference for Timer chip scheduling. The Timer interrupt triggers the Cycle Task whose subtasks queue the appropriate poll messages for the Loop Xmit Task. Current times used are:

Poll indexes for new lists	10 ms
Poll answers for active lists	15 ms
Diagnostic Test poll	36 ms
Poll for commands	41 ms
Poll for alarms	51 ms

The Primary Console CPU resides in the same Multibus crate as the Primary Link Driver and is the link to the Host computer. If a command is sent from the Host the Primary Console queues it to the Host Command Queue and interrupts the Primary Link Driver which merely triggers the Host Receive Task. The Primary Console system may also queue commands in the same way.

Tasks

The five tasks used in the Primary are the Host Receive Task, Loop Xmit Task, Loop Receive Task, Cycle Task, and the Diagnostic Task.

The Host Receive Task removes messages from the Host Command Queue and the Primary Console Command Queue. It recognizes only two types of messages - requests for new data lists and setting commands. This underscores the simplicity of the Host interface as these messages merely represent read and write access to accelerator data. Requests for new data lists will, on succeeding cycles, result in answer lists being sent to the Host. Setting commands are simply passed on to the Secondary addressed by the command.

The Loop Receive Task processes all responses received due to Primary polls of the Secondaries, including index lists, answer lists, commands, alarms and test poll responses. Index lists are saved in the memory block which supports that data list. Answer lists are arranged in proper sequence using the previously saved index lists and then delivered to the requestor. Commands are requests for data lists or setting commands on behalf of a Secondary. They will be queued for the Loop Xmit Task. Alarm messages are queued to send to the Host via the Primary Console. An attention interrupt alerts that CPU to examine the Host Response Queue. The test poll responses are used to determine if all Secondaries are able to communicate on the loop.

Present Status

The Primary to Host connection must await the completion of the new central control computer changeover. However, we are installing Secondary stations and interconnecting them to the Primary on the SDLC link. We presently have all nine Linac systems, the 750 KeV beam line and the preaccelerator ground station operating. We plan to have the Host computer working with the new Linac system before the analog and binary I/O connections are made to the Linac hardware.

As a test of the communications software we called up Memory Dump Pages on 10 Secondaries each of which requested data lists of eight bytes of memory from eight other stations. The time required for the Primary to poll for the 10 answer lists, arrange them into the correct order, and deliver the results to the 10 requesting Secondaries (a total of 100 messages) was 36 ms.

Discussion

In summary, the system described here promises to provide reliable responsive control for the Linac. SDLC, because it is an intercomputer link standard, supported by available LSI circuits, easily handles the high speed variable length memory to memory data transfers required. The choice of MC68000 microprocessor has been viewed by some as overkill for a Secondary station. It does not increase the cost, so the real question is "Why not use the MC68000?" It has been our experience that considerable time and effort is saved by using a more capable processor. Traditional problems of processor speed, limited address space and error detection and recovery are alleviated by the inherent design of the processor itself.

The software development has been done using a Motorola EXORmacs multi-user development system. The system has been reliable and efficient to use with its screen editing facilities and hard disk. Application programs written in PASCAL are more readable and easier to debug and maintain than we have experienced before.

Modularizing the design so that a Secondary connects to all data in a geographic area rather than by function was obvious in this case since the old Linac control system was organized by area. Nevertheless, a new system design could profit from the same approach in that stand-alone operation can support an entire local area of the accelerator, and analog signals are all close to the digitizer and the microcomputer.

Acknowledgements

We would like to acknowledge the continued support of this project by the Accelerator Division Heads Russ Huson and Rich Orr and the Controls Group Leaders Mike Gormley and Dixon Bogert. We want to recognize the contributions of Bob Florian, Al Forni, Al Jones, and Jay Ticku by way of their enthusiastic efforts in the design, fabrication, checkout and installation of the hardware components of this system.

References

1. R.W. Goodwin and M.F. Shea, A Distributed Linac Control System Featuring SDLC Loop Communication, Proc. of the 1979 Linear Accelerator Conference, Montauk, NY, pp. 274-278 (BNL-51134).