LBL-9152

CONF-790549--10

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA, BERKELEY, CA

Presented at the Topical Conference on Computerized Data-Acquisition Systems in Particle and Nuclear Physics, Santa Fe, NM, May 14-17, 1979

DATA ACQUISITION USING PDP-11 AND MBD BRANCH DRIVER AT L.B.L.

Everett H. Harvey, Jr.

May 1979

MASTER

Prepared for the U. S. Department of Energy under Contract W-7405-ENG-48

DATA ACQUISITION USING PDP-11 AND MBD BRANCH DRIVER AT L.B.L.

Everett H. Harvey, Jr.*

ABSTRACT

A data acquisition system has been designed and preliminary versions have been implemented for physics experiments at LBL. It utilizes an MBD microprogrammed branch driver and a PDP-11 operating under RSM-11M. An MBD executive is used to implement software breakpoints within the MBD. The PDP-11 data acquisition buffer management scheme allows for multiple tasks dynamically accessing the data stream. Emphasis is directed towards future expansions. High data rates are also emphasized.

INTRODUCTION

A data acquisition system for physics experiments at the Lawrence Berkeley Laboratory BEVALAC has been designed and implemented. It utlizes a PDP-11 computer with an RSX-11M operating system and an MBD Microprogrammed Branch Driver Interface to CAMAC.

This introduction of actual distributed processing in the MBD as well as the virtual distributed processing provided by RSX-11M has prompted the development of a method of memory sharing oriented towards multiple data streams and multiple processors.

The distributed intelligence provided by the MBD allows for high performance within a generalized framework.

This paper will concentrate on some of the features of the MBD as a separate processor and upon the memory sharing technique of buffer management.

Some preliminary performance test results will be presented.

BACKGROUND

Physics experiments at the BEVALAC require both high performance and flexibility. Data is typically organized to an 'event' structure where as event consists of a variable number of data words. These events can range from a few words of data up to several thousand words of data. To ease the data management problems each event will be prefixed by a word count and an event type identifier. The typical experiment would require as many of these events as possible during the allowed experimental beam time.

Since the experimental apparatus and the CAMAC interface are capable of providing data at rates which alone can saturate most computer systems, each event is recorded on magtape or some other storage media for later analysis. However, there is a need for a quick analysis to determine if the experiment is working as expected. This analysis portion is not being considered in this paper except to address the fact that there are many levels of computational effort required. What is discussed is a method of passing data to user coded analysis tasks.

SYSTEM DESCRIPTION

The MBD Subsystem

The CAMAC interface is an MBD branch driver which is capable of addressing the lower 64 thousand words of PDP-11 memory via Direct Memory Accesses. It also has an interface to the CAMAC LAM signals. The MBD has its own instruction set and there exist several manuals on programming the MBD.[1,2]

Programming the MBD is normally a difficult task since there are no communications with the "outside" except via the PDP and CAMAC bus systems. However, since the MBD controls CAMAC and the CAMAC configuration is changed the most often as experiments change, there exists a need for good MBD programming tools. This data acquisition system provides for the following MBD programming features.

MBD Programming Tools

1. An RSX-11M QIO callable driver for the MBD supporting the following functions:
   - (a) Downloading user written MBD code
   - (b) Data transfer from PDP memory to MBD memory and vice versa
   - (c) Data transfer from PDP memory to MBD registers and vice versa
   - (d) Initialize MBD execution
   - (e) Specify PDP-11 event flag numbers and/or AST routines to allow interaction with the RSX-11M operating system
   - (f) Stop MBD execution
   - (g) Report on MBD status
2. An MBD executive which communicates with the PDP-11 QIO requests
3. An interactive dubugging utility program supporting the following features:
   - (a) Display MBD status
   - (b) Display and modify MBD memory and registers
   - (c) Two independent breakpoints
   - (d) Program execution control: GO, CONTINUE and SINGLE STEP
4. A skeleton MBD program for communication with a data acquisition task in the PDP-11.
5. A macro assembler library for producing MBD code on the host PDP-11.[3]

---

*Real Time Systems Group
Lawrence Berkeley Laboratory
1 Cyclotron Road
Berkeley, California 94720

This work was funded by the Department of Energy.

6. An utility program for downloading MBD code from files maintained by the host PDP-11

The PDP-11 Subsystem

The data acquisition program in the PDP-11 is designed as a circular buffering program to obtain data from one or more input sources and move that data to a permanent storage output device such as magtape or disk. Standard contron features such as start, stop, suspend and resume data taking are provided.

Buffer Management Scheme

The data acquisition system may be viewed as a memory allocation program in that it manages contiguous sections of PDP-11 memory referred to as buffers.

These buffers can exist in one of three states:

1. A buffer may be in-use by the data acquisition subsystem.
2. A buffer may be in-use by a user level or analysis task.
3. A buffer may be 'free' and available for use by any task.

In order to implement this basic allocation scheme in a method which allows multiple independent accesses to the data, each buffer is given a semaphore to describe the state of that buffer. The association chosen was as shown in Table 1.

Table 1

Buffer Semaphore States

Semaphore	State	Controlling Tasks
Negative	Contents changing	Data acquisition system
Positive	Contents frozen	User tasks
Zero	Available	None

In addition then, all that is required is some pointers to buffer locations and lengths. These are collected with the semaphores into a buffer control block. Then any buffer in a positive or zero semaphore state may have its semaphore incremented by a user task and that buffer will have stable contents until the task decrements that buffer's semaphore. All that is required is access to the buffer descriptor block and associated buffers. Each event placed in the buffers will have an identifying, ascending sequence number so that each task can monitor its own progress through the data. Implicitly it is assumed that any buffer available has contents which are correct in form.

There is a routine provided that may be included in a user task which will perform this buffer locking. The buffer may then a analyzed rapidly or copied and the buffer then released. This would provide a sampling of the data stream.

By providing a specified switch this same routine can lock the next buffer in the circular buffer sequence before releasing the previously locked buffer thereby insuring that all data appearing in the buffers is processed. An application of this method could be a user level task which copies data from buffers which have been stored on disk to magtape. This could be of advantage in an environment in which data comes in bursts and the disk is capable of higher data rates than magtape. This application may require careful choice of buffer sizes.

A single RSX-11M global event flag is used to signal that data acquisition has suspended execution and that newly filled data buffers might be available for user tasks. This global event flag is also used to resume user tasks that have suspending when they are unable to acquire a buffer.

The data acquisition task is the only task which is allowed to change a buffer semaphore to a negative state. The negative values usually have a specific meaning as shown in Table 2.

Table 2

Semaphore Value	Meaning
-1	Buffer proceeding to permanent storage
-2	Buffer controlled by data acquisition task
-3	Buffer allocated to buffer filler

There is a single routine within the data acquisition task which uses the buffer decsriptor block to acquire buffers for data acquisition.

The advantages of the buffer allocation scheme are:

1. Analysis type or other user tasks may be added or changed easily and at any time.
2. Buffers are allocated according to demand.
3. Data acquisition is independent of user and analysis tasks.

Preliminary Results

Some preliminary measurements have been made to determine basic performance levels. The particular choice of buffer size and number as well as event sizes was arbitrary. Three buffers of 1024 words each were used. An event consisted of five header words and three data words obtained from CAMAC. Unused portions of buffers are padded with a special 'null' event. Measurements were integrated over long periods (10 to 30 seconds) to provide a 'steady state' value where events are available in a continuous and instantly available mode. No user tasks were in operation while these measurements were taken.

Events/Buffer	Buffers/Second	Conditions
127	63	No recording
1	75	No recording
1	12	Magtape recording

The magtape system was an IBM 729 7-track unit with a KS90 controller. This is a 112.5 inches per second drive which writes at 800 bits per inch.

REFERENCES

1. S. Shlaer, An MBD Primer, Informal Report LA-55-11-MS, Los Alamos Scientific Laboratory, Los Alamos, New Mexico (1974).

2. R. F. Thomas, Jr., Programming the MBD Model II, Informal Report LA-5404-MS, Los Alamos Scientific Laboratory, Los Alamos, New Mexico (1973)

3. Original version by Richard F. Thomas, Jr., Los Alamos Scientific Laboratory, Los Alamos, New Mexico.