

Lawrence Berkeley National Laboratory

Recent Work

Title

SOME EXPERIENCES WITH PULSE HEIGHT ANALYSIS USING A DISK

Permalink

<https://escholarship.org/uc/item/0603z5z6>

Author

Jenson, David W.

Publication Date

1969-11-01

Presented at the Fall Symposium of Digital
Equipment Computer Users Society,
Las Vegas, Nevada, November 17-18, 1969

UCRL-19402
Preprint

ej. J

RECEIVED
LAWRENCE
RADIATION LABORATORY

DEC 17 1969

LIBRARY AND
DOCUMENTS SECTION

SOME EXPERIENCES WITH
PULSE HEIGHT ANALYSIS USING A DISK

David W. Jenson

November 1969

AEC Contract No. W-7405-eng-48

TWO-WEEK LOAN COPY

*This is a Library Circulating Copy
which may be borrowed for two weeks.
For a personal retention copy, call
Tech. Info. Division, Ext. 5545*

LAWRENCE RADIATION LABORATORY
UNIVERSITY of CALIFORNIA BERKELEY

UCRL-19402

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

SOME EXPERIENCES WITH PULSE HEIGHT ANALYSIS USING A DISK

David W. Jenson
Lawrence Radiation Laboratory
University of California
Berkeley, California

ABSTRACT

A partial description of two systems for pulse-height analysis that store histograms on a movable-head disk is given, with emphasis on techniques for continuously updating histograms stored on a disk and for controlling data flow through asynchronous parallel processes of differing priorities. One of the systems is capable of updating 16,384 channels of histogram on the disk at a continuous rate of 750 events per second using a PDP-5. Some projected data-handling rates for a PDP-8 and a head-per-track disk are given.

INTRODUCTION

With the use of better detectors and multiple-parameter techniques, it is becoming more common for an experimenter to require that 8 to 16 thousand single- or double-precision channels of histogram be accumulated. To provide enough random access core memory to store the entire histogram is expensive, but if the data rate is not too high, it is often feasible to accumulate the histogram on a disk. This paper discusses some of the techniques evolved during development of two pulse-height analysis systems that store the entire histogram on a movable-head disk, one using a PDP-8 and one a PDP-5. The PDP-8 system collects data onto a buffer track for 1 second out of every 6, then uses part of the 5-sec dead time to update the histograms on the disk. The PDP-5 system continuously updates the histogram on the disk simultaneously with data taking. The intent of this paper is to indicate the potential of this type of system and to provide some guidance to those who may be contemplating a similar endeavor. Some projected data-handling rates for a system using a PDP-8 to continuously update histograms on a head-per-track disk are given.

Description of the Disk

The disk in both systems is a Data Disc M6, which has a movable head with access to 130 tracks on a removable disk surface and three fixed heads on a nonremovable surface. The time for one revolution of the disk is 50 msec. The movable head steps from one track to another at a rate of 5 msec per track, with a 100-msec head-settling time after stopping. Thus the average access time to an adjacent track is 130 msec ($5 + 100 + 25$) and the maximum possible access time is 795 msec ($5 \times 129 + 100 + 50$).

Each track is formatted with 29 data sectors of 128 words each, and associated with each data sector is a one-word address sector and a one-word write-protect sector. Each address sector contains a unique 12-bit address composed of seven bits of track number (only 128 or the 130 head positions are used) and five bits of sector number (where only numbers 0 through 28 are used). The

controller, which was designed and built at Lawrence Radiation Laboratory, allows one to issue commands such as read or write the next address sector, data sector, or protect sector, and to step the head one track in either direction. It is the program's responsibility to find the desired data sector. Typically, the program issues a read-address-sector command to find where the head is, steps the head to a new track if necessary, then issues read-address-sector commands until the disk revolution has brought the desired data sector into position. It can then issue a read- or write-data command. Normally interrupts are used so the computer may carry out other tasks while the correct sector is being sought. All data transfers must start at the beginning of a data sector, but may then continue for any number of words from one up to a full track. The controller uses the three-cycle data break with the PDP-8 and the one-cycle data break with PDP-5. The data rate is one word every 13 μ sec while data is being transferred. The disk controller automatically generates and checks parity for each word.

The Display

A display controller was also built that accepts simple display commands from one or two of the fixed-head tracks on the disk. Each command consists of a 10-bit position value, one bit that indicates load X or load Y buffer, and one bit that specifies whether or not to intensify the point. Thus the commands are analogous to the standard display commands on the PDP-5/8. When one track is used, a total of 1856 random points can be displayed at a refresh rate of 20 pictures/sec, and when two tracks are used 3712 random points can be displayed at 10 pictures/sec. This display controller thus removes the need for the program to continuously refresh the display. For a new system, one would probably specify a storage scope to serve this function.

GENERAL PROGRAMMING PHILOSOPHY

In a situation in which the display does not need to be continuously regenerated and a disk is available for rapidly loading programs into core, it is reasonable to partition the system into as many independent tasks as possible, each one to be handled by a separate program loaded from the disk. Each

program may operate on common variables and experiment parameters that are kept permanently resident in core, and on the histograms on the disk. While data is being taken and the histograms updated, only the routines that are necessary for these tasks are in core. Any other functions, such as calculating a new display or responding to experimenter commands from the teletype, cause data taking to cease and other programs to be loaded in from the disk. If each function, or command, that the experimenter wishes to initiate is carried out by a separate program loaded in from the disk, a simple keyboard monitor with a program directory will serve as the "command decoder" of the system. Each program may then request further parameters via the teletype if necessary. It is also generally desirable to allow the experimenter to specify intervals at which the system will automatically interrupt data taking and recalculate the display.

Appendix A contains an example of how the program partitioning was carried out for the PDP-5 system. In the PDP-5 system each program was stored on a separate track on the disk as a direct image of a part of core. A simple permanently resident routine quickly loads in a new program when called with the track number of the desired program. One of the programs that can be loaded is a keyboard monitor with a program directory in it that allows programs to be called by name from the teletype. Each task that the experimenter may wish to initiate is then carried out by a program called by name from the teletype.

Careful partitioning of the system and use of the disk to quickly load one part of the system at a time into core eliminate the need to spend large amounts of time producing very compact code. These techniques have the additional benefit of making it easy for different persons to write different parts of the code. In connection with these points, it is sometimes distressing to note how much time can be wasted producing elegant or compact or fast code in a situation where elegant, compact, or fast code is not needed. The true measure of a programmer's usefulness is not how tightly he can code a particular routine by spending considerable time on it, but rather how well he can allocate his limited resources as a programmer to those portions of his system that really need to be short or fast. Use of the disk saves a large amount of programming time by eliminating the need to make programs short.

PDP-8 SYSTEM

Experiment Background

The first system written uses a PDP-8 to collect data for an experiment on the Bevatron, which has a 1-sec beam spill every 6 sec. The experimenters wanted to be able to handle up to 1000 events during each 1-sec beam spill. Each event has four parameters of 10 bits each. They wanted to save the "raw" data on magnetic tape for later analysis by the PDP-8 and by larger computers, and to accumulate event counts in three double-precision histograms (of 512, 1536, and 4096 channels each) on the basis of some simple calculations on the raw data (using windows and selection of bits from various parameters). Since a total of 12,288 words is required for the histograms, it is clear that they must be stored on the disk. It is also clear that the 5 sec between each beam spill can be used for updating the histograms

and recalculating the display. The intermittent arrival of data may not be typical of most experiments, but some of the methods for handling the flow of data through the system may be of general interest.

PDP-8 General Data Flow

The general data flow is as follows (refer to Figs. 1 and 2). The raw data, which may be coming in either from the ADC or from a previously recorded magnetic tape, is entered into one of the raw-data buffers. This raw-data buffer is then emptied to the magnetic tape (if the data came from the ADC) and to a "thawing" routine that does the simple operations on the raw data that are necessary to determine the histogram channels to be incremented, and puts this new information in thawed-data buffers. The thawed-data buffers are emptied to one of the fixed-head tracks on the disk, which can hold the data for a total of 1143 events. After the beam spill has ended (or after the buffer track on the disk is full, if the raw data is coming from magnetic tape), a section of the histogram on disk is brought into core, the buffer track is scanned for histogram channels in the section of histogram currently in core, those channels are incremented, the histogram section is written back onto disk, and the next section is brought into core. When all the sections of the histograms have been updated in this manner, the display is recalculated.

PDP-8 Buffers

The histogram updating is thus divided into two phases, one taking place during the 1 sec of data-taking, the other occurring after the burst of data. During the first phase there are two kinds of buffers (refer to Fig. 3) in use: raw-data buffers of 512 words and thawed-data buffers of 384 words each. Both kinds of buffers hold the appropriate form of data for 127 events, and the first word of each buffer tells how many events the buffer contains. The raw-data buffers use four consecutive words for each event to store the original 4 parameters of 10 bits each from the ADC. The thawed-data buffers have three words for each event, each word containing the channel to be incremented in each of the three histograms. The thawed-data buffer is divided into three 128-word parts, and each part contains the data relative to one of the three histograms. Thus the data words for each event are not consecutive, but 128 locations apart, and every third 128-word sector on the buffer track contains the data for a given histogram. There are several buffers of each kind, and there is a raw-data buffer list and a thawed-data buffer list. The entries in the lists point to the word before the buffer, and this word is used to hold status information about the buffer (refer to Appendix B).

During the second phase, when the histograms are updated, there is one 2048-word buffer to hold a section of the histogram from the disk and a pool of 128-word buffers to hold sections from the buffer track on the disk. The buffers for the second phase can obviously overlay the buffers for the first phase.

PDP-8 First Phase and Program Control

During the data-taking phase several processes are proceeding asynchronously and in parallel to fill and empty buffers. The ADC process is filling a raw-data buffer. The magnetic tape process and the

thawing process are emptying a raw-data buffer. The thawing process is filling a thawed-data buffer, and the disk process is emptying a thawed-data buffer to the buffer track on disk. Each process consists of a driver subroutine and a main module. When the driver subroutine is called it initiates the process and sets up the main module of the process to continue as time or the condition of I/O devices warrants. The driver subroutine then returns without completing the task. The task is completed under control of the main module. For example, the thawing-process driver sets up pointers to the appropriate buffers, changes the statuses of the buffers, and sets a software flag indicating that thawing is in progress, then immediately returns. The main idle loop of the system, upon detecting this flag, jumps into the thawing main module. For processes which are I/O device handlers, the procedure is slightly different. For example, the disk-handler driver subroutine sets up pointers to the thawed-data buffer to be written, determines the disk address at which it is to be written, sets the buffer status to "emptying to disk," issues a read-address-sector command to the disk controller, and sets up the interrupt routine to return to the appropriate place in the main module of the disk handler when an interrupt arrives from the disk. Thereafter the main module of the disk handler is called by the interrupt routine whenever action is needed, until the process of writing the buffer onto the disk is complete.

All processes are activated (by calling their driver subroutines) by a "check status" routine. This routine has a section for every process. Each section checks the statuses of the buffers and the process to see if the process is now inactive and can be activated. For instance, one section of the status-check routine tests to see if the ADC handler process is now inactive and if there is an empty raw-data buffer. If both these conditions are satisfied, it starts the ADC handler process by calling its driver subroutine. As another example, there is a section to determine if there is a full raw-data buffer, an empty thawed-data buffer, and an inactive thawing routine. If all these conditions are satisfied the thawing process driver subroutine is called.

Whenever a process completes its task, it sets the status of any buffers it has used to the appropriate value, sets its own status to inactive, sets a check status flag, and "dies" (the thawing process simply returns to the main idle loop; the I/O handler processes return to the interrupt routine without initiating any I/O activity that would cause a later interrupt). The check-status flag is really a software interrupt that tells the system to make a pass through the check-status routine, whenever it is convenient, to see if any processes may now be activated. Thus the completion of one task does not directly initiate another task, but rather it causes various status bits to be set, reflecting the new state of the data and the system. Processes are then initiated when the necessary conditions for their operation are satisfied.

This technique has several advantages. It makes it easy to use pools of buffers and change the number of buffers. It provides a good way to control all the processes that are operating in parallel and that are starting and stopping asynchronously. It makes it easy to handle the case in which more than one process is emptying a buffer at the same time. And finally, it makes a modular system in which it is

easy to introduce new processes and new paths of data flow without having to worry about detailed timing considerations.

PDP-8 Second Phase

During the second phase the data on the buffer track on the disk is used to update the histograms on the disk. Each histogram on the disk is divided into sections of a maximum of 2048 words each, and each section is stored on a separate track. Thus one section is required for the 512-channel histogram, two sections for the 1536-channel histogram (all histograms are double precision), and four sections for the 4096-channel histogram. To begin the updating process, one of these sections is read into core. Then the program begins reading in the 128-word sectors of the buffer track that correspond to the histogram whose section is being updated. At the same time another process scans the data in these 128-word buffers and increments the indicated channels in the section of histogram in core. Since only every third sector must be read for any given histogram, and a pool of several 128-word buffers is maintained, it is usually possible to complete the updating of the histogram section in core in one revolution of the disk. Since there is a total of seven sections of histograms to be updated, and it takes approximately 280 msec per section (130 msec head-step and latency time, 50 msec read section, 50 msec read buffer track, 50 msec write section), it takes approximately 2 sec to update the entire 6,144 channels of histograms.

PDP-5 SYSTEM

PDP-5 Experiment Background

The second system uses a PDP-5 to collect data from an experiment on the 184-inch synchrocyclotron. In this experiment the data arrives continuously (on the time scale of the disk-revolution time of 50 msec). Each event produces a 12-bit number from an ADC and two routing bits. The experimenters wanted to accumulate 4×4096 single-precision channels of histogram. The goal was to design a system that could continuously update the histogram on the disk simultaneously with data-taking. It was assumed that data taking would have to be temporarily interrupted whenever it was desired to calculate a new display.

PDP-5 Data Flow

The basic method for updating the histogram on disk is similar to that used in the PDP-8 system. The entire histogram is divided into sections, in this case 16 sections of 1024 channels each. As each 14-bit datum arrives (refer to Fig. 4) it is divided into a four-bit and a 10-bit part. The four-bit part is used to select one of the 16 first-in-last-out "hash" buffers that correspond to the 16 sections of histogram. The 10-bit part indicates which channel in the section is eventually to be incremented, and is entered into the selected hash buffer. While the incoming data is being sorted into these buffers, the histogram on the disk is being brought into core for updating one section at a time. Each section in turn is read into core. The 10-bit numbers in the hash buffer corresponding to this histogram section are taken out of the buffer and used to indicate which channels to increment, and the histogram section is written onto the disk again.

If a hash buffer becomes full before the system finishes updating all the other sections and gets back to this one, the data taking must be turned off. If the events are evenly distributed over all the histogram sections, the maximum data rate is thus a function of the total number of words devoted to the hash buffers and the total amount of time needed to complete one full scan of all the sections of the histogram. Unfortunately, histograms of uniform height are not of much interest. In order to more uniformly distribute the events over all the buffers, the four bits that indicate the histogram section are taken from the least significant bits of the 14-bit datum. Thus each of the 16 histogram sections on the disk actually contains every sixteenth channel of the "real" histogram. This technique seems to do a good job of randomizing, or "hashing," the distribution of events among the 16 sections. The disadvantage of this technique is that the histogram on the disk must be sorted every time the real histogram is needed, such as whenever the display is recalculated. This sorting process, however, is quite simple and takes about 2.5 sec for every 2048 channels sorted.

In this particular system an additional complication existed because timing considerations (one word every 13 μ sec from the disk, 6 μ sec core cycle time on the PDP-5) dictated that while data was being transferred to or from the disk, the program could not execute any indirect memory references, any JMP, or any JMS instructions (one can, however, execute an effective jump by depositing an address in core location 0, the program counter on the PDP-5). For this reason a buffer with a capacity of 50 events was added to accept data directly from the ADC prior to sorting it to the hash buffers. In this way the sorting operation could be avoided during disk data transfers.

PDP-5 Program Flow

The program is initiated by issuing a command (to the disk-handling routine, which uses the interrupt) to read the first section of the histogram from the disk. When the section has been brought into core a flag is set indicating which section is available and the command to write the section out again is given. Just before starting to write the section out the disk routine clears the "histogram section available" flag. Completion of the write operation causes a command to be issued to read the next section. Thus the histogram is continuously scanned under control of the interrupt routines. Normally the ADC is handled by the interrupt routine, with each event placing two words in the ADC buffer. Whenever a data transfer to or from the disk is taking place there is the previously mentioned restriction on the instructions that may be executed, so the interrupt is left off for the duration of the transfer and a special routine is executed that checks for ADC data and puts it in the ADC buffer by means of instruction modification (to avoid indirect memory reference). The main "interrupt-on" loop checks first to see if there is a histogram section available in core and if the appropriate hash buffer has any data in it. If so, the process of updating the histogram section from the data in the hash buffer is continued until either the buffer is empty or the histogram section is no longer in core. After completing all histogram updating that can be done, the main interrupt-on loop then checks to see if there are any events in the ADC buffer to be sorted into the histogram section buffers.

PDP-5 Program Control

The technique for controlling the flow of the program is similar to that in the PDP-8 system. In a system in which many different processes proceed in parallel, it is difficult, if not impossible, to have the completion of one operation directly start the operation of another. Rather, the communication between processes is handled by flags and status words, most of them indicating the status of various buffers. The priorities of the processes are then established, in part, by the order in which the flags and statuses are scanned for jobs to be done. As an example, it would be very difficult to have the interrupt-off disk routine directly start the interrupt-on histogram updating routine when a section has just been read into core, since the interrupt-on processor may have been interrupted in the middle of something. Instead, a word is set indicating that a histogram section is available, and the interrupt-on processor can then initiate the updating when it is convenient.

PDP-5 Timing

As previously mentioned, if the events are evenly distributed over all the histogram sections, one limitation on the maximum data rate is imposed by the number of events that can be stored in the hash buffers and the time it takes to make one complete scan of the histogram on the disk. In this system each track on the disk has two of the 16 sections. To read and then write both the sections on a track takes three revolutions of the disk, which, with the head-stepping time to the next track, adds up to a nominal 280 msec for every two sections. To scan all 16 sections and step the head back to the first section would then take about 2.3 sec. Since there is a total of 2048 words of hash buffers, the maximum data rate would nominally be 890 events per second (2048 divided by 2.3). It turned out that the disk used in this system had been adjusted to have a head-settling time of slightly more than 100 ms, and this, together with the effect of random fluctuations in the distribution of events in the hash buffers, gave rise to an observed maximum rate of 750 events per second.

The other factor that might limit the maximum data rate is the speed of the computer in the internal handling of the data. This is particularly a problem on the PDP-5, which has a 6- μ sec memory cycle time and an extra memory cycle for each instruction to fetch the program counter, making it only about one sixth as fast as the PDP-8. It was found that the most critical area was the routine to fetch a word out of a hash buffer and increment the indicated channel in the histogram section in core. In this system, there is a 34-msec interval after the histogram section is completely in core and before it is written out again. To keep up the maximum data rate, the hash buffer must be completely emptied during this time. After some attention to this area, it was found that the computer speed was just adequate at 750 events/sec.

PROJECTIONS

Single Precision

On the basis of this experience with a system using a PDP-5 to continuously update histograms on a movable-head disk it is possible to make some

projections on what might be achievable with a PDP-8, a PDP-8/I, or a PDP-8/L and a head-per-track disk. If one adopted the same basic data organization, with 16 sections of histogram of 1024 channels each and a total of 2048 words of hash buffers, the limiting factor would be the time it takes to scan the 16 sections on the disk. Using, for example, a DF-32 disk, it would take 66 msec to read or write a 1024-word histogram section. Thus if one allows one revolution (33 msec) after reading and before writing to update the section in core, the time per section will be 165 msec or 2.64 sec for the entire histogram. This assumes that the disk addresses of the sections are staggered so as to avoid any read-latency time after writing the previous section. Thus the maximum data rate would be around 750 events/sec. Although the DF-32 does not require head-stepping time, this advantage is counteracted by the much higher data-transfer rate of the Data Disc disk. With a RF-08 disk, which is faster as well as larger than the DF-32, the time to transfer one section would be 16 msec, with 16 msec time between reading and writing. In this case the time to scan the histogram would be 0.8 sec, implying a data rate of around 2500 events/sec. If one were to use a computer with 8 K of memory, and to use the extra 4 K for increasing the size of hash buffers, the expected maximum data rates would be 2300 events/sec for the DF-32 and 7500 events/sec for the RF-08. By developing a scheme to overlap the reading or writing of one section with updating of another (i.e., double buffering the histogram section transfer process) one might even be able to achieve a rate of 10,000 events/sec. Such a scheme would probably require rewriting histogram sections in a different place from where they had just been read, in order to eliminate latency time and to reduce the time between successive transfers to only the time, involved in processing the disk and ADC interrupts. At the highest data rate, one may be approaching a limit due to the speed of the computer in internal handling of the data. It is probable that at least 7500, and very possibly 10,000, events/sec is attainable, based on experience with the PDP-5 system (taking into account the approximate factor-of-six difference in the speed of the machines and the considerable extra work and lost time due to the restrictions imposed on instructions executed during disk data transfers). More channels could be handled at proportionally lower data rates. Thus with a RF-08 and an 8 K memory one could easily accumulate in a 64-K channel histogram at a rate of 1800 events/sec.

Double Precision

Although the PDP-5 system as written does not have double-precision channels in the histogram, it does not appear that there would be much reduction in the maximum data rate if this provision were included. The high-order words for each channel would be stored on a different area of the disk from the low-order words. Thus the histogram sections brought into core in the course of normal scanning of the histogram would contain only the low-order part. An entry would be made in another buffer for every channel whose low-order part overflowed. When this buffer is full, or on a semicontinuous basis, the high-order parts of the appropriate channels could be incremented. With 12 bits for the low-order part, the rate at which high-order parts need to be incremented is on the order of only one channel per second.

CONCLUSION

A PDP-8 system was developed that can handle intermittent bursts of 1150 events in one second, with two additional seconds required to update the 6144 double-precision channels of histogram on the disk. If a second buffer track were used, the system would be capable of accepting 2300 events during a 1-sec period and then updating the histograms on disk in 4 sec. A PDP-5 system was developed that can update 16,384 channels of histogram on the disk at a continuous rate of 750 events/sec. A similarly organized system using a PDP-8 with 8 K of memory and a RF-08 head-per-track disk could easily update the same number of channels at a rate of 7500 events/sec, and possibly at a rate of 10,000 events/sec.

APPENDIX

A. Commands (Partitions) in PDP-5 System

- SET: Allows experimenter to set the value of various parameters, such as type and scales of display, gain stabilization anchor point, display recalculation interval.
- STORE: Starts data-taking process. Automatically interrupted at intervals specified by the experimenter to recalculate display. Return to monitor indicated by "CTRL D."
- LOOK: Recalculate display.
- ZERO: Zero out the histogram on disk.
- DUMP: Provide decimal dump of indicated portions of histogram on the teletype.
- SUM: Sum the indicated channels.
- DRAW: Plot histogram on X-Y plotter.
- TAPE: Dump histogram onto magnetic tape.

B. Buffer Status Bits in PDP-8 System

Raw-Data Buffers

1. Empty.
2. Filling from ADC
3. Ready to empty to magnetic tape.
4. Ready to empty to "thawing"
5. Emptying to magnetic tape.
6. Emptying to "thawing."

Thawed-Data Buffers

1. Empty.
2. Filling.
3. Full.
4. Emptying.

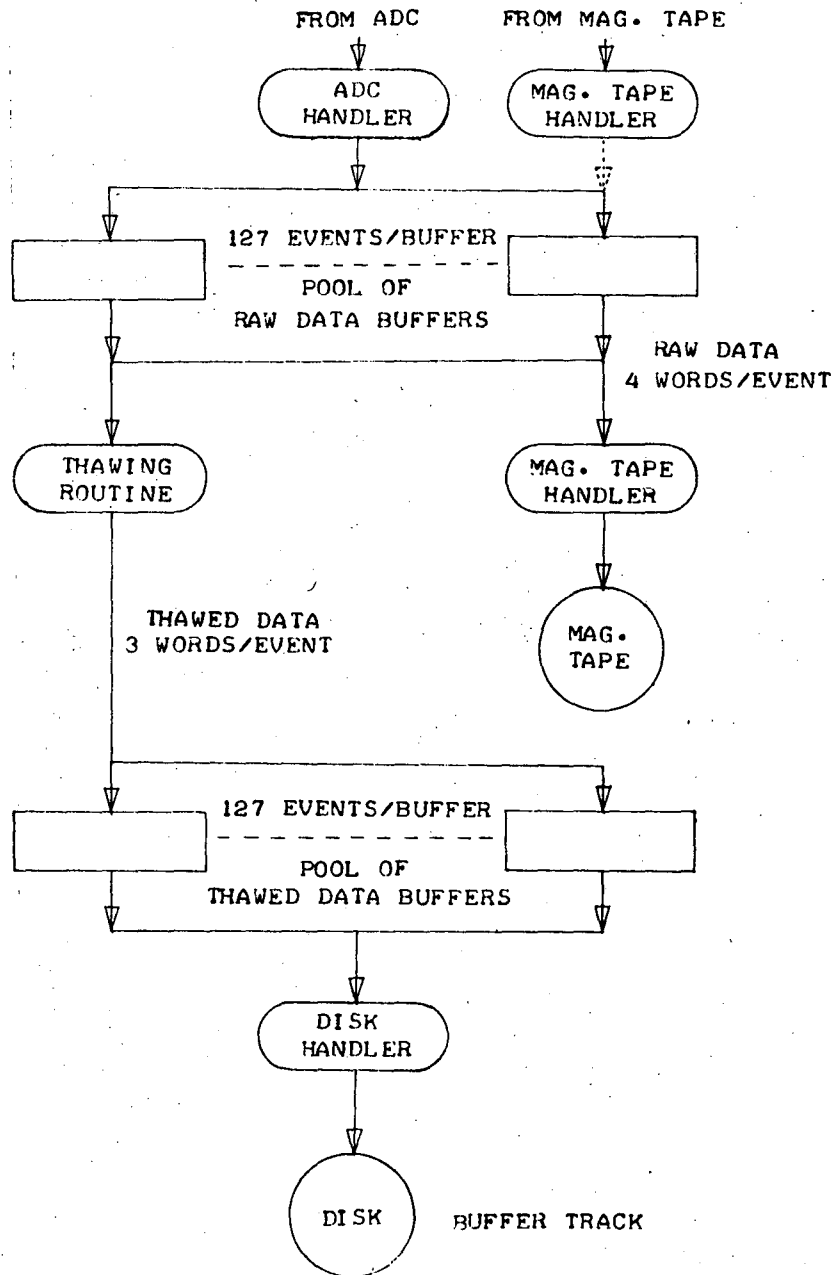


FIG. 1: PDP-8 DATA-TAKING

XBL 6911-6536

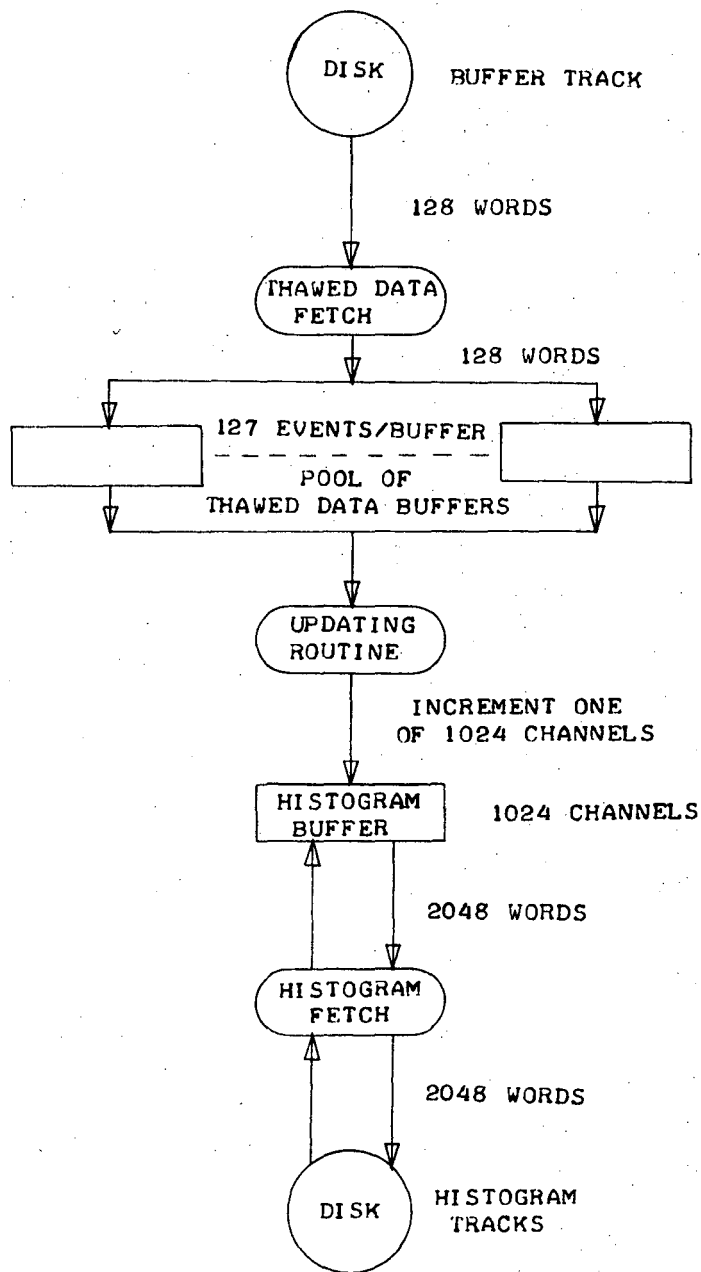
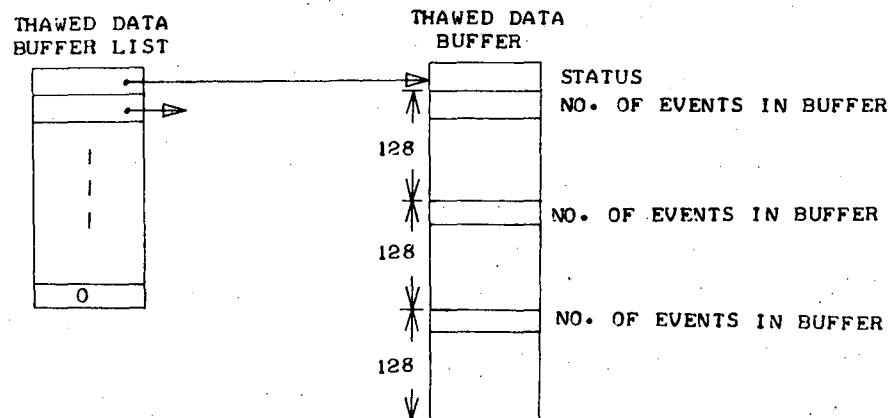
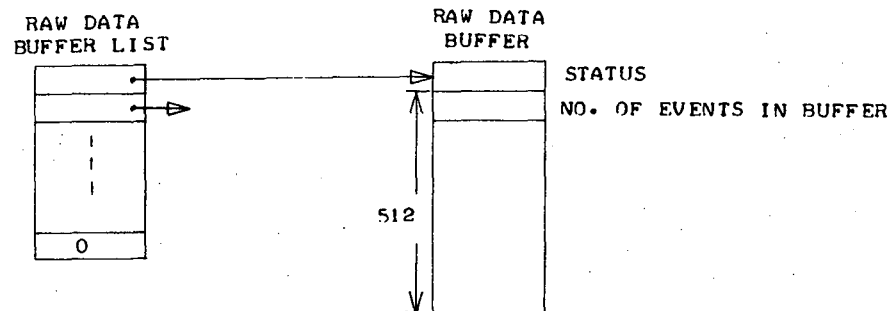
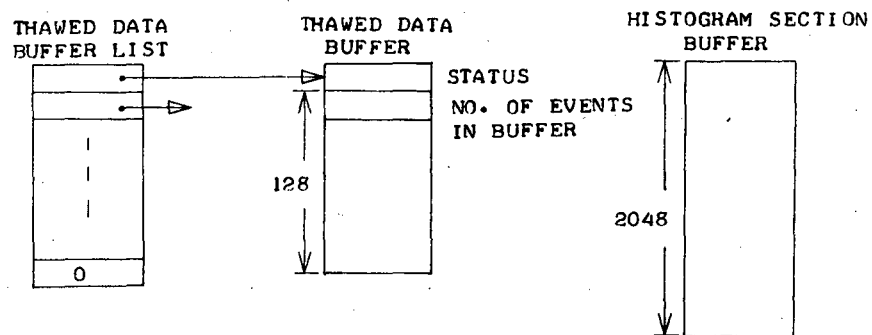


FIG. 2: PDP-8 HISTOGRAM UPDATING

XBL 6911-6534



DURING DATA-TAKING



DURING HISTOGRAM UPDATING

FIG. 3: PDP-8 BUFFERS

XBL 6911-6535

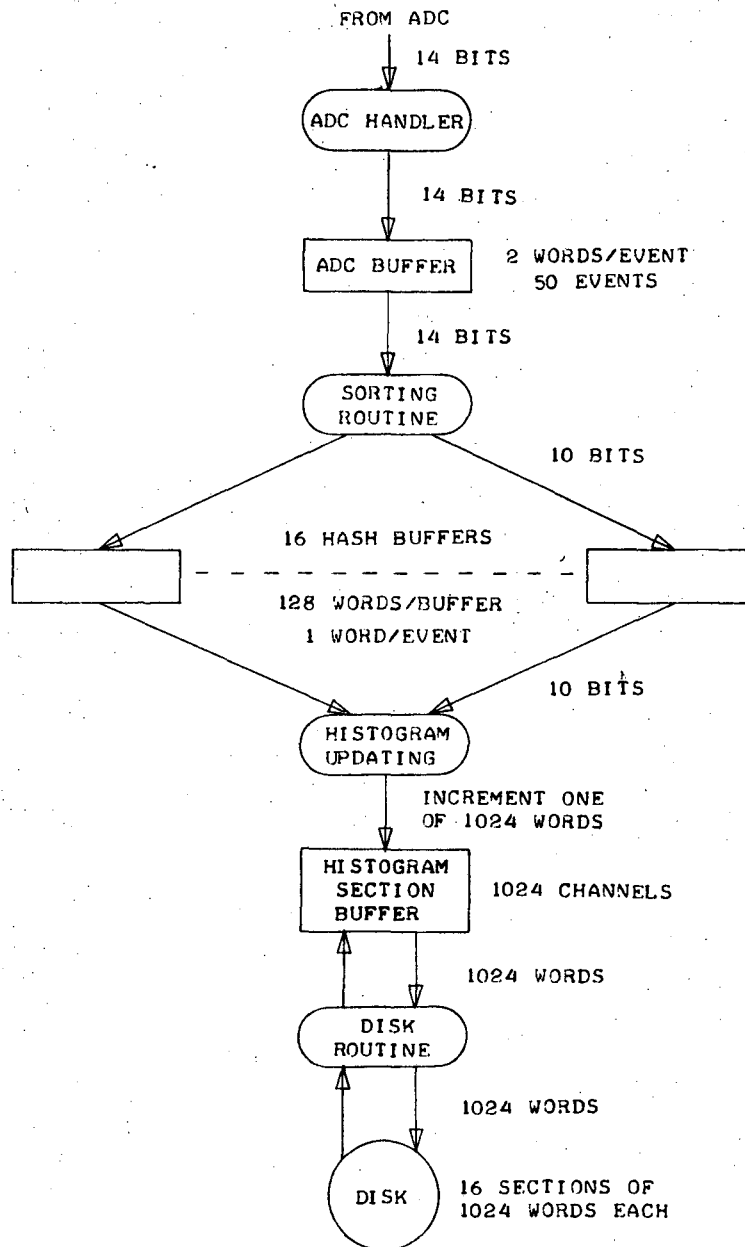


FIG. 4: PDP-5 DATA FLOW

XBL 6911-6533

LEGAL NOTICE

This report was prepared as an account of Government sponsored work. Neither the United States, nor the Commission, nor any person acting on behalf of the Commission:

- A. Makes any warranty or representation, expressed or implied, with respect to the accuracy, completeness, or usefulness of the information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or*
- B. Assumes any liabilities with respect to the use of, or for damages resulting from the use of any information, apparatus, method, or process disclosed in this report.*

As used in the above, "person acting on behalf of the Commission" includes any employee or contractor of the Commission, or employee of such contractor, to the extent that such employee or contractor of the Commission, or employee of such contractor prepares, disseminates, or provides access to, any information pursuant to his employment or contract with the Commission, or his employment with such contractor.

TECHNICAL INFORMATION DIVISION
LAWRENCE RADIATION LABORATORY
UNIVERSITY OF CALIFORNIA
BERKELEY, CALIFORNIA 94720