

MAX-PLANCK-INSTITUT FÜR PLASMAPHYSIK

GARCHING BEI MÜNCHEN

The Institut fuer Plasmaphysik
Data Acquisition System

DAS

Concepts and Facilities

Robert F. Lathe
Erich Mueller

IPP R/10

May 1975

Die nachstehende Arbeit wurde im Rahmen des Vertrages zwischen dem Max-Planck-Institut für Plasmaphysik und der Europäischen Atomgemeinschaft über die Zusammenarbeit auf dem Gebiete der Plasmaphysik durchgeführt.

IPP R/10

R. Lathe

The Institut fuer Plasmaphysik
Data Acquisition System DAS
Concepts and Facilities

May 1975

Preface

This report describes the basic design concepts and facilities of the general data acquisition system (DAS Release 1) developed by the Data Acquisition Project at the Institut fuer Plasmaphysik. The report is divided into three chapters as described in the following paragraphs.

Chapter 1 contains information on the general philosophy of DAS, the user requirements and limitations of Release 1.

Chapter 2 discusses the major software design concepts. Special emphasis is given to the data acquisition task manager and the DAS system structure. The major standard facilities, system and data files are described.

Chapter 3 is concerned with the DAS hardware environment. The minimum computer configuration and supported extensions are described. Particular attention is given to user device hardware interfacing requirements.

Notation Conventions

- Brackets [] Brackets are used to enclose optional elements, e.g. Y[ES], indicating that the letters ES may be omitted at the user's discretion.
- Square □ The character □ is used to denote a space, as in "file□name", which means that "file" and "name" must be separated by a blank.
- End of line <CR> The <CR> symbol indicates that the Carriage Return key is to be pressed. This is interpreted as the end of an input line from the console.
- Braces { } Braces are used to enclose two or more elements from which a choice must be made. For example:
(YES)
< >
(NO)
means that either the text string YES or the string NO must appear in the indicated position.
- Octal value (8) The symbol (8) indicates that the given value is in octal (base 8) notation. For example, 100(8) is the decimal value 64.

Table of Contents

Preface

Notation Conventions

1.0: General Information

- 1.1: Introduction
- 1.2: General Philosophy
- 1.3: General Requirements
- 1.4: General Limitations
 - 1.4.1: Single User System
 - 1.4.2: Multiple Tasking Facilities

2.0: The LAS Software Environment

- 2.1: Design Concepts
- 2.2: Function Groups
 - 2.2.1: Data Acquisition and Data File Management
 - 2.2.2: Control Systems
 - 2.2.3: Data Processing
- 2.3: System Files
 - 2.3.1: System Definition File DASSDF
 - 2.3.2: Communication Vector File DASCVT
 - 2.3.3: Operator's Console Log File DASLOG
- 2.4: Data Files
- 2.5: User Device Descriptor and Control Blocks
- 2.6: Memory Management
 - 2.6.1: Memory Layout
 - 2.6.2: Interrupt Vector Area
 - 2.6.3: DOS Supervisor Area
 - 2.6.4: DOS I/O Handler and Buffer Area
 - 2.6.5: Stack Area
 - 2.6.6: DAS Supervisor Area
 - 2.6.7: DAS Overlay Area

3.0: The DAS Hardware Environment

- 3.1: Minimum Computer Configuration
 - 3.1.1: Central Processor
 - 3.1.2: Operator's Console
 - 3.1.3: System Disk
 - 3.1.4: Line Printer
- 3.2: Additional Computer Peripherals
- 3.3: User Hardware
 - 3.3.1: Standard Interrupt Based Devices
 - 3.3.2: CAMAC Based Devices
 - 3.3.3: Direct Memory Access Based Devices

References

1.0 General Information

1.1 Introduction

The primary functions of a data acquisition system for use in experimental physics laboratories are:

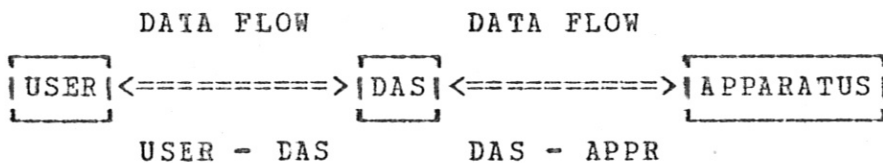
To organize a collection of (possibly over a period of time varying) hardware into a consistent whole and to control the hardware so that it may be used efficiently in the carrying-out of experiments.

To assist experimentalists to make the best use of an experimental apparatus by providing the means of analyzing the results of past experiments which may then be used as a basis for determining the most effective way of achieving the goals for which the experiment is designed.

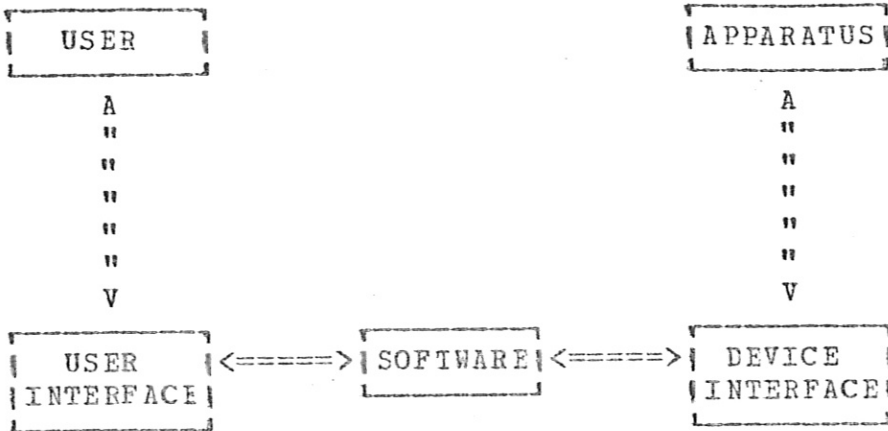
This chapter contains information on the general philosophy of DAS, the design constraints and general restrictions of the first release.

1.2 General Philosophy

In contrast to a general purpose computer system, a data acquisition system is primarily a means of communication between the user and the experimental apparatus (see figure below). The hardware resources encompass not only the computer and its associated peripheral devices but also interfaces to and often parts of the experimental apparatus. The speed of the system is measured relative to response times to demands imposed either by the user on the one side or by the apparatus on the other. Of primary concern to the user, particularly the non-computer oriented user, is the simplicity of the man-machine interface. The major conflicts arise in attempting to maximize resource utilization and user interface simplicity while concurrently obtaining high speed responses.



The diagram above depicts a data acquisition system as a bi-directional data path between user and apparatus. Additionally, a data acquisition system is a data translator, transforming events occurring in the apparatus into terms meaningful to the experimentalist and, conversely, demands from the user into signals acceptable to the apparatus. The translation process occurs in the steps shown in the diagram below:



The user interface is commonly referred to as the user or Operator's Console. The data pathway shown above forms the basis of a feedback control loop. In many instances, the operator receives information from the apparatus via the system, makes a decision and then gives a command which the system returns to the apparatus. Often, the path may be short-circuited in the software section so that events occurring in the apparatus are interpreted and a programmed response is returned to the apparatus. In such cases, the operator is notified that the action has taken place but is not required to be directly involved in the decision making process.

1.3 General Requirements

During the system planning phase, a number of system requirements were identified which then determined the basic DAS structure and which formed the basis of the system concept. The identified requirements are discussed in the paragraphs below.

At the outset of the project, it was decided that each experiment would have a local minicomputer for data acquisition and experiment control. Thus a major requirement is that the system be portable. The restrictions imposed by a machine independent system were lessened somewhat with the decision to use members of a compatible family (PDP-11) for the local computer installations. Therefore the implementation should take advantage of hardware features but at the same time must have provisions for non- or upwards-compatible features. This is accomplished through the means of conditionally assembled procedures.

Associated with the requirement that the system be portable is the requirement that the system also be flexible. The number and type of user devices for data acquisition and machine control varies from experiment to experiment and with time. Therefore, a means needs to be provided to allow the connection of user devices and their associated software quickly and easily. The problem was solved through the use of DCB lists (see sections 2.1 and 2.5) for user devices and Function Group tables for user data processing routines.

In line with the general philosophy discussed in section 1.2, the user interface must be simple and remain constant. In general, the user has little or no knowledge of computer systems which suggests the implementation of a conversational type of system. In DAS, all system messages and system input requests are simple English text strings. Operator replies are non-formatted and may be

abbreviated to any understandable form. Messages which concern user devices have standard text strings into which the device identifier contained in the DCB is inserted. The user need learn only a minimum about one system and can then operate on any installation.

Not every experiment can afford a large local configuration, therefore the system must be operable on relatively small configurations and still be expandable to support large installations. This requirement is partially fulfilled by the use of conditionally assembled procedures and partially by the dynamic determination of the host machine configuration. The system thus supports large installations and disables features or substitutes devices on installations which do not contain all supported devices.

In order to implement the system in the shortest period of time, it was decided that DAS should run as a sub-system under an already available operating system, namely PDP-11 DOS. This decision imposes certain limitations upon the multiple user and multiple tasking facilities which could be implemented as described in sections 1.4.1 and 1.4.2.

Finally, it is required that the DAS be a closed system. That means that the user need not use utility routines or other programs other than those incorporated into DAS for data processing or data file management (see section 2.2.1).

1.4 General Limitations

1.4.1 Single User System

A data acquisition system may be classified as either a single or multiple user system. In single user systems, there exists only one Operator's Console and only one operator may give commands to the system. In multiple user systems, more than one Operator's Console with associated operator are possible. In this case, two or more operators may give commands to the system (semi-) independently.

The first release of DAS is implemented as a single user system, however, the system may be extended to support multiple users. The single user implementation implies that only one operator may command DAS to start or stop activities via the Operator's Console, however, more than one user terminal may be attached for purposes of data display and interaction with user programs. The advantage of a single user implementation lies in its simplicity, the disadvantage in the requirement for a higher degree of coordination between the various users, if more than one user exists. The first release of DAS is best suited for small applications and less well suited for larger applications.

1.4.2 Multiple Tasking Facilities

In the first release of DAS, multiple tasking is limited to two basic activities, data acquisition and process control. Process control tasks are executable at any time, whereas data acquisition tasks are executable only when the data acquisition Function is in control of the system. The primary reason for this restriction lies in the provision of internal buffering of acquired data which is held in core and not buffered to secondary storage. This provision was necessitated by system history and is not a basic

conceptual limitation of DAS. The multi-tasking of other Functions is limited in release 1 of DAS by the simplified core management facilities implemented in the single user concept.

2.0 The DAS Software Environment

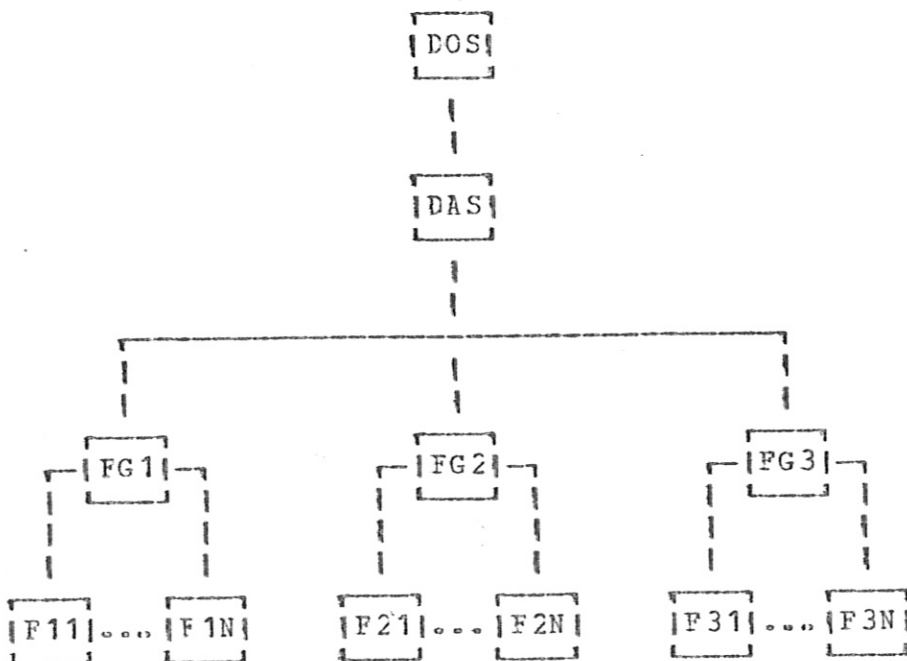
2.1 Design Concepts

As in the design of any data acquisition system, the user requirements form the basis of the system. The requirements discussed in section 1.3 are briefly summarized by the following points:

- 1) the system must be portable and flexible
- 2) the system must support small and large installations
- 3) the system must run under an existing operating system
- 4) the system must be closed
- 5) the system must be easy for the user to operate.

An additional consideration is that the experiments from which data is to be gathered do not deliver data continuously, but rather in bursts with relatively long periods of time between data acquisition phases.

The formal representation of the system as a tree structure allows an implementation consistent with the imposed constraints. The resulting tree structure imposed may be graphically displayed as in the figure below. At the root of the tree is the available operating system followed on the next level by the DAS resident portion. All functions to be performed by the system fall into three categories called Function Groups. They form the third level of the tree structure and are described in section 2.2. The last level of the tree represents the system Functions. As in any tree structure, the flow of logic must follow the paths between the tree nodes, thus nodes on the same level may not directly communicate with one another.



The conceptual structure of DAS conforms with the capabilities of the available linker which produces a tree-structured overlay system. Furthermore, the structure allows the resident portion of the system for small installations to be contained in only 16K bytes of storage (see section 2.6). It is evident that due to its formal structure, DAS meets the constraints 1, 2 and 3 above. A tree structure also allows a large number of functions to be implemented so that requirement 4 above could be met.

The major task of DAS is to gather data from the attached experiment. The requirements with regard to the number of data producing devices, amount of data, and transfer rates vary from experiment to experiment and also with time, as new devices are added to or old devices are removed from the system. The DAS concept must therefore provide for these physical differences and fluctuations. This is accomplished via a data acquisition task manager which utilizes DAS Device Descriptor and Control Blocks (DCB's). The contents of the DCB's are described in detail in section 2.5.

In order to integrate a user device into DAS, a DCB must be provided at system generation time. The description and location of the device is contained in the DCB along with device status indicators. The attachment of a user device to DAS implies at the same time the inclusion of up to two routines, namely:

- 1) user dialog routine
- 2) device I/O routine.

The user dialog routine provides a means by which the user may input device specific parameters which are subsequently available to the user data processing programs (see section 2.5). In the event that no user parameters are required, the dialog routine may be omitted. The device I/O routine is used to initialize the specified device and perform I/O either on a programed or interrupt basis (see section 3.3). Data may be either internally stored in the computer or externally stored in the user device before final storage in a data file. In order to minimize the amount of data gathered, the user devices may be logically attached or detached dynamically. Only devices which are logically attached provide data to be stored in a data file.

The data acquisition task manager scans the DCB list to determine the devices attached, the amounts of and types of data to be gathered. Prior to the data acquisition phase, the task manager initializes all logically attached devices and interrupt handling routines. Upon receipt of an end-of-experiment signal, the original data are stored in a unique data file in the format described in section 2.4. The data acquisition task manager provides a highly flexible means for implementing a central task of DAS and allows the connection of virtually all types of user devices.

2.2 Function Groups

The general activities of DAS fall into three general categories or "Function Groups":

- 1) Data Acquisition
- 2) Control Systems
- 3) Data Processing

The services provided in Function Group 1 (Data Acquisition) are an integral part of the basic DAS system. These consist of data acquisition and data file handling facilities. The services provided in Function Groups 2 (Control Systems) and 3 (Data Processing) are dependent upon the individual systems and thus are only outlined here. All DAS activities are noted in a system log file which is printed at the end of an operating session and may be stored for future reference.

2.2.1 Data Acquisition and Data File Management

The Data Acquisition Function Group allows the operator to enable the system for data acquisition, input the data from the various logically connected devices and store the original data along with certain system information into a standard format data file. In addition, the operator may dynamically re-configure the logical system between experiments and print a summary of the current system logical configuration. Devices may be logically connected and disconnected to and from the system dynamically so that only those devices which are pertinent to a given experiment or series of experiments are enabled for data acquisition. The size of the data files is minimized so that a maximum number of files may be stored on a given storage medium.

A major concern of users is the management of data files and therefore major emphasis has been placed on the solution of this problem. Facilities are available to create, copy, and delete data files as well as simplified means for reclaiming the acquired data for processing. Means are also provided for searching for files and determining the amount of available space on a given storage device. All references to data files are made via shot number (and possibly the user device code) so that the user need not be concerned with the exact file name or physical format of the data file. An important point is that only original (i.e. not pre-processed) data is stored on the data files so that the original data is always available for new processing.

System files are not directly accessible by users but, under certain circumstances, may be read or altered via facilities provided by DAS. Facilities are also provided to the installation manager for the generation and maintenance of system files.

2.2.2 Control Systems

The second Function Group, "Control Systems", is foreseen as a means for implementing feedback control systems in DAS. Typical functions would be the initialization of control systems, and attaching and detaching asynchronously operating feedback loops. In asynchronous functions, the control system is logically attached by the operator and then runs independently of operator intervention and Function Groups 1 and 3. The function normally detaches itself automatically upon completion or upon the occurrence of non-recoverable error conditions. In the current version of DAS the routines required for asynchronous control systems must be core resident whereas synchronous routines may reside on the overlay medium. The system is limited therefore in the number of attachable asynchronous control tasks, the limit depending upon the size of the routines required and the amount of core storage available.

2.2.3 Data Processing

Function Group 3, "Data Processing", is available for implementing user data processing programs. DAS provides special FORTRAN compatible routines for reading original experimental data in user programs and a set of plotting routines for the Operator's Console and plotter (if available). Additional routines are available for entering messages into the system LOG file and for format-free input of data from the Operator's Console. User programs are usually written in FORTRAN and thus may be written and tested on the IBM system. (Routines are available for reading data files from tape on the IBM system). The user is free to concentrate on solving the problems associated with processing the data and need not be concerned with I/O handling. Additionally, processing programs may be developed and tested off-line and then be included in DAS.

2.3 System Files

The DAS system utilizes three DAS system disk files, the contents of which are only indirectly available to the user. The purpose and contents of these system files are described in the paragraphs below.

2.3.1 System Definition File DASSDF

The System Definition file is used as an indicator of the location and name of the next data file to be created. DASSDF is updated when a new data file is created or when the location of the data file storage medium is changed. The contents of DASSDF are also stored in memory for fast access during the operating session. The contents of the file is an ASCII text string which has the format DDU:VNNNNN.EEE where:

DD	A two character device identifier which names the device on which the next data file is to be stored.
U	A single digit (0-7) which identifies the data file storage device unit number.
:	A mandatory separator symbol.
V	A single character data file structure indicator which identifies the file structure to be used in creating the next data file.
NNNNN	A five decimal digit identifier which corresponds to the next experiment number.
.	A mandatory separator symbol.
EEE	A three character experiment name identifier.

The DASSDF file consists of one 80 byte record and as such contains blanks or nulls in the remaining character positions.

2.3.2 Communications Vector Table File DASCVT

The file DASCVT contains a copy of the DAS Communications Vector Table (see section 2.5) which is updated each time table entries are changed. The purpose of the DASCVT file is to enable the user to begin a new operating session with the same logical system configuration as that of the last session, although the option of resetting the logical configuration is also available.

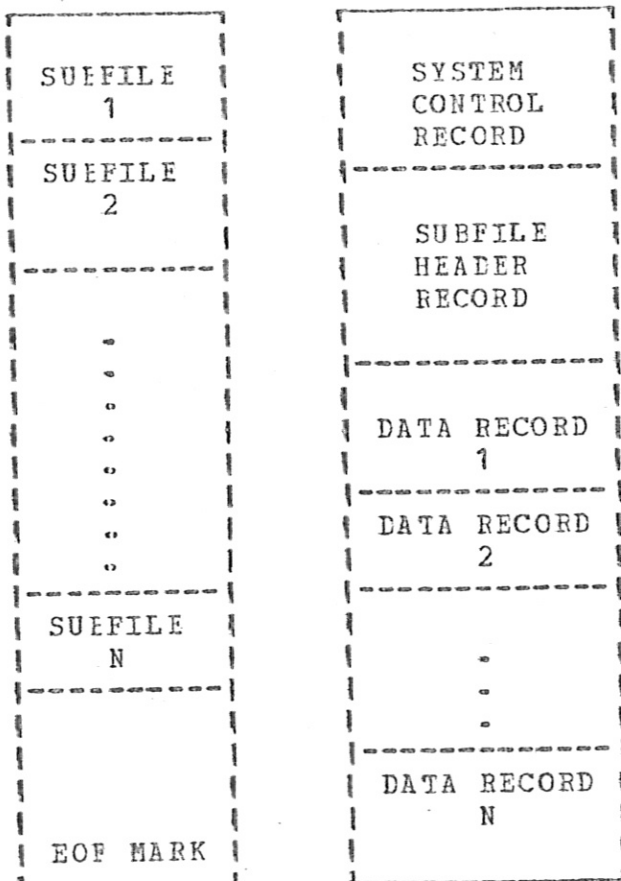
2.3.3 Operator's Console Log File DASLOG

The Operator's Console log file, DASLOG, contains a copy of all DAS system messages output on the Operator's Console during an operating session. In particular, each system function identifies itself upon selection by the operator. As described in Section 3.1.2, a basic component of the DAS hardware is a display screen used as a portion of the Operator's Console. The DASLOG provides a convenient means of saving all important system messages which, due to the limited capacity of the operator's display, would otherwise be lost. The contents of DASLOG are automatically printed on the system line printer (or equivalent) upon termination of a session so that a hard copy of the sequence of functions performed during the session is available.

2.4 Data Files

The original data resulting from each experiment are stored as a separate data file with a file name associated with the running experiment number. Data files are composed of one or more subfiles, wherein each subfile is associated with a particular user device. All data records are written with a fixed record length of 80 bytes, thus devices which deliver very little data result in somewhat inefficient storage usage.

The structure of a data file is shown below.



File Structure Subfile Structure

As indicated above, subfiles consist of three record types, one System Control Record (SCR), one Subfile Header Record (SHR) and one or more 80 byte data records. The SCR is used in the file search procedures and is not generally available to user programs. The SHR contains up to 40 bytes of information input to the system via the user device dialog routine and the following system information:

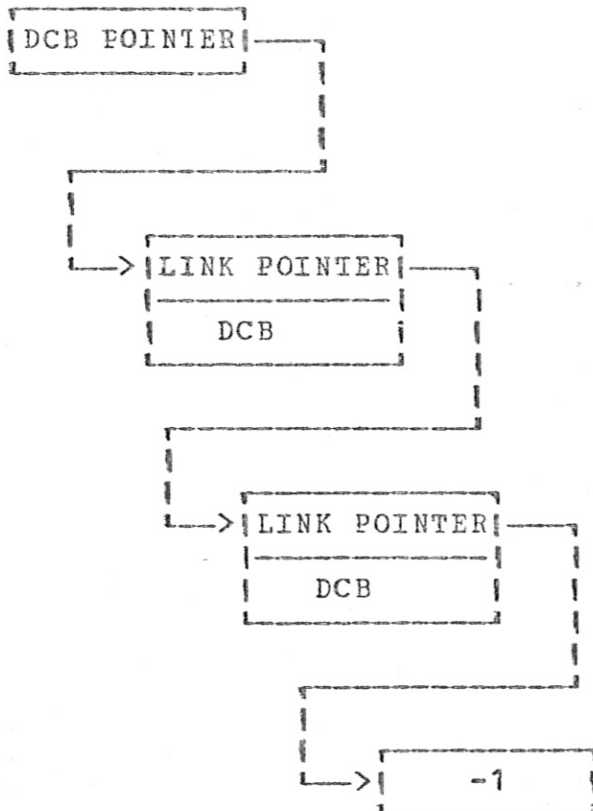
- 1) user device code
- 2) user device descriptor string length
- 3) user device descriptor string
- 4) number of data words
- 5) data word length expressed in bytes
- 6) date of the experiment

The SHR therefore provides the primary means of passing information pertinent to a given device and experiment to the user data processing programs.

2.5 User Device Descriptor and Control Blocks

The Device Descriptor and Control Block (DCB) forms the software interface between the DAS supervisor and specific user device dialog and driver routines. The list of DCB's forms the DAS Communications Vector Table. One DCB is constructed for each user device physically attached to the system. DAS is unaware of the existence of any devices for which no DCB exists, even though the necessary routines may be otherwise correctly linked to the system.

The DCB's are linked together in a simple chain as shown below.



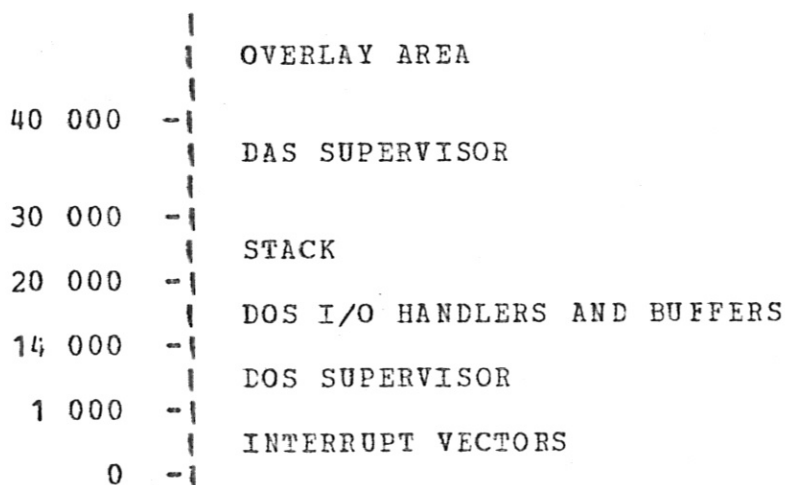
Each DCB is composed of three groups of parameter fields headed by a linkage pointer. The first group of fields contains a description of the hardware (interrupt type, attachment status, device address on the Unibus or CAMAC system) and pointers to the dialog and interrupt routines. The second group of parameters contains the device code number, a device descriptor text string, data descriptors, and the date of the experiment. The third group contains up to twenty parameters available for definition by the user either fixed or as inputs from the device dialog routine. The information contained in groups two and three is stored along with the original data in the data files and is therefore available for use in the data processing programs.

2.6 Memory Management

DAS is a single-user multi-tasking system based upon an overlay structure. In general, data acquisition and system control tasks are interrupt driven, providing a fairly high degree of multi-tasking. Due to limitations of the Overlay Supervisor, which uses a tree-structure in loading overlays, only one primary overlay area is available. Therefore, the system cannot provide services requiring routines from different overlay segments.

2.6.1 Memory Layout

The core memory is divided into six major segments as depicted below. The address values given (in octal) are approximate and depend upon the exact status of the computer at a given time. The diagram shows that the resident system requires approximately 40000(8) (16K) bytes; thus the computer should contain at least 32K bytes of memory so that the overlay area is of reasonable size.



2.6.2 Interrupt Vector Area

The interrupt vector area consists of 1000(8) bytes starting at address 0. The first 300(8) bytes are reserved by DOS for system communications and standard peripheral device interrupt vectors. The addresses 300 - 376(8) are used for DAS standard interrupt vectors and from 400 - 776(8) for CAMAC interrupt (LAM) vectors. All interrupt vectors consist of four bytes divided into a two byte pointer to the interrupt handling routine and a two byte processor status word.

2.6.3 DOS Supervisor Area

The DOS supervisor area contains the resident portion of the DOS supervisor standard on the PDP-11. These routines supervise the management of the standard I/O device handlers and buffer areas including the system disk. The user is referred to the DOS Concepts and Capabilities Manual (DEC-11-OCCMA-A-D) for more detailed information.

2.6.4 DOS I/O Handlers and Buffer Area

The memory area beginning at approximately 14000(8) contains the I/O handlers and buffers for the standard peripheral devices. Inasmuch as the handlers and buffers are dynamically supervised (attached and detached) by the DOS supervisor, no exact upper limit may be given. The value given of 20000(8) is reached when doing file transfers using the DAS file transfer utility but is not necessarily the maximum upper limit.

2.6.5 Stack Area

The region from approximately 20000(8) to 30000(8) is reserved for the stack or temporary storage area. The stack is managed as a standard push-down stack and is particularly useful for storing temporary data required in re-entrant and I/O routines. The I/O buffer area directly below the stack is software protected from stack overflow.

2.6.6 DAS Supervisor Area

The region beginning at 30000(8) and ending at approximately 40000(8) contains the DAS resident portion. The exact size of the resident supervisor is variable and depends upon exactly which modules are made resident when DAS is generated. Normally, only those modules are made resident which are absolutely necessary and consist of the following:

- 1) Function Keyboard I/O handler
- 2) Operator's Console I/O handler
- 3) Standard Peripheral device supervisor
- 4) DCE table
- 5) Overlay Supervisor
- 6) Overlay pointer and control tables

As implied in the above list, the resident portion consists primarily of routines and tables for handling I/O requests. In addition, modules required for asynchronous interrupt driven experimental device monitoring and control functions must also be included.

2.6.7 DAS Overlay Area

The overlay area is assigned the remaining storage beginning at approximately 40000(8). Therefore, for systems with a total core storage of 32K bytes, approximately 16K bytes are available for the overlay segments. The overlay segments contain modules for performing data acquisition, data processing and system control and monitoring. Only one overlay segment may be resident at a given

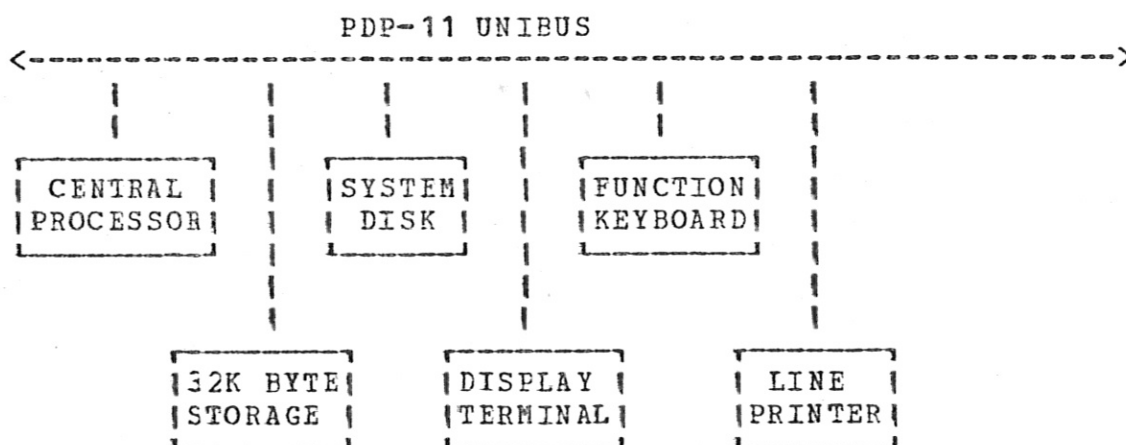
time; it may contain modules for performing one or more functions and may overlay itself. The user is referred to the PDP LINKER manual (DEC-11-ULKAA-A-D).

3.0 The DAS Hardware Environment

It is evident from the diagram in section 1.2.1 that DAS does not operate in a completely independent environment, but rather in response to operator and hardware demands. So long as no demands are pending, DAS is in a dormant or wait state awaiting the next demand request. The following sections describe the minimum and extended computer hardware and user device hardware environment.

3.1 Minimum Computer Configuration

The DAS system is written for the Digital Equipment Corporation PDP-11 family of computers and as such utilizes many of the hardware and software facilities available for that computer type. Detailed descriptions and specifications of the various hardware components may be found in the respective DEC processor and peripherals handbooks. A block diagram of the minimum DAS configuration is shown below.



3.1.1 Central Processor

The central processor required for DAS is any of the PDP-11 models 05 to 45. A minimum of 32K bytes of storage, either core, MOS, or Bi-polar is required, however larger storage configurations are supported as described in section 2.4. Provision is made in DAS via the means of conditional assembly directives to take advantage of various model-dependent features such as hardware multiply/divide and floating point instructions. A time-of-day clock is also required for the generation of system messages.

3.1.2 Operator's Console

The first release of DAS supports a single Operator's Console which consists of the following three hardware modules:

- 1) alpha-numeric and graphical display screen
- 2) typewriter keyboard
- 3) function keyboard

The first two modules are generally combined into a single hardware unit, an alpha-numeric and graphics display terminal. DAS supports Tektronix 4002 and equivalent teletype compatible display terminals. For purposes of controlling DAS, a teletype or equivalent printer terminal could be substituted, however response times would be considerably increased due to the operator dialog concept of DAS. All messages and requests for operator input are displayed as text strings on the display. Replies are entered by the operator as alpha-numeric text strings via the typewriter keyboard. In addition, graphical displays requested by the operator are plotted on the display screen.

The third module of the Operator's Console, the function keyboard,⁽¹⁾ is a set of 16 momentary contact push-buttons with associated status lights used to select DAS Function Groups and Functions. DAS responds to a function keyboard demand by turning on the associated status light, indicating the operator demand has been accepted and is being processed. The system status may be easily determined from the status lights which are turned on. The wait state is indicated when they are all turned off.

3.1.3 System Disk

The DAS system disk is required for storage of the DAS overlay module files and system files. Temporary files required by user data processing programs are normally routed to the system disk. In addition, if the computer installation contains only one disk, then data files are stored on the system disk.

3.1.4 Line Printer

The minimum DAS hardware configuration contains a 132 column wide line printer used for the output of the Operator's Console log file, the list of storage medium (disk, tape) contents and the results of user data processing programs. If no line printer is available, these outputs may be routed to the Operator's Console.

3.2 Additional Computer Peripherals

DAS supports a number of other standard PDP-11 devices which are not included in the minimum configuration. Up to 56K bytes of storage are supported which would provide approximately 40K bytes for the DAS overlay area. Installations with more than one disk storage unit may use one disk unit as the DAS system disk and one or more disks for data file storage. The various magnetic tape storage mediums (DEC-tape, cassette tape and 9 track magnetic tape) are supported as long-term data file storage mediums.

The devices supported by DAS fall generally into three categories, which are as follows:

⁽¹⁾ Circuit diagrams (drawing number EL 261) for the function keyboard are available from D. Zimmermann.

- 1) standard interrupt based devices
- 2) CAMAC based devices
- 3) direct memory access based devices

The method by which the input/output of data is accomplished is implied in the above list. User devices may also be handled in programmed transfer mode where higher speed transfer rates are not required. For installations where very high total transfer rates are needed, data may be stored in the user device during the experiment. The external storage devices are scanned by the task manager after an experiment but the data is stored in a data file in the normal manner.

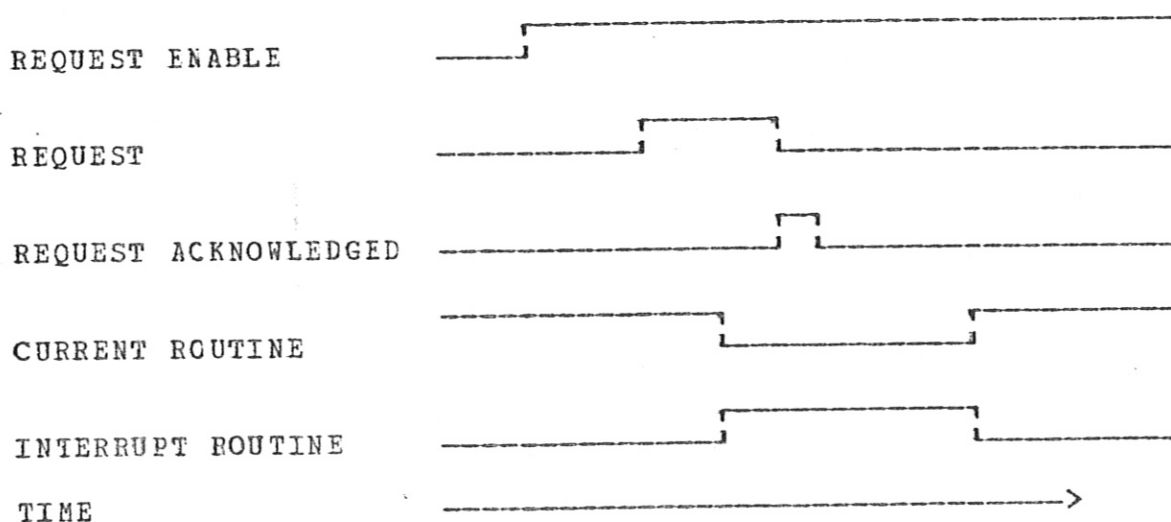
3.3 User Hardware

3.3.1 Standard Interrupt Based Devices

Standard interrupt devices are attached directly to the PDP-11 via a DR11-C interface. Two triads of signals are defined for the interface; the first (A) group signals the transfer of one data word, up to 16 bits in length; the second (B) group signals the end of transfer of a group (or logical record) of one or more data words in length. All transfers are based upon the so-called "hand-shake" principle. The three signals are defined as follows:

- 1) Request Enable, when set, allows an incoming demand from the device to be serviced by the interrupt service routine. If not set, the demand is ignored.
- 2) Request, when set, indicates a demand from the device. The currently running routine is interrupted and the device demand is serviced by the interrupt service routine. The request signal must remain set until a request acknowledged signal is returned to the device, at which time the demand must be reset.
- 3) Request Acknowledged, when set, indicates that the interrupt service routine has been started and the current request has been accepted. The request signal should be reset.

The time sequence for a standard interrupt is depicted below.



3.3.2 CAMAC Interrupt Based Devices

Devices based upon the CAMAC standards (see EUR 4100) are treated by DAS in a manner similar to that for standard interrupt devices. Interrupt service routines and, possibly, user dialog routines are also required. The primary difference between the two groups of devices lies in the manner in which they are connected to the PDP-11, either directly or via the CAMAC system.

The "hand-shake" triad of signals for a CAMAC device are as follows:

- 1) Request Enable is generated by enabling the CAMAC System Controller interrupt, the Crate Controller Type A Branch Demand, and the CAMAC module LAM via ENABLE LAM to the appropriate module sub-address;
- 2) Request is generated by the CAMAC module via a LAM;
- 3) Request Acknowledged is generated via a CLEAR LAM to the appropriate module sub-address.

3.3.3 Direct Memory Access Based Devices

The logical attachment of direct memory access (DMA) device to DAS once again implies the attachment of up to two software tasks as described in section 2.1.1. The primary difference lies in the means by which data is transferred. DMA data transfers are automatically carried out by the hardware, thus only initialization and end of record functions need be handled by the interrupt routine. Data transfers are therefore considerably faster for DMA devices so that a currently running routine is interrupted for a much shorter period of time than is the case with standard devices.

References

DOS/BATCH Concepts and Capabilities; Digital Equipment Corp;
Maynard, Mass; DEC-11-OCCMA-A-D

DOS/BATCH Monitor Programmer's Manual; Digital Equipment Corp;
Maynard, Mass; DEC-11-OMPMA-A-D

DOS/BATCH Linker Programmer's Manual; Digital Equipment Corp;
Maynard, Mass; DEC-11-ULKAA-A-D

PDP-11 Processor Handbook; Digital Equipment Corp; Maynard,
Mass;

PDP-11 Peripherals Handbook; Digital Equipment Corp; Maynard,
Mass;

CAMAC, A Modular Instrumentation System for Data Handling;
Commission of the European Communities; Luxembourg; EUR 4100e