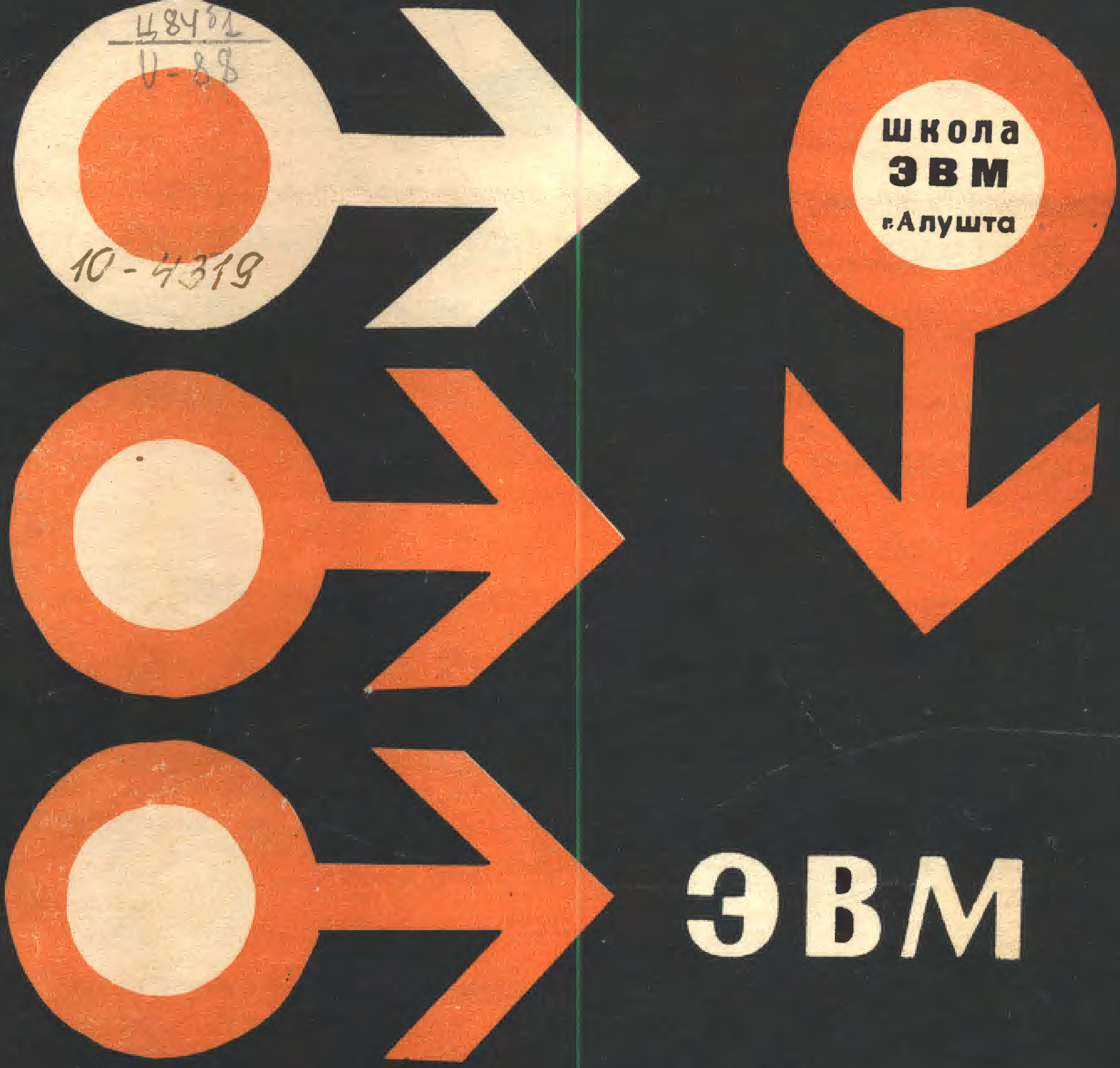


школа
ЭВМ
г.Алушта

ЭВМ
В ЭКСПЕРИМЕНТАЛЬНОЙ ФИЗИКЕ

1968 г.

ОБЪЕДИНЕННЫЙ ИНСТИТУТ ЯДЕРНЫХ ИССЛЕДОВАНИЙ
ЛАБОРАТОРИЯ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ И АВТОМАТИЗАЦИИ

10 - 4319

THE USE OF COMPUTERS IN EXPERIMENTAL PHYSICS

Лекции, прочитанные в Школе ОИЯИ
по применению электронных вычислительных
машин в задачах экспериментальной физики.
г. Алушта, Крым, СССР, 5-19 мая, 1968 г.

Dubna 1968

Широкое внедрение электронных вычислительных машин в различные звенья физического эксперимента за последние годы вызвало определенный интерес со стороны физиков-экспериментаторов к вопросам вычислительной техники и программированию. Персонал физических лабораторий при подготовке экспериментов или обработке экспериментальных данных вынужден (в большинстве случаев самостоятельно) осваивать технику ЭВМ и методы работы на вычислительных машинах.

При всем многообразии материала как по самим вычислительным машинам, так и по вопросам программирования, в процессе такой работы возникают естественные трудности, связанные, главным образом, с ограниченностью литературы, рассчитанной на физика-экспериментатора или на лиц, занимающихся развитием методических вопросов экспериментальной физики. Если учесть при этом, что методика использования ЭВМ в экспериментальной физике быстро совершенствуется, то будет понятен интерес со стороны физических институтов к летней школе Объединенного института ядерных исследований - "Применение ЭВМ в задачах экспериментальной физики".

Школа проводилась Лабораторией вычислительной техники и автоматизации ОИЯИ (директор - член-корреспондент АН СССР проф. М.Г.Мещеряков) в г.Алуште (Крым) с 5 по 19 мая 1968 года.

Программа школы наряду с основополагающими вопросами включала также лекции по некоторым конкретным современным методикам. Для чтения лекций были приглашены ведущие специалисты из Объединенного института ядерных исследований, институтов стран-участниц ОИЯИ, а также коллеги из европейских исследовательских центров - ЦЕРНа (Швейцария) и Сакле (Франция).

Не имея возможности опубликовать весь материал, ректорат Школы подготовил к изданию отдельные лекции, сохранив, в основном, их в том виде, в котором они были представлены авторами.

Лиц, интересующихся лекциями в полном объеме, мы адресуем в библиотеку ОИЯИ, где находится полный сборник прочитанных в школе лекций: "Применение ЭВМ в задачах экспериментальной физики".

Ректор Школы
доктор технических наук

Г.ЗАБИЯКИН

Отпечатано методом ксерокс-ротапринт с материалов, подготовленных ректоратом Школы.

CONTENTS

J.C.GOUACHE, J.TREMBLEY. The L.S.D. Project. (Spiral Reader)... 5

A.LEVEQUE. Measurement Problems in Large Hydrogen Bubble Chambers 25

H.LIPPS. 6000 Series SCOPE at CERN 45

D.LORD. Review of On-Line Computers and Their Uses in High Energy Physics 63

J.ZOLL. Program SUMX 79

Рукопись поступила в издательский отдел
13 февраля 1969 года.

THE L.S.D. PROJECT

(Spiral Reader)

J.C.Gouache and J.Trembley[x)]

Abridged and edited by I.Skryl

Introduction

High energy physics and the study of the elementary particles require higher and higher statistics. In the field of bubble chamber physics, higher statistics mean more and more particles to scan and to analyse. To satisfy such needs, automatic machines are under construction or development.

The first automatic machine was the HPD or FSD[xx/] with its mechanical flying spot digitizer. The following two machines were the PEPR[xxx/] and the Spiral Reader. The PEPR differs from the well-known HPD in two principal ways.

I) The spot of light which scans the photograph in PEPR is generated by a CRT (cathode ray tube) rather than by mechanical means.

2) The spot of light is capable of being expanded into a line segment with any desired orientation. This line segment detects track elements in an analogue manner, some of the non-track background signal is automatically filtered out.

The Spiral Reader differs radically from the HPD and the PEPR in that it has an operator "on-line". In addition it has its own method of digitization invented in 1958 by B.McCormick from the Alvarez

---

[x)] CERN - College de France

[xx/] HPD or FSD: Hough Powell Device or Flying Spot Digitizer.

[xxx/] PEPR: Precision Encoding and Pattern Recognition.

group (Berkeley) where it started to operate for Physics experiments in 1965. A radially oriented mechanical line segment (a slit) scans spirally outwards from a vertex. An operator looking on a projection table centres the spiral scan on the vertex of the event and helps if necessary the measurement by giving instructions to the computer. The filtering of the Digitizations is performed off-line on a large computer (in the case of Berkeley and CERN on a CDC 6600).

Until now the most automatic machines, HPD and PEPR, require human intervention; this is done off-line by manually measuring a few points on each track, to prepare roads which are needed for the final evaluation. With the Spiral Reader the assistance is given during the measurement by the precise positioning onto the vertex.

Fig.I is a photograph of the Spiral Reader at CERN with its electronics and on-line computer (PDP 9), taken in April 1968 during the mounting of the machine.

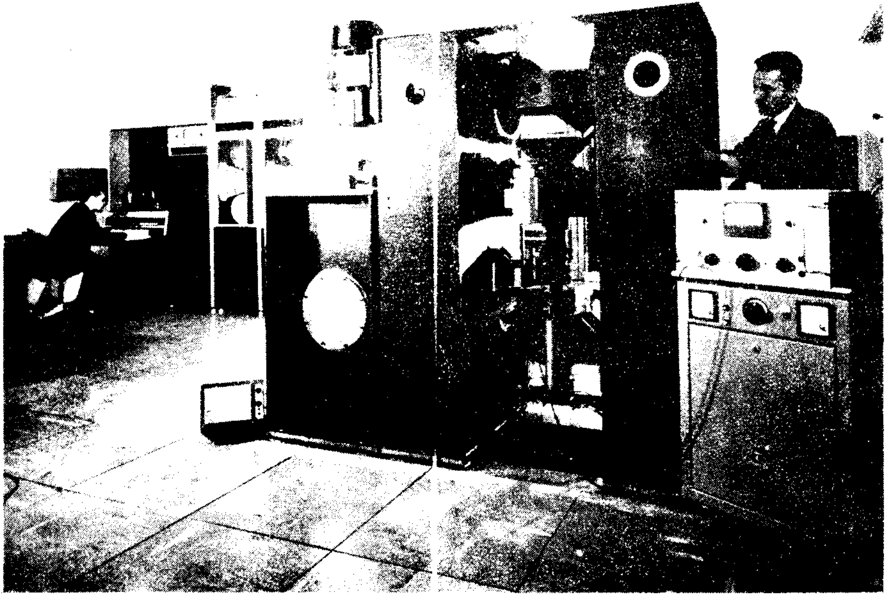


Fig. I. The Spiral Reader at CERN with its electronics and on-line computer (PDP-9).

Principles of the LSD (Spiral Reader)

The LSD performs the analysis of the events using the particular star pattern of the reaction in measuring them with polar coordinates ($R = \theta$) related to the normal cartesian coordinate system by the fiducial and vertex X-Y coordinates.

The polar system scans the film optically from its centre point outward (increasing radius) and clockwise (increasing angle). The centre of the spiral scan may be moved to any point of the film with the help of the two stages and is referenced by the rectangular coordinate system. As the scanning slit which is mechanically realized crosses a track, a photomultiplier tube produces a pulse which initiates the storage of the polar coordinates at that time. When the data is processed off-line, the computer will reconstruct the event and reference it to the fiducial marks.

II. Description

II.I Mechanics and Optics

Fig.2 is a sketch of the general assembly of the machine showing, from left to right :

a) The projection table

b) The film transport system

c) The X-Y stage with the optics housing above it

d) The cone-periscope

e) The auto-fiducial system and the TV camera

II.I.I Film Transport System

The film transport system is designed to take 300 metre spools of unperforated film of either 35 mm or 50 mm. The maximum speed is

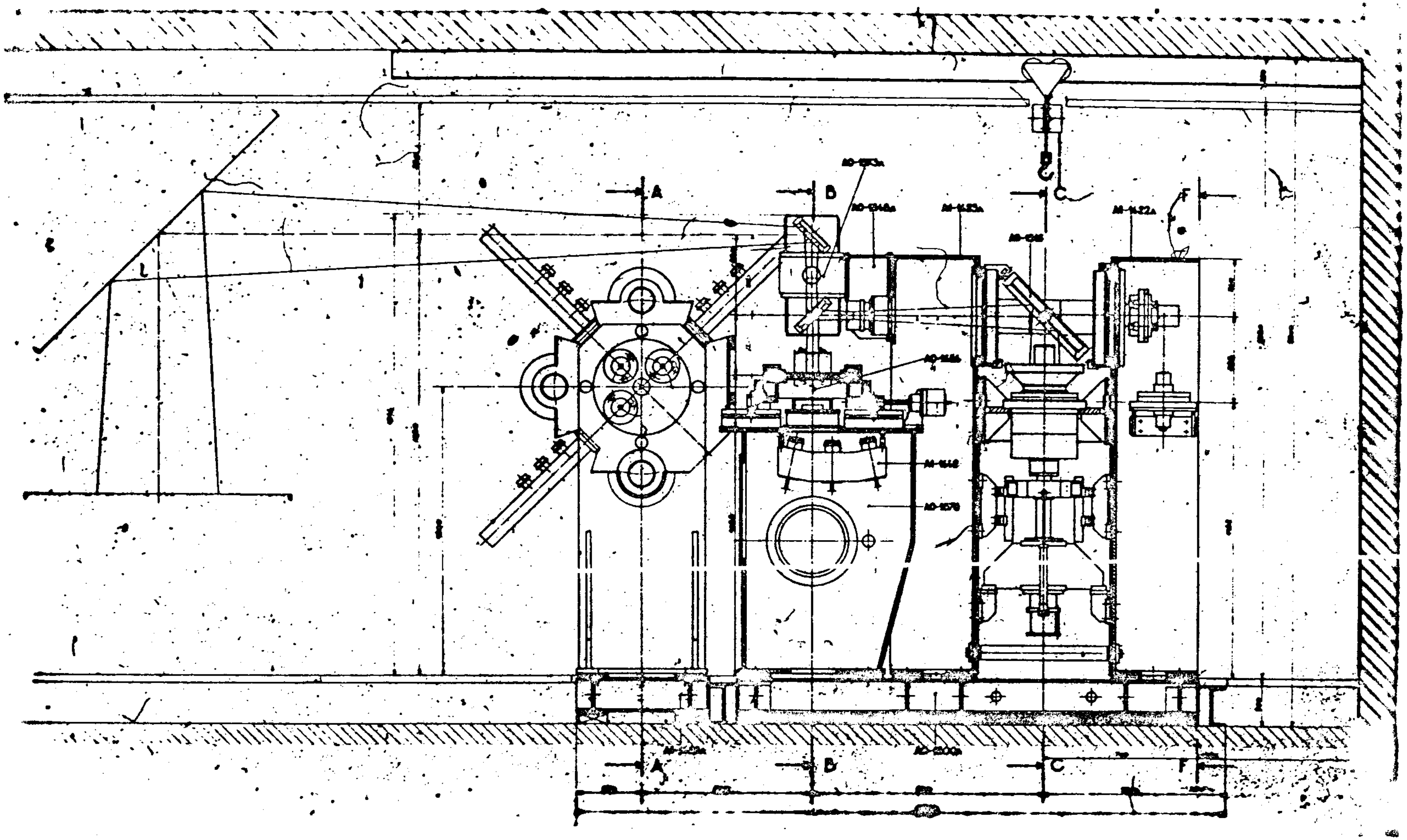


Fig.2. A scetch of the general assembly of the SR showing.

5 metres/sec. The film transport which takes the three stereoviews, makes an angle of 90° with the image position on the stage. The film, after passing through a vacuum buffer, turns 90° by sliding on 45° rollers, thus permitting the X translation of the upper stage, the Y movement of the lower stage; both movements are absorbed by the vacuum buffer of the transport.

II.I.2 The X-Y Stage

The X-Y stage is formed by two movable tables running on rollers and driven by ball-screws directly coupled to printed circuit motors. The lower table, moving in the longitudinal direction of the film, has a total travel of 200 mm and the upper one a cross movement of I50 mm. The maximum speed of the tables is I2 cm/sec. On the upper table a film clamp system maintains the films in place during the measurement by pressing them between two glass plates. The table position is picked up by Heidenhain linear encoders giving a least count of 2.0 microns.

The movement of the stage can be obtained either under computer or under manual control. The manual control is realized by a speed ball and used principally for fiducial measurement, vertex positioning and crutch points.

The centring on the vertex or fiducials and the crutch points, is observed on a TV set with a higher magnification than on the projection table (200 times compared to I0 times).

A film frame counting system (Brenner mark detector) is mounted on the upper stage which steps up a film frame counter in the computer to permit the research of the next frame and to position the picture to within 2/I0 of a millimetre.

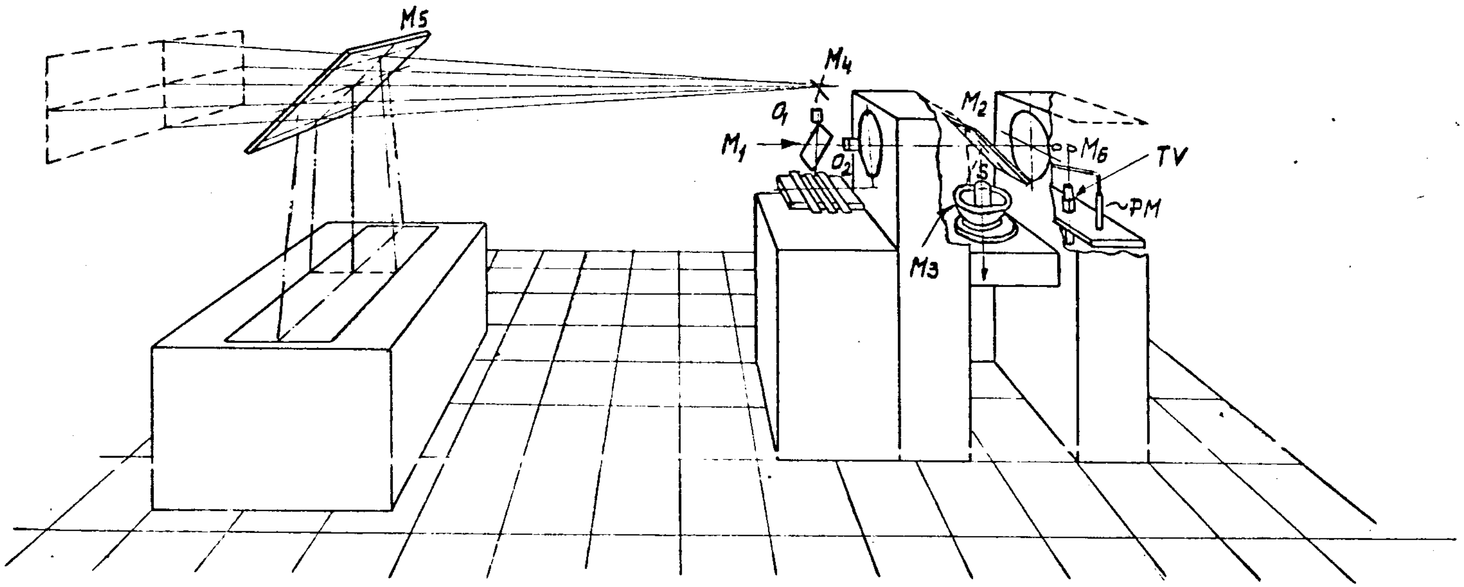


Fig.3. A scheme of the general optics layout of the SR.

II. I.3 The Optics

Fig.3 gives a scheme of the general optics layout of the machine. The light is produced by a Xenon lamp delivering 450 Watts and illuminating the film after a 90^o deflection on to a cold mirror to avoid film heating. The machine is designed to project only one view at a time while the film transport system is able to take the three views. The passage from one view to the other is done with the lower stage (Y-displacement of the X-Y stage).

The semi-transparent mirror No.I splits the projection in two parts, one part going via objective No.I, mirrors No.4 and 5 on the projection table, the other part being reflected on the cone for the measurement.

The projection for the measurement is done by objective No.2 on the virtual horizontal plane A. In reality the image is focused on a slit "S" (mounted on a sliding piece moving up and down along the axis of a cone supporting the mirror No.3 and turning around) after the reflection on the last mirror No.3 fixed on the edge of a rotating cone.

The first objective is a Schneider Repro-Claron (355:I:9) giving a magnification of IO on the table. The second objective is again a Repro-Claron (305:I:9), the magnification on the slit being 2.8 times the image on the film.

The mirror No.2 is a semi-transparent mirror leaving part of the image to be projected on the TV camera and automatic fiducial measuring system.

II.I.4 The Cone-Periscope

The cone-periscope assembly consists of a vertically ball-bearing suspended drum, which rotates presently at a constant speed of 900 turns per minute. To provide a spiral scan of the event centred on its apex, a periscope tuble is moving up and down along the axis of a cone. The periscope movement is independent of the rotation of the cone. The light reflected by the mirror No.3 is projected on a small slit (50μ x I000μ) and deflected by a silvered prism on a fibre optic pipe which then guides the light to a photomultiplier situated outside the cone-periscope assembly. The vertical movement of the periscope is achieved by a printed circuit motor acting on a ball screw. The R and θ coordinates are provided by Heidenhain encoders. The R encoder is a standard linear encoder of the same type as the X-Y encodes of the stages. The encoder is specially developed by Heidenhain, Germany; it is a disc with 32,400 division per turn (I2 micron light - I2 micron dark), multiplied electronically by 4 to produce a least count of 2.5 microns at the largest scan radius on the film.

Presently the size of the spiral scan is 80 cm in radius in the 2m chamber and 53 mm of the film.

For controlling the movement of the periscope, its position is given by a linear potentiometer providing feedback information in the position servo for maintenance.

II.I.5 The Auto-Fiducial System

The fiducial measuring system is arranged such that it will be possible to measure 4 fiducials simultaneously. Four adjustable "V"-shaped slits (5mm x 20μ) are mounted in front of 4 pairs of fibre optics light guides connected to 8 photomultipliers. The system has to be changed and readjusted for each chamber type and is now designed for the HBC 200 and HBC 8I of CERN.

II.2. Control Electronics and Interface to the PDP9 Computer:

II.2.I. The PDP9 Computer and Standard Peripherals.

The PDP9 is a computer manufactured by "Digital Equipment Corporation", Maynard, Massachussets. It is an I8 bit machine, provided in its basic version with an 8I92-word memory, a model 33 send-receive Teletype, and a paper-tape reader-punch. The teletype is used by the computer to permit messages to the operator and is used by the operator to provide requested information to the computer (operator number, data, present film frames etc...) or to alter the normal sequence of the program, as will be discussed later. In addition to this basic configuration we have purchased 2 magnetic tape units with controller: one unit for input of scanning information, the second one for output of digitizings from the spiral scan and for output of indicative information. Finally, in order to display for the operator the R - θ coordinates of the digitizings which are written onto the output tape, we have purchased the type 34H display controller with a RM 564 Tektronix storage oscilloscope.

II.2.2. Flow of Data Between the Different Parts of the Machine:

The LSD might be considered as consisting of 3 parts: the computer, the mechanical assembly, and the operator. Each one of these 3 parts sends or receives messages which are not directly usable or understandable. The role of the control electronics, at the center of these three poles, is to work out the necessary translations and allow transmission of data between the three parts.

Fig.4 shows how all data paths flow through the control electronics (except the exchange of data between computer and standard DEC peripherals, i.e. tape units, teletype, storage oscilloscope).

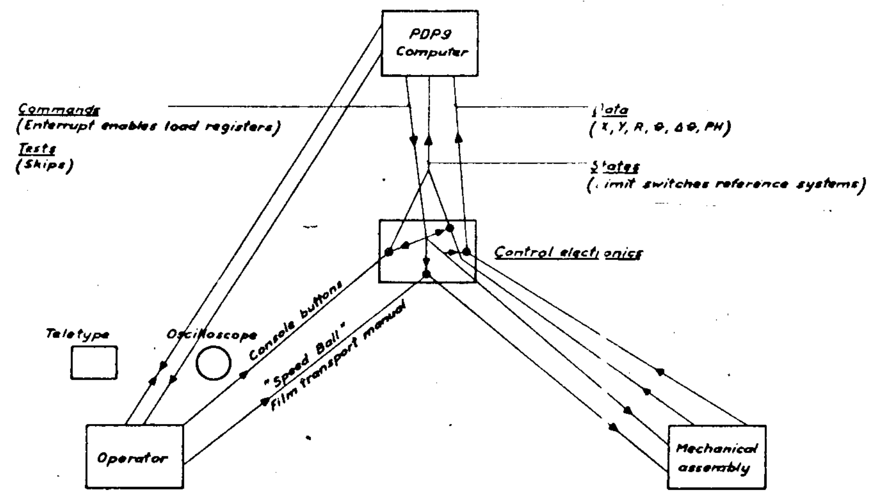


Fig.4. Data path flow through the control electronics.

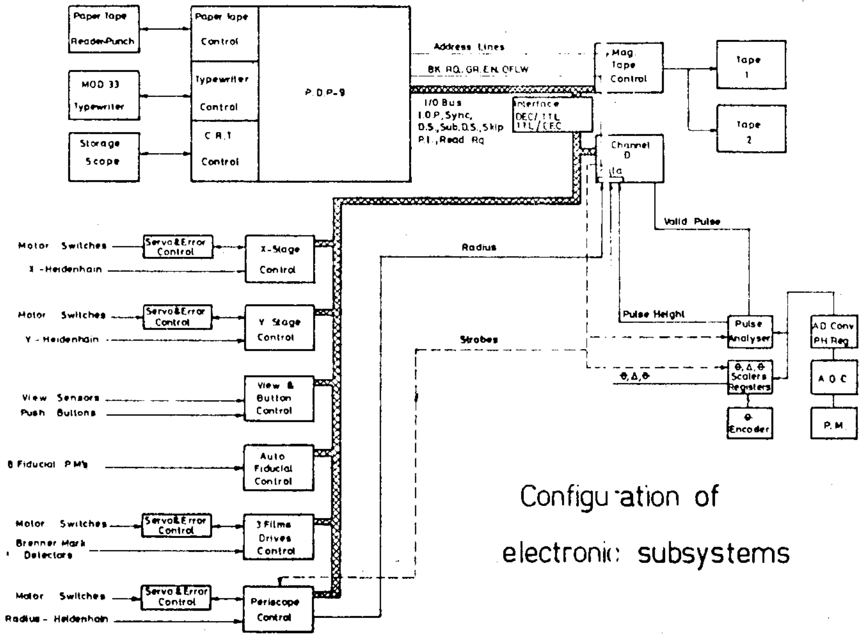


Fig. 5. Configuration of electronic subsystems.

From this figure, one can see for instance that the computer sends commands such as a velocity for the periscope or the X-Y stage. This command loads a so-called "velocity register" in the control electronics, the contents of which determine the proper width of the 20 kHz pulses applied to the printed circuit motor located in the mechanical assembly.

Another example: the operator initiates an operation during the measuring sequence by depressing any one of the IO console buttons. This action sets the corresponding flip-flop of a so-called "command and status register" in the control electronics. The content of this register read into the computer directs the program to take the action the operator wants to be performed.

These two examples show clearly the role of transmission, translation and coordination of the control electronics. Without further analysing how the signals are dealt within the control electronics, let us mention the different kinds of signals it will send or receive:

a) from the computer it receives:

- commands (e.g.to read a register into the computer)
- data (e.g. a velocity)
- tests (e.g. the next sequential instruction of the program will or will not be skipped depending upon the stage of a given flip-flop)
- enables (e.g. a given flip-flop is permitted to cause a program interrupt when it comes to a I)

to the computer it sends:

- data (e.g. X-Y coordinates)
- results of tests (e.g. the pulse which will cause a skip)
- Interrupt (e.g. if previously enabled, the flag corresponding

to a limit switch on the X stage will cause an interrupt when the stage hits the limit)

b) from the mechanical assembly it receives:
- raw data (e.g. the periscope displacement from the Heidenhain linear encoder)
- mechanical states (e.g. which view is positionned under the optical axis from the view sensors, or the crossing of the reference marks)

to the mechanical assembly it sends:
- command pulses to the motors
- command signals to the film damp and the film transport system (e.g. stop film accurately when the next Brenner mark passes under the detector)

c) from the operator it receives:
- requests from the console buttons
- displacement commands of the X-Y stage via the "speed-ball" .

II. 2.3. Configuration of Electronic Subsystems:

The LSD control electronics is divided according to the mechanical divisions. Except "interface" which simply translates the DEC negative logic (0, -3V) to the "Texas Instruments" positive logic (0, +5V) and conversely, no other element is common to the electronics subsystems; each one constitutes an independent whole which separately controls its corresponding mechanical part.

The logic part of these electronic subsystems is designed with microcircuits of the so-called TTL (Transistor-Transistor Logic) technology manufactured by "Texas Instruments" , series SN 74N and

SN 74 H. The analogue parts use integrated operational amplifiers manufactured by "Fairchild" (μA700 Series) and "Motorola" (MC I433

Figure 5 shows the different electronic subsystems. All parts situated above the interface constitute the computer and its peripheral equipments. All parts below constitute the control electronics. On the left hand side can be seen the subsystems which are constantly under direct control of the program, which cannot perform any single operation without the appropriate command from the program. On the rigth hand side, the digitizings channel (Channel D), only needs to be initialized by the program; two registers are loaded - word count and initial address register - which define respectively the size and starting address of the input buffer for the spiral scan digitizings; once the channel is initialized the digitizings come at their own rate with no further intervention of the program until the spiral scan is finished.

This figure also shows how the 3 PDP9 words are inputted for each digitizing (total 54 bits). A pulse coming from the periscope photo-multiplier (PM) is first normalized by an automatic gain control (AGC) system and is then digitized via an analog-to-digital converter (A.D.Conv.), the pulse height being stored in a pulse height register (P.H.Reg.). If the edges of the pulse correspond to given specifications the Pulse Analyser issues a signal "Valid Pulse" to the Channel D electronics. This signal in return, causes Channel D to issue 3 strobe signals, separated by 4μS, to input successively the radius from the periscope scaler, the angle θ and pulse width $\Delta\theta$ as given by the θ encoder, and the pulse height as stored in the pulse height register.

II.3 Software

The necessary software for the LSD consists essentially of 4 different programs:

- an on-line control program "ASTERIX" (size 6k-PDP9 words)
- a maintenance and check program "RAVEN" (size 2.5k-PDP9 words)
- an off-line filter program "POOH" (size I3k-CDC 6600 words)
- a calibration program "SCALP" (size I0k-CDC 6600 words)

In addition let us mention that the control and maintenance programs are punched on cards and are assembled on the CERN CDC 6600. This produces a binary magnetic tape containing relocatable subroutines which are linked together at loading time. The assembler which was initially developed in Berkeley has been considerably improved. The linking loader has been entirely developed at CERN.

II.3.I. The On-Line Control Program

This is the normal working program during LSD operation. It does no calculation or filtering but just outputs onto magnetic tape the raw data from the spiral scan, the vertex, crutch points, stopping points and fiducial coordinates. It directs the measuring sequence and makes provision for the operator to modify this sequence (remeasurement of a vertex, of an entire event, reject of an event etc.) via the teletype keyboard. It prints messages to the operator: indications or comments of the scanner, error conditions, current roll, frame, experiment numbers, requests a new magnetic tape, etc. Before the first event is measured it calibrates the different periscope velocity ramps for the spiral scan (more points are needed around the vertex than far away), and checks for consistency of the X,Y and periscope scalers during measurement.

The program also controls the data collection from the spiral scan. In order not to limit the number of digitizings which are inputted for one vertex, the program makes use of a dynamic buffering, i.e. two input buffers are filled alternatively by the digitizings coming from Channel D electronics and a full buffer is immediately emptied onto magnetic tape and is ready to accept more data when the other buffer is full in its turn.

While the digitizings are coming in, the program displays on the memory scope the values of R and θ which are being written into the input buffer before it is outputted onto magnetic tape. Another feature of the program is the automatic measurement of the fiducial marks. Eight photomultiplier tubes receive light from four pairs of slits parallel to the legs of 4 fiducial crosses. For measurement, a sweep of the X-stage is initiated by the program. Positions of the stage are stored for leading and trailing edges of the pulses generated by the 8 fiducial legs in their appropriate photomultiplier tube. Under the assumption that the legs have an angle of $\pm 45^{0}$ on the X-axis, the centre of the fiducial crosses can be calculated. To be accepted however, as the final fiducial positions, they have to pass, successfully, consistency checks and more precisely, that the distances between one and the other lie within given tolerances. If these checks fail for one or several of them, the operator is requested to measure them manually. The program also controls the film transport system. For instance, during initialization the operator is requested to provide the present frame numbers. The frame number of the first event to be measured is read from the input tape and the 3 films are moved and correctly positioned on the desired frame number.

The 6k-word control program is entirely written in machine language. The teletype input output routines occupy third of the program, the automatic measurement of fiducials another third, the rest being used for periscope, stage, film transport control, data input-output, display and utility programs. The 2 input buffers are presently 999 words long.

II.3.2 The Maintenance and Check Program

This program had been originally developed at Berkeley and was simply adapted, taking into account CERN modifications.
Its operation involves the taking of I500 digitizings from a spiral scan (this implies a 4500 word input buffer) and the display of the entirely or part of the data. Under command of the teletype keyboard one can display points within some range of pulse height or pulse width to check the functionning of the pulse analysis system. One can also intensify some points of a given pulse height against a background of all the data. One can select an area of the display and magnify it on the entire display range to check the optics, the mechanical precision, the linear and rotary encoders. One can display leading edge, mid-point or trailing edge of the periscope pulses. Finally one can calculate the percentage each bit in the 3 data words is in the "one" state, to detect any "stuck bit" and consequently any failure in Channel D electronics.

II.3.3 The Off-Line Filter Program

CERN has presently no experience with the filter program since the first LSD output tape has only been produced very recently, and due to the absence of the Heidenhain rotary encoder, the θ values are very crude. However, let us mention the principles of the filter program as it is implemented and in production in Berkeley. The points digitized in the first 1/5 of the spiral scan, where the filtering effect of the slit is of most value, are histogrammed into bins of a 3.5^{0} width, which corresponds to I.5 track widths at the origin of the radial length and IO track widths at the upper limit of the bins. The bins have successive slopes comprised between two maximum values which correspond to a minimum radius of curvature for the tracks considered. When a bin contains more than one half of the number of cone revolutions in the histogramming region, the points are considered as constituting a possible track. When two bins contain the same points with a minimum percentage of 60% they are considered as belonging to the same track. An equation of the type: $\theta = \theta_0 + \alpha\tau + \frac{\beta}{\tau} + \gamma\tau^3$ is then used for a least squares fit of the points and those which do not fit within given tolerances are successively eliminated. When this histogramming is terminated, the tracks are extended upwards. Each time a new point is added the fit parameters are updated, the point is considered as belonging to the track if the fit parameters lie within given tolerances from the parameters of the preceeding fit.

If a "crutch point" has been added by the operator, a similar histogramming takes places downwards (in the vertex direction), the tracks are followed in the same way down to the vertex and then from the crutch point upwards, as far as possible.

At this stage, the filter program may have provided spurious tracks. A match in space, using information from the three views finally performs elimination of spurious tracks by consistency checks on direction, sign and magnitude of curvature, dip and azimuth.

II.3.4 The Calibration Program

The existence of this program is due to the fact that the spiral reader uses two different sets of coordinate systems: X-Y rectangular coordinates for the fiducials, vertices, "crutch points" and stopping points, and R-θ polar coordinates for the spiral scan digitizings. The calibration program calculates the coefficients for the transformation of the polar coordinates into rectangular coordinates. This is done by measuring a so-called "chicken path" pattern with a spiral scan. The X-Y coordinates of the cross centres are known precisely by construction of the glass plate. The X'-Y' coordinates of the cross centres are calculated from the R-θ coordinates of the LSD output data. Partly empirically, partly using formulae of the known mechanical and optical distortions, an equation is established which allows, by interpolation between crosses, the transformation from the X'-Y' to the X-Y system and ultimately the R-θ to the X-Y system.

III. Associated Topics

III.I. Accuracy of the Spiral Reader

The CERN-College de France L.S.D. Project being in the period of mounting and testing and the θ encoder not yet installed, it is impossible for us to give any information about the accuracy of measurement from our machine.

The only comments possible on that subject are those made by the Berkeley group on their machine Mark I. (Spiral scan radius 40 cm, and off-axis optics).

It is always claimed by the Berkeley group that the accuracy of the measurements made with the S.R. is comparable with the accuracy of the conventional measuring machines (Franckenstein or IEP) on the same length of tracks (G.Lynch UCRL I7328).

Naturally we have to keep in mind when accuracy is discussed, two things that limit the precision:

I) The slit is radial and would digitize only the dark spots that cut by more that 20% the light going throught the slit. In this case a track that has turned from 20 to 40 degrees is no more digitized.

2) The length of track measured is finite and equal at best to the radius of the spiral.

S.R.Berkeley Mark I - 40 cm
S.R.Berkeley Mark II - 80 cm.
L.S.D. CERN - 80 cm.

In both laboratories the radius of 80 cm is achieved by having a greater demagnification optically and not mechanically.

Measurements done on IEP's at CERN show no remarkable difference between measurements on 40 cm or 80 cm, or 80 cm but with half magnification.

We hope to be able to measure with a precision as good as for the IEP, but this has to be confirmed experimentally, when the L.S.D. will be in operation.

References

1. L.W.Alvarez; Round Table Discussion on Bubble Chambers, Proceedings of the 1966 Conference on Instrumentation for High Energy Physics.
2. R.K.Böck; Notes on SR Accuracy, Alvarez Physics Note 59I, July 6, I966.
3. G.V.Butler; Spiral Reader Electronics - Description and Specifications, Nov. IO 1966. Lawrence Radiation Lab., UCID 2842.
4. G.V.Butler; SR System Electronics Conventions and Guide Lines, UCLRL Engineering note, April 26, I966.
5. J.Burkhard et al.; A Filter Program for the SR, Proceedings of the I965 Conference on Programming for Flying Spot Devices, Columbia, New York I966.
6. G.R.Lynch; The Accuracy of the SR, Alvarez Physics Note 575, Oct. 22, I965.
7. G.R.Lynch et al., The POOH Scrapbook, Alvarez Programming Note.
8. J.Stedman; SR Control and Data Acquisition with PDP-4, Nov.29, I965, UCRL I6555.
9. G.R.Lynch; Remarks on the use of the SR, Jan. I967, UCRL I7328.

MEASUREMENT PROBLEMS IN LARGE HYDROGEN BUBBLE CHAMBERS

A. Leveque[x)]

Abridged and edited by S.Corenchenko

At Serpukhov for the first time intense beams of high energy particles have become available; opening a new field in particle physics. Present bubble chambers have well known limitations; in particular they do not provide unambiguous identification of high energy interactions: this was the main motivation for building larger chambers.

The purpose of these lectures is to discuss the new experimental possibilities offered by this new generation of equipment. The major concern is the accuracy of the data since reactions will be identified on the basis of kinematic analysis. An important quantity is the "measurements error parameter" ε which depends on quality of the chamber; rough estimates of ε will be given, reviewing various sources of errors specific to large chambers. Expected physical results and limitation will be investigated, mainly from the point of view of very high energy interactions.

I would like to acknowledge stimulating discussions with C.GENSOLLEN, J.LOTTIN, C.FRANK and R.HUBBARD while preparing these notes.

[x)] Sacle, France

SECTION I. POSITION ERRORS

In standard treatment of bubble chamber data, the measurements error ε is usually defined as the root mean square deviation of measured points from the fitted curve after reprojection on the front glass of the chamber; the measurements errors on momentum and angles Δp, $\Delta\theta$ and $\Delta\lambda$ are all proportional to ε.

When a charged particle traverses a plate of Huckness t, it undergoes a large number of collisions, most of them producing very small angular deflections.This additional error varies as $\ell^{1/2}$ while the measurement error on angular goes with $\ell^{-3/2}$. The total angular error is minimum for some optimum length, proportional to $\sqrt{p\varepsilon}$ and independent of the applied magnetic field.

ILLUMINATION AND OPTICS IN GIANT CHAMBERS

As far as data processing is concerned, the main difference between the new generation of bubble chambers and the present one is related to the new optics and illumination devices.

It seems appropriate to recall outlines of those designs before going into possibles sources of error specific to those giant chambers.

Table I shows the main specifications of some giant chambers now under construction. With the exception of Mirabelle, hydrogen chambers come in the form of a short vertical cylinder. Mirabelle and propane chambers are long horizontal cylinders. As a consequence the first type calls for a small number of cameras, each of them seing most of the chamber volume while in long chambers only part of the cameras are useful in measuring a particular event.

In the first place most groups had to abandon, not without reluctance, the familiar large glass windows either because they were too costly to make or because, in the hydrogen case the necessary holes in the magnet would produce important magnetic field losses. Most groups have adopted retrodirective bright field illumination using scotchlite and fish-eye optics.

Chamber Illumination[x/]

A bubble can be seen either because it scatters light out of a beam and appears dark against a clear background (bright field) or because the scattered light is caught by a camera placed in such a way as not to receive the direct light, (dark field).

[x/] An extensive review of illumination and photography in bubble chambers has been written by W.T.Welford.Bubble and Spark Chambers, edited by P.R.Shutt vol. I - p. 233 (1967).

Scotchlite is a reflective sheeting made of small spheres of highly refractive material with reflective coating at the back; it is readily seen in figure a that the condition for parallel incoming and outgoing rays is: $n_2 = 2 n_1$.

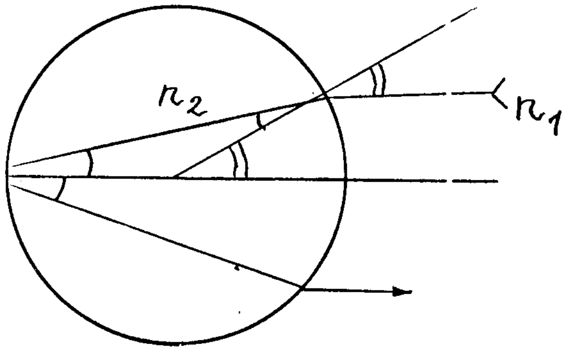


Fig. a

In practice most of the outgoing light is scattered within an angle of ± 1°, however extinction is not complete at higher angles and the remaining light level is usually considered too high for dark field illumination.

In bright field illumination a large fraction of the light is retroreflected and little flash power is needed; it is then possible to gain in resolution using slow emulsion with fine grain. The main problem is contrast: the image of a point bubble has the well known Airy diffraction pattern. However most points give wider distributions, due to the large focal depth. In Mirabelle the dimension of such Airy discs, reprojected in space for points at the limit of the nominal focal depth, is of the order of 700μ. Now provided the bubble stays small enough not be resolved, the diffraction pattern is independent of bubble size while the contrast is proportional to bubble surface. For the sake of contrast it is therefore important to produce large bubbles, this in turn may lead to difficulties in operating the chamber and to additional uncertainties as we shall see later.

Camera Optics

Wide field angle considerations have currently led to the adoption of spherical portholes or fish eyes projecting into the chamber liquid.

The entrance pupil of the objective sits at the center of the fish eye window. In the ideal case of a thin lens, the angle of the light ray with the optical axis reaches large values; this would impose severe tolerances on film flatness. Instead one uses a telecentric lens where all light rays fall almost perpendicularly on the film. For this purpose Mirabelle is using a special elliptic lens as shown in figure I, but the position and shape of the entrance pupil now varies with the inclination of the light ray.

Optical Errors

The fish eye - telecentric lens system is fixed to the chamber body while the hot components following the entrance pupil are part of the camera assembly; deformations during expansions may occur. Other mentioned sources of errors have

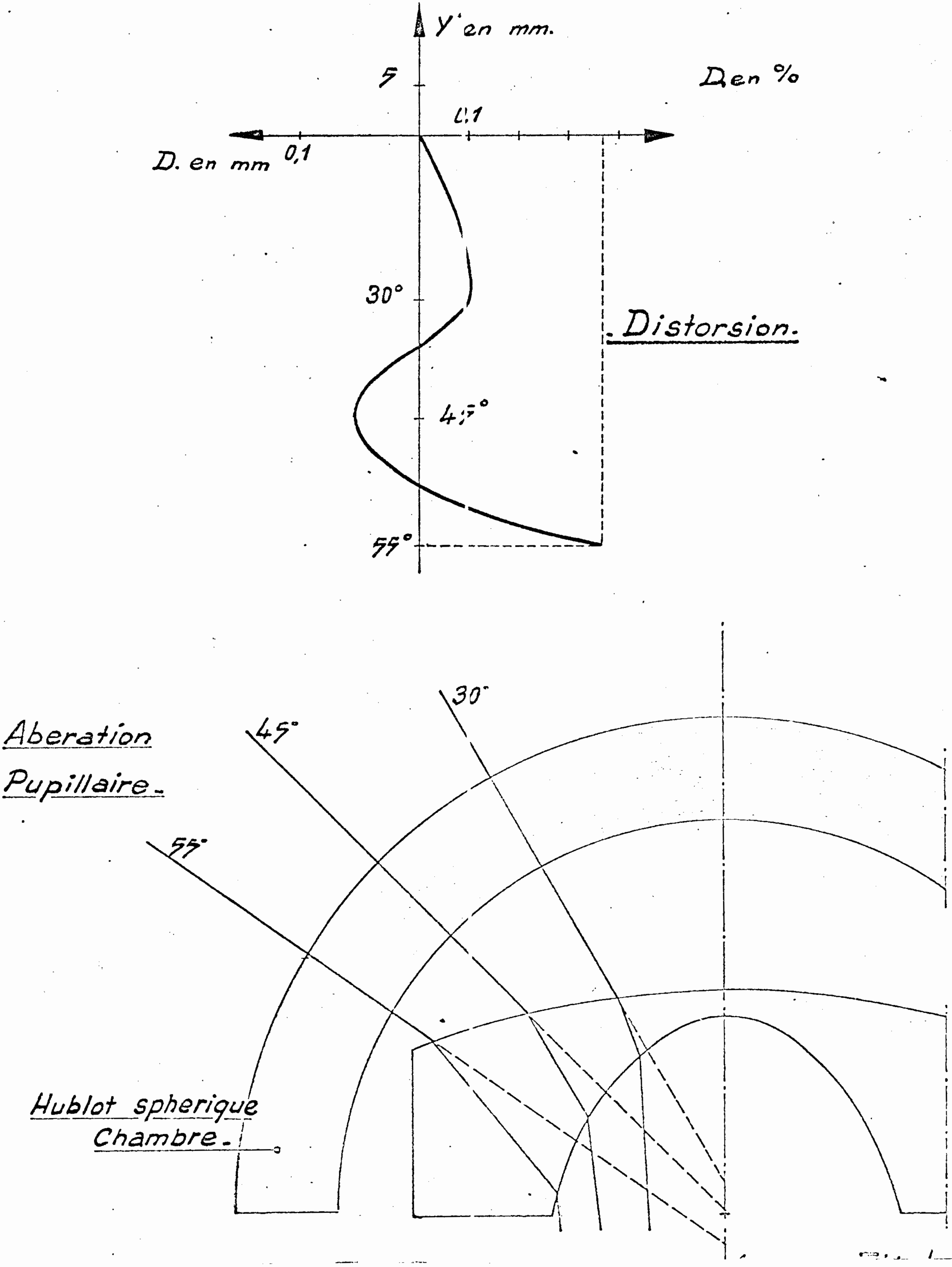


Fig. I. Configuration of the Mirabelle special elliptic lens.

been refractive index variations due to non unifirm temperature gradients in the glass, distortion of fish eyes due to chamber pressure variations, film flatness and film distorsion. The sum of these effects has been estimated to limit bubble relocation in the Argonne chamber to about 100μ.

Mirabelle optics are shown in figure 2. Two points are worth mentioning here: first, the cold part of the optics sees the hot part under a very small angle, thereby reducing temperature gradients in the glass; second whole objective turns around the entrance pupil in order to minimize errors due to deformations of the expanding chamber.

Optical Distortions

The appearance of events on film is most sensitive to the orientation of the optical axes. In many projects such as CERN, Argonne, Gargamelle, etc. the optical axes are not parallel and therefore pictures are rather different from what we are used to. The scanning difficulties exist in any case and do not seem to be greatly increased by a reasonable amount of barrel distortion; it has even claimed that this was a good thing as equal volumes of hydrogen could be contained in the same solid angle and information was more evenly spread on the film.

In Mirabelle we have parallel axes and pictures will be very similar to what we are used to; this appears to be a good argument to retain distortion-free optics. In any case this is essentially a scanning problem as such distorsions are taken into account in the reconstruction programs.

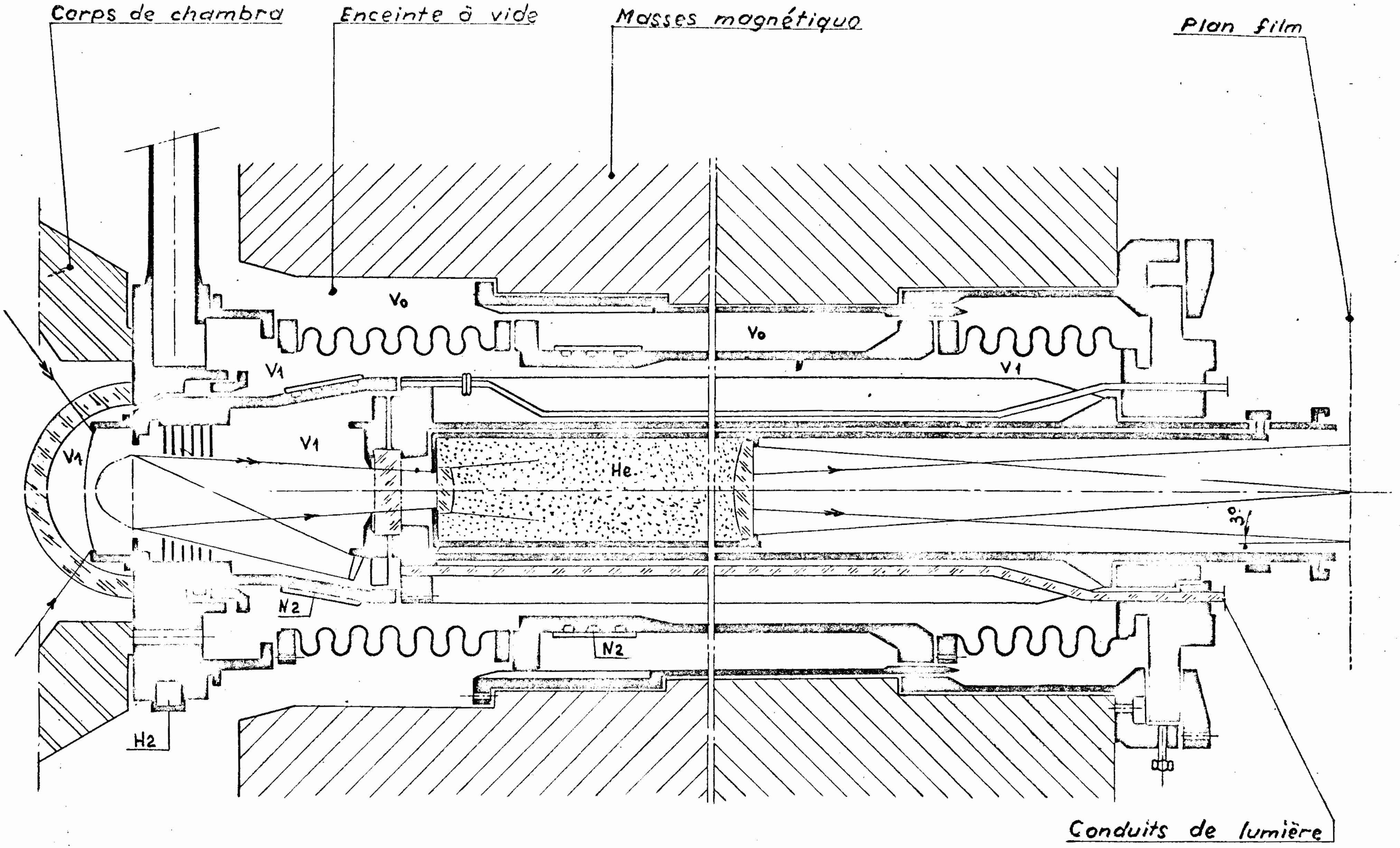


Fig. 2. The project of the Mirabelle.

SOURCES OF ERRORS IN IARGE CHAMBERS

Thermal Turbulences

Thermal turbulences in the chamber liquid affect the measurements accuracy in two ways.

-Physical bubbles will be displaced between the time they are created and the time they are photographed

-Light rays traversing the liquid will be deflected by random refractive index variations resulting in apparent displacement of the bubble image.

In the chamber, heat is produced in two main ways: first irreversible work appears during the expansion and recompression cycle and, second heat transfers take place between the liquid, heat exchangers and chamber walls. Liquid hydrogens is a poor heat conductor with a very low viscosity; under these conditions, heat is almost entirely transported by the convection of small eddies.

The dynamical temperature distribution in the chamber cannot be accurately predicted and theoretical evaluations /I,2/ have been made only for idealised chambers with a simple heat flow pattern.

Reinhard/2/ has considered a cylindrical vessel with the same dimensions as the European Bubble Chamber: the vertical walls are perfectly insulating; the top and bottom surfaces are kept at a uniform and constant temperature, their difference being ΔT ; heat created in the liquid is not considered.

This number ΔT is not very significant, since the temperature distribution in real chambers is rather far from the simple one of the model; however it is related to the heat flux Q by

$$\Delta T = \left(\frac{Q_{watts}}{2,3 \cdot 10^3} \right)^{3/4}$$

The temperature fluctuations and eddy sizes are functions of Q, it is assumed that the heat flux is more important than the details of the temperature distribution as far as distortions are concerned.

The eddy size and velocity are then respectively of the order of 2.5 mm and I00 microns per millisecond (table 3). A reasonable flash delay for a big chamber is $5 \mu s$, as big bubbles are needed for bright field illumination. The corresponding laternal displacement of 50μ is rather smaller than the measurement error.

A more serious effect is due to statistical deflection of the light rays by the eddies. The error angle is roughtly proportional to the heat flux transported by turbulent convection through the visible volume. The corresponding error in space for fish eyes optics is:

$$\Delta = \sim \sigma^{3/2} . Q^{7/8}$$

σ is the total path in the turbulent medium. Table II summarizes the results of Reinhard's calculations for two values of the heat flux Q and three different horizontal planes in the chamber, the beam plane (σ = 215 cm) and the extreme planes σ = 115 (top) and σ = 315 cm (bottom). Δ_{tot} includes both refraction and displacement errors; it should not be directly compared to the pointing error ε , at least when the track is measured by means of an H.P.D. In the latter case the master point transmitted to the geometry program is deduced from "hits" detected in some 20 consecutive scans with an average of one hit every two scan lines for a minimum ionizing track ($L \sim$ 2 cm).

Let us now consider the effect of a simplified eddy considered as a sphere of radius a and uniform temperature T + Δ T floating in a liquid of temperature T. The track length affected by a single eddy is

$$\ell = \frac{a \cdot \sigma}{d}$$

d and σ being the distances of the camera to the eddy and to the track. We shall also figure the effect of this eddy as a global displacement of an element of length ℓ; the total segment of length L contains

$$\frac{L}{\ell} = \frac{L d}{a \sigma}$$

such elements; assuming the pointing error to be negligible compared to Δ and each element ℓ to contain at least one hit, Δ is to be divided by a smooting factor $\sqrt{\frac{L}{\ell}}$.

In the CERN chamber or Mirabelle this factor varies from three for an eddy sitting on the track to about one for an eddy close to the fish-eye lens. The latter estimate is in fact pessimistic and deeper eddies will cancel partly the laternal error introduced by the first one. It is hoped that the smoothing factor will usually approach its maximum value. The turbulence error in the CERN chamber would then be of the order of 3 mm for Q = 100 watts. Under the same conditions and because of the smaller distance between cameras and bubbles Mirabelle would be some $\sqrt{2}$ times more accurate. This also points to the necessity of using the relevant cameras for accurate measurement of each track. This "smoothed" Δ is now to be compared to a pointing error δ of about 0,1 mm in space. If one takes the above figures seriously the orevall measurement accuracy will be severely limited by the turbulence effect. However they are based on an idealized model; it may well be that the heat flux in the visible region of a carefully designed is much smaller than 100 watts. Unfortunately no large chamber has been operated at a normal expansion rate so far, and the magnitude of the effect remains unknown.

Other Thermal Effects

Steady temperature gradients in the liquid will also cause refractive index variations; the error seems to be a minor one as Thomas/3/ and Turner/4/ mention figures of the order of 7 μ. Also, because heat production is a volume effect while adsorption takes place through the walls, convective motion may have to be considered in large chambers where the surface over volume ratio may be smaller than in present ones; however it is felt that the relevant velocities will be small.

Displacement of Bubbles During Expansion

Let us consider a simple cylindrical model with a piston at the top (fig. b)

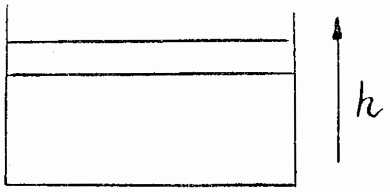


fig. b

During the expansion the piston moves up about 0.0I h, and a bubble at the center of the chamber will be displaced by $5\ 10^{-3}$h (~ I cm). Now, if one sets beam arrival and flash symmetrically with reference to the time of minimum pressure in the chamber, the liquid elements will be found in the same position at both instants. This would not be true any more if the above conditions is not exactly fulfilled. Moreover due to finite velocity of sound (~ I000 m/sec) the pressure is not uniform in the liquid and tracks will appear compressed or expanded along the direction in which the expansion takes place. Of course proposed chambersare very different from this simplified model, and proper estimates have to wait for experimental results. Note also that optical distorsion due to pressure gradient have been estimated by the British group to yield an error of 60μ .

As far as real chambers are concerned the above errors clearly increase with the expansion rate and flash delay. One should then compromise between the low temperature operation and long flash delay required to grow large bubbles and the low expansion rate and small flash delay required to minimize experimental errors.

Conclusion

Most estimates mentioned so far are largely uncertain or at least over simplified; in the present state of the art they may be far from the actual values. However the two main troubles are expected to come from heat fluxes in the chamber and from bubble displacements connected with the expansion process. Table III is the result of a study by L.Turner on the Argonne chamber. In the following a measurement error $\varepsilon = 300\mu$ will be assumed.

SECTION 2. EXPECTED RESULTS

In the "low energy" region a large chamber would overcome some limitations of current bubble chamber techniques: these are

a) the lack of statistics,

b) the difficulty of obtaining unbiased samples of events in one channel without a background of wrongly identified events,

c) too big errors on invariant masses resulting in bad separation between close resonances.

In present strong interaction physics, the absence of a resonance in a definite mass is often as important as evidence for some new state. Hence any progress among these lines is of great importance. It is hoped that large chambers will yield more accurate information when exposed to present accelerators.

As the incident energy increases, the particle energy becomes less and less sensitive to its rest mass.

Ambiguities between different reactions may be serious, the more so as bubble density no longer helps. Present chambers are certainly to cope with high energy reactions as they are currently doing with "low energy" ones. We now ask the question: will new chambers be good enough to select reactions on the basis of kinematics alone? Can they discriminate between Λ and K ? More generally, what can we measure at high energies? Some of these questions have been discussed by Fischer/I/ and Trilling/6/.

We now summarize our tentative conclusions, based on rather crude assumptions and approximations:

- Neutral particles such as Λ, K^o and γ will be measured nearly as accurately as charged particles produced at the main vertex when an observed two body decay is observed.
- Charged decays will not be easily detected for momenta over 7 GeV/c: pattern recognition techniques may possibly improve this limit.
- Charged tracks will often scatter before leaving the chamber and cannot always be measured up to the optimum length.

- When the nature of all tracks is known the fitting procedure will efficiently select events without neutral particles; it will not be nearly so good in comparing hypotheses where two tracks of comparable momentum exchange their identities. Bubble density will not help very much at high energies; in any case big chambers are not well adapted to ionisation measurements.
- One constraints fits will suffer from ambiguities; in some cases interactions or decays will be observed on some tracks, which may solve some of these ambiguities. The role played by IC fits in very high energy physics will clearly be less important than it is now, but no general statement can be made, as the number of ambiguous fits depends very much on the nature of the reaction.

We have only mentioned γ detection - namely introduction of plates at the far end of the chamber. This is a specific detector for neutral pions and gamma rays but momentum measurements are not expected to be very good.

Let us conclude with a word of caution: these estimates apply to a chamber completely free of systematic errors or distortions, and a perfect knowledge of optical constants is assumed. The errors in the Programs are also supposed to be exact and perfectly propagated. Any deviation from these conditions will result in less optimistic predictions.

References

I. D.B.Thomas. CERN Report 67-25, p. 2I5.

3. H.P.Reinhard. CERN /TC/ BEBC 66-62.

4. L.Turner, quoted by M.Derrick report AP/DC/HPC 9, p. 36.

5. C.M.Fisher. Rutherford Laboratory Reports. PHEL/S/IOI p. 5 and AP/DS/HFC/9, p.40. Heidelberg Colloquium Report. CERN 67-26, p. 25.

6. G.Trilling. SLAC Report SE, p. 40 (I962).

7. G.Rabinowitz. Proceedings of the Columbia Conference on Programming for Flying Spot Devices. New York, I966.

Table I

Main Parameters of Giant Chambers

	I2 foot	I4 foot	3.5 m	5 m	HFC	Mirabelle	Gargamelle	Wisconsin	SKAT
Laboratory	Argonne	Brookhaven	CERN	Dubna	RHEL	Saclay	Saclay	"	Serpukhov
size meters	4x2	4.25x5	3.7x2	5x3	I.8x2	4.5xI.6	4.8xI.9	4.35xI.7	4.5xI.5
visible vol. cubic meters	24	46	20		2.5	6	I0	6	6
Liquid		Hydrogen and Deuterium					Heavy Liquid		
Magnetic field		Supra Conducting					Classical		
H Kg	20	30	35	30-40	70	20	I9	I9	25
Cameras	4	5	5	4	4	8	8	8	4
Lens Openning Angle	I20°	90°		90°	I00-I40°	I40°	II0°	II0°	80°

The film width is always 70 mm.

Table II

		Top	Beam	Bottom
Q = IOO W				
ΔT = 2.3 . IO^{-3}	$\sqrt{\bar{\alpha}^2}$ rad	0.35	0.48	0.58
α = 2.7 mm	Δ refr.	0.40.	I.03	I.83
V_{eddy} = 92 /ms	Δ total	0.6I	I.I3	I.89
	p max	547	292	I75
Q = 250 W				
ΔT = 4.6. IO^{-3}	$\sqrt{\bar{\alpha}^2}$ rad	0.8I	I.IO	I.33
= 2.I	Δ refr.	0.93	2.26	4.20
V_{eddy} = I3I /ms	Δ total	I.05	2.35	4.25
	p max	354	I56	86

Relevant Results of the Argonne Summer Study

Table III

Limits on Bubble Relocation

Source	Assumptions	Deviation (microns)
Turbulence optical	200 cm distance, IO^{-3}Watt/cm^2 heat flux	I00 to 380
random velocity motion during flash	2 msec flash delay	I60
	30 msec period, 2 msec flash delay, 0.2 msec flash duration	I70
Temperature Gradient	ΔT = 2 millidegree	7
Pressure Gradient	p = 3.3 psi	44
Fisheye Windows		
misalignment	I00 misalignment	I00
refractive index	$\Delta T = 4^{o}$/cm; $dn/dT=5xIO^{-6}deg^{-I}$	74
change of shape	$\alpha = I.6xIO^{-6}deg^{-I}$	4
Vibration	200 sec flash duration	I6
Film Distortion	$I\mu$ on film	60
Film Flatness	flat to within 20μ	40
Total Deviation	added in quadrature	300 to 470

6000 Series SCOPE at CERN

H. Lipps[x)]

CERN has a CDC 6600 computer since January 1965 and a CDC 6400 since April 1967. The present note outlines the logical organisation of these computers and of the operating system which is currently in use at CERN. Only the basic facilities of the operating system are described, and these only in general terms. No attempt is made to cover all the details required by a user of the system; instead, the presentation is restricted to the general concepts used. In a few places the methods used to implement the system are outlined.

The operating system used at CERN is a major revision of the CDC operating system SCOPE version 2.0. The amendments include an automatic method of magnetic tape assignment and tape label checking, a completely new set of input/output routines, new programs to control the slow peripheral equipment, and a considerable number of modifications to job and equipment scheduling.

6000 COMPUTER SYSTEM ARCHITECTURE

The Main Frame

The main frames of the 6400 and 6600 computers are compatible from the system programmer's point of view. However, the 6600 at CERN has 128K of memory whereas the 6400 has only 64K of memory. Moreover, the central processor of the 6600 is several times faster than the central processor of the 6400, firstly, because instruction execution times are shorter, and secondly, because several instructions may be executed concurrently on the 6600. Both computers have a core store with a 1 microsecond memory cycle time. Each bank of core contains 4K words and all banks are interleaved. Thus, on the 6600 any 32 consecutive addresses belong to different banks of core, whereas on the 6400 any 16 consecutive addresses belong to different banks. Each central memory word consists of 60 bits.

The main frames of the two computers consist of the following units (See Figure 1) :

1 Central Processor
1 Central Memory
10 Peripheral Processors
12 Data Channels
1 Clock Channel

The central processor performs a set of arithmetical and logical operations under control of instructions which are held in central memory. It may reference central memory by means of instructions which either load individual words from central memory into working registers or store the contents of some working register into a central memory word. Unlike other computers the central processor has no instructions to initiate or control input/output operations. All input/output operations must be performed by the peripheral processors. The central processor communicates its input/output requests to the peripheral processors by placing appropriate information into central memory words which are monitored periodically by some peripheral processor. The conventions which the central processor must follow to communicate with the peripheral processors via central memory depend entirely on the operating system which controls the computer.

[x)] CERN

Each of the peripheral processors has its own memory of 4K words of 12 bits, called "peripheral memory". The program which is executed by a peripheral processor is held in its peripheral memory. This memory cannot be addressed by any other peripheral processor, nor can it be referenced by the central processor. The peripheral processors communicate with one another via tables and flags held in central memory. Each peripheral processor can also refer to all channels. Thus, it can input information from any peripheral device reading either single (12-bit) words into its accumulator or blocks of words into its peripheral memory. The block transfer terminates when either the specified word count is reached or the channel is disconnected by the controller of the peripheral device - which ever occurs first. For example, a magnetic tape controller would terminate an input operation on account of end of physical record, even if the word count specified for the input instruction is larger than the number of words contained in the physical record. Output from the peripheral processor to a peripheral device follows the same principles. All block transfers are unbuffered, i.e. the peripheral processor is tied to the block transfer instruction until the transfer is complete.

Other peripheral processor instructions transmit single 60-bit words or blocks of words between the central memory and the peripheral memory. There is a repertoire of logical instructions, fixed point additions and subtractions, shift instructions, and channel control instructions which make each peripheral processor appear like an independent small but fast general purpose computer. However, the peripheral processors do not, in fact, execute user programs, or parts thereof, because any programming errors could have catastrophic consequences on the overall system performance. Moreover, experience has shown that the ten peripheral processors available in the system have only a limited amount of spare capacity left after having been assigned to execute all system monitor functions and all input/output tasks, including some functions such as character conversion, packing and unpacking of data, and the like.

Central Processor Programming

The central processor has 24 addressable registers, called X0, X1, ..., X7, A0, ..., A7, and B0, ..., B7. (See Figure 2). The X-registers are 60 bits long and may be considered as general purpose accumulators. There are instructions which use any two of the X-registers as operands of an arithmetical or logical operation; the result is again stored in some X-register.

The A- and B-registers are only 18 bits long. The B-registers generally serve as index registers. (B0 is always zero).

A range of instructions is available to transmit a constant, the contents of an A- or B-register, or the lower 18 bits of an X-register, or the sum or difference of such quantities to an A-, B-, or X-register.

The A-registers, with the exception of A0, are used as address registers for references to central memory : Whenever an instruction places its result into register A1, this automatically implies that a word is fetched from central memory and placed into X1. The central memory address of the word loaded into X1 is determined by the new contents of A1. As X1 is associated with A1, so is X2 with A2, X3 with A3, and so on. However, while setting of A1, A2, ..., A5 implies transfers from memory to X1, ..., X5 respectively, does setting of A6 or A7 imply a store operation from X6 or X7 to central memory.

For example, to evaluate the expression

$$Z = U^2 + V^2$$

the following sequence of central processor instructions might be used :

```
A1 = address of U
A2 = address of V
X6 = X1 * X1
X0 = X2 * X2
X6 = X6 + X0
A6 = address of Z
```

The somewhat peculiar instruction set of the 6000 series computers is motivated by the attempt to increase the performance of the 6600 by a maximum of overlap between instructions. For example, the two memory fetches at the beginning of the above sequence may be initiated during consecutive machine cycles, i.e. within 100 nanoseconds. Unless the operands U and V happen to be stored in the same bank of memory, the two fetches will overlap in time. In addition, the 6600 has two multiplication units, permitting the two consecutive multiplications to be executed almost concurrently. An elaborate reservation control unit in the 6600 ensures that logically dependent operations are executed in their proper sequence. The programmer using assembly language may use his skill to choose registers and instruction sequences efficiently and thus save central processor time, but he need not care how the reservation control unit of the 6600 will achieve the overlap. The final result will be the same as if the instructions would have been executed in strict sequential order. The 6400 computer does not overlap instructions, but performs them one by one.

Typical instruction times are (in microseconds)

Operation	6400	6600
Floating point addition	1.1	0.4
subtraction	1.1	0.4
multiplication	5.7	1.0
division	5.6	2.9
Fixed point addition	0.6	0.3
Shift	0.6	0.3
Logical product	0.5	0.3
Transfer of control	1.3	08.-1.5

There are four additional 18-bit registers in the central processor : An instruction address register (P-register) contains the address of the instruction currently executed. Two registers (RA-and FL-register) are used for memory protection and relocation, and one (EM-register) holds indicator bits controlling the "exit mode" of the program. (See below). The RA-,FL-, and EM-registers are not under control of the programmer, but are set by the operating system.

At any given time several independent programs may be held in central memory. Each one will occupy a contiguous area of the store. The first address of a program is called the reference address of the program.

While the central processor executes the program, its reference address is contained in the RA-register. All addresses in the program are relative to its reference address; e.g. the reference address itself is referred to by the program as program address zero. During execution of all central processor instructions which refer to central memory the contents of the RA-register is automatically added to the address specified in the program. This feature permits the operating system to interrupt a program at any time during execution, move the program's area to a different section of the store, adjust the RA-register accordingly, and continue program execution without any need to adjust explicitly all addresses specified in the program.

The number of central memory words allocated to a program is called its field length. It is held in the FL-register. If the program uses an address which is larger than the contents of the FL-register, then the corresponding operation will not be performed. Whether the program is allowed to continue depends on the contents of the EM-register. If the error is to be treated as fatal, then a stop instruction is automatically written into location 0 of the program, and the P-register is set to zero, thus stopping the central processor in location 0.

One of the peripheral processors executes a supervisory program which monitors the central processor operation. To do this, the peripheral processor instruction set includes two special instructions. One reads the current contents of the central processor P-register into the accumulator of the peripheral processor. If P = 0, the monitor program assumes that an error condition has interrupted the normal flow of the program.

A peripheral processor may execute an "exchange jump" instruction to switch the central processor from one program to another. This instruction specifies an address in central memory. A block of 16 words, called the "exchange package" will be stored there. When the exchange jump instruction is executed by a peripheral processor, the central processor stops as soon as all instructions are executed which are held in the location currently addressed by the P-register. Then the current contents of all registers shown in Figure 2 are stored in the exchange package while the previous contents of the exchange package is entered into the registers. The central processor will then continue to execute instructions in accordance with the new contents of its registers. For example RA + P is the address from which the first instruction will be taken. If, later on, another exchange jump specifies the same exchange package (and assuming that the contents of the exchange package has not been changed in the meantime) then the program which has been interrupted by means of the first exchange jump will continue execution as if no interruption had occurred.

Peripheral Equipment

At present the following peripheral devices are connected to the 6600 computer at CERN:

1 model 6603 disk unit (7.5 million words)
one half of a 6638 disk unit ($\frac{1}{2}$*15 million words)
10 model 626 magnetic tape units (240 KC)
6 model 607 magnetic tape units (≤ 120 KC)
1 model 6602 display console
1 model 405 card reader (1200 CPM)
1 model 415 card punch (240 CPM)
2 model 501 line printers (≤ 1000 LPM)

1 Calcomp plotter
7 Teletype printers
3 Flying spot digitizers

The 6400 differs from the 6600 in as much as it has no flying spot digitizers and no Calcomp plotter, and only 2 Teletype printers, 4 tapes of each kind, but a model 863 drum unit.

THE 6000 SCOPE OPERATING SYSTEM

All jobs run at CERN operate under control of the SCOPE operating system. The main functions of SCOPE can be summarized as follows :

System initialisation and recovery
Communication with the operator
Control of off-line operations
Job initiation
Resource allocation
Loading of programs
Input/output processing
Error detection
Program recovery
Job termination
Accounting
Provision of a program library.

System Initialisation and Recovery

To initialize SCOPE, the operator must perform a "dead start" operation. A library of programs is thus loaded from magnetic tape and stored partly in the core memories of the main frame and partly on direct access storage devices.

From then on the operator can communicate with SCOPE via the two display screens and the keyboard of the display console. Options available to him include

Changes to peripheral equipment status
Control of off-line operations
Selection of job initiators
Selection of system displays
Recovery of system after a breakdown

About 10 000 words of central memory are reserved for exclusive use of SCOPE. This area, called "Central Memory Resident" contains the most frequently used control routines and all tables used by SCOPE.

In particular Central Memory Resident contains seven blocks of 192 words each, called "control point areas". SCOPE reserves a control point area for a job as soon as it begins to execute the job. Status information pertaining to the job is recorded in its control point area and used to control the progress of the job. Only when execution of a job terminates, is the control point area cleared for re-assignment to another job. As SCOPE has seven control point areas, no more than seven jobs may be executed concurrently.

The control point concept is fundamental to the logical structure and operation of SCOPE. The multiprogramming system may be visualized as a set of seven virtual computers, "control points", each one being able to execute one job at a time. The control points share the use of the central memory, the peripheral devices, etc. on a dynamic basis. For example, each job which is being executed occupies a contiguous block of central memory. The first address ("reference address") of that block and its length ("field length") are recorded in the control point area assigned to the job.

Categories of Programs

There are five categories of programs within the SCOPE library :

System control programs

Job control programs

System service programs

User service programs.

User callable subprograms

The system control programs are all loaded into core memory (central memory or peripheral memory) at the beginning of system initialisation, and they remain there permanently. These programs are responsible for the coordination of the activities of all control points and for the interaction between the system and the console operator. There are five such programs :

The System Monitor program

The Central Memory Move program

The Idle program

The Peripheral Processor Resident program

The System Display program

The System Monitor program is loaded into peripheral processor 0. Its main functions are :

Allocation of the central processor to a control point

Monitoring of central processor programs for I/O requests

Allocation of central memory to control points.

Allocation of peripheral processors 1-8 to control points

Reservation of peripheral equipment needed by control points

Reservation of channels and system tables on demand of peripheral processors

Maintenance of a real-time clock

Maintenance of an accounting file

The System Monitor uses the Central Memory Move program to move the contents of central memory areas assigned to control points. (See Figure 3).

The Idle Program is a dummy central processor program to which the central processor is assigned by the System Monitor program whenever none of the seven control points requires the central processor.

The System Display program is loaded into peripheral processor 9 at the time of the dead start operation. It remains there permanently and ensures the communication between SCOPE and the display console operator. Keyboard entries permit the operator to select a variety of different display formats to obtain appropriate information about the current status of the system. The operator may also change system status and performance by means of keyboard entries.

A copy of the Peripheral Processor Resident program is loaded into the lower locations of peripheral processors 1 to 8. It includes an idle loop which each of these peripheral processors executes as long as the System Monitor has no task assigned to the peripheral processor. The System Monitor program can assign any one of several job control programs to any one of the "pool processors" 1 to 8.

Job control programs monitor the execution of a particular job; thus they may be collectively described as the routines needed to control the monoprogramming system in charge of each control point. For example, the following tasks are performed by job control programs :

Selection of a new job for execution

Request of central memory for execution of the job

Request of peripheral devices, e.g. tape units

Loading and initiation of a user program

Termination of a user program

Release of user scratch files on disk

Termination of the job.

User service programs are central processor or peripheral processor programs which may be invoked by a user job which requires assistance from SCOPE. Calling appropriate peripheral processor programs the user program may e.g.

Perform input/output operations

Send messages to the console operator

Load overlays of programs

Request or release magnetic tape units

User service programs which are executed by the central processor may be called via appropriate control cards. They permit e.g.

Allocation of user files for off-line printing
or punching

Copying of files

File positioning

Request or release of magnetic tape units

Compilation of FORTRAN programs

Translation of assembly programs

System service programs are central processor programs almost like user programs, but performing functions specially requested by the console operator, e.g.

Input of jobs on the card reader

Printing of results

Punching of cards

Plotting of graphs

Dumping of files on magnetic tape

Dumping of accounting information

The operator may initiate the system service program which he requires, by assigning it to a free control point. Further operator commands are available to reserve the appropriate peripheral equipment for the execution of the system service program. During their execution the system service programs are controlled by the System Monitor program and the job control programs like any user program. However, the system service programs may request the services of peripheral processor programs which have access to tables in the Central Memory Resident area of SCOPE.

Utilisation of Central Memory

Approximately the first 10,000 words of central memory are reserved for the Central Memory Resident. This area contains

- Pointer to Tables
- Channel Status Table
- Peripheral Processor Communication Area
- Control Point Area
- Central Processor Control Programs
- Equipment Status Table
- Track Reservation Tables
- Library Directory
- Resident Library
- File Table
- Dayfile Buffer
- Teletype Buffers
- Tape Status Indicators

The Channel Status Table records for each data channel whether the channel is free or reserved for the use of a peripheral processor.

The Peripheral Processor Communication Area contains for each peripheral processor the name of the program and the control point number to which it is assigned. In addition, it contains any requests which the peripheral processor submits to the System Monitor program.

The Equipment Status Table specifies the type, status, channel number, and equipment address of each peripheral device, and also, for which control point the device has been reserved.

The disks and the drum are never reserved for a single control point. Instead, there is a Track Reservation Table for each of these devices which contains one bit for each track (approximately 4000 words). This bit is set to 1 while the track is reserved for some control point and reset to 0 when the track is free again.

The Library Directory contains the name of each library program and the location where it is stored. The most frequently used routines are held in the Resident Library (i.e. in central memory itself), the remaining programs are stored on the drum or on a disk unit.

The File Table contains one entry about the current status of each data set available either on disk or on magnetic tape. This entry specifies e.g. the peripheral device on which the data set resides, whether the data set belongs to a control point, and which is the most recently used disk address.

All accounting information is recorded in a system file, called the Dayfile, by the System Monitor program. The most recently used accounting messages are held in a circular buffer, the "Dayfile Buffer", in order to reduce the number of accesses to the disk and also to permit the System Display program to display these messages to the console operator.

Three small buffers are available for each Teletype station; one for the message being input on the keyboard of the station, one for the message output from the program for which the Teletype is reserved, and one for system messages which SCOPE may "broadcast" to all Teletype stations.

Associated with each magnetic tape unit is a small display unit which permits the display of a five-digit reel number and the required protect status for the reel. This display is used by SCOPE whenever the operator is requested to mount a new tape reel. SCOPE then monitors the tape unit and when it becomes ready automatically reads a tape label recorded at the beginning of the tape. If the correct tape has been mounted, the display of the reel number is cleared. Otherwise SCOPE will unload the tape and repeat the request for the tape reel. To control this "Tape Number Display System", SCOPE retains a small entry for each of the displays in the Tape Status Indicator area of Central Memory Resident.

The part of central memory which is not occupied by the Central Memory Resident is available for the use of the seven control points. Each active control point occupies a contiguous block of central memory. The sequence of these blocks corresponds to the control point numbers; the memory occupied by control point 1, if any, is adjacent to Central Memory Resident, and so on. Any unused memory space is located at the top of the store.

It is possible for a job to extend or reduce its central memory space by means of appropriate control cards. When the System Monitor program encounters such a request, it will wait until all input/output operations of the higher control points have been completed but not initiate any more such requests. When all peripheral processor activity at the higher control points has ceased, the System Monitor program will initiate the Central Memory Move program in order to move the central memory areas occupied by the higher control points in accordance with the memory space request issued by the control point which wishes to change its space allocation. When this has been done, normal processing resumes at all control points.

If a control point requests a reduction of its current memory assignment no problems arise and the operation will always be executed as just described. However, if a control point requests an increase of its current memory assignment, then the additional space may or may not be available. If the space required is not available the request must be delayed until some other control point releases enough of its space. It would thus be possible to generate a dead-lock in which no job can continue without an increase of its memory space which is bigger than what is currently free. To avoid these situations, an increase of memory space is always executed in two steps: first, the space currently available to the control point is released, and then the total space required to continue is claimed back. Thus, jobs waiting for an increase of their memory space will not occupy memory.

Control of Off-line Operations

Under normal operating conditions the console operator will assign a system service program to a control point which controls all slow peripheral devices. This program, called BATCHIO, performs the following operations automatically and concurrently:

Input of jobs from the card reader

Printing of results on all line printers

Punching of cards

Plotting of graphs

Control of all Teletype stations

Control of the Tape Reel Number Display System

The advantage of combining all these operations within a single program is that it leaves six control points for user jobs (or other kinds of system service programs) and requires only a single peripheral processor to control all slow peripherals.

BATCHIO monitors the card reader continuously, and when it becomes ready, it reads the job submitted by the operator and places it as a data set on a disk unit. An entry is made in the File Table which identifies the data set as a job waiting to be initiated, i.e. assigned to a control point. It also specifies the type of the job, its priority, and a sequence number which identifies the job uniquely.

Any data sets produced by a job for off-line printing, punching, or plotting are also recorded in the File Table. BATCHIO scans this table regularly, and when it finds such a data set, BATCHIO will output it to the appropriate device. Print files may be printed either in accordance with FORTRAN editing conventions or with a standard page layout; card files will be punched in one of three modes: Binary mode, Hollerith mode, or EBCDIC mode, depending on what the job producing the data set has specified.

Job Initiation

When SCOPE is initialized by a dead start operation, all control points are "unassigned", i.e. idle. To start operations, the operator assigns some or all control points to a system service program such as BATCHIO or to a "job scheduler". The latter is a job control program which scans the File Table for user jobs which are waiting to be executed.

Depending on the job scheduler which has been assigned to the control point by the console operator, only particular types of job will be selected for execution at that control point. The first card of each job, the "job card", specifies the job type by means of a one-letter code; L for long job, S for short job (i.e. requiring no more than one minute of central processor time), X for express job (i.e. reduced turn-around time), and so on.

Within the type or types of job which a job scheduler considers for execution, it will select the job which has the highest priority. If this choice is not unique, the job with the lowest sequence number, i.e. the job which has been submitted first, will be selected for execution.

The cards immediately following the job card constitute the "job control record". When the job is assigned to a control point by the job scheduler, the job control record is read into a portion of the Control Point Area associated with the control point. The job control record contains a sequence of control statements which are processed, one by one, by some job control program. Basically, there are two kinds of control statements: Job control statements which set various conditions for the further execution of the job, and loader control statements which load and initiate binary programs required by the job. Thus, a job may involve the execution of several private programs and library programs in succession, each program with its own requirements for memory space, equipment, conventions for error recovery, and so on. The programs of a job may share the use of a set of scratch files, i.e. data sets which are generated by the job and deleted automatically as soon as the job terminates execution.

Resource Allocation

A job may request and release central memory and peripheral devices, request off-line operations, and define limits on the use of central processor utilisation by means of various control statements. These are illustrated by the following typical examples:

```
SPACE (18000)
```

requests a total of 18000 words of central memory for further execution of the job. SCOPE permits a maximum space requirement of 46000 words if the job is executed on the 6400, and of 60000 words if the job is executed on the 6600 computer.

```
CPTIME (180)
```

sets a limit of 180 seconds for the use of the central processor by the job. This time limit applies to either the 6600 or the 6400 central processor, whichever executes the job. In contrast, the statement

```
64TIME (180)
```

specifies a time limit of 180 seconds only if executed on the 6400 computer, and is ignored on the 6600.

```
LINELIM (3000)
```

specifies that any of the user programs being executed may not output more than 3000 lines to the off-line print file.

```
PLOTLIM (1200)
```

restricts the output of a user program for off-line plotting to a limit of 1200 seconds of plotter time.

```
CONSOLE (41)
```

requests that the Teletype station which is identified as having logical equipment number 41 is reserved for use of the job.

```
TAPE (TAPE1, TAPE7, TAPE4)
```

specifies that the data sets named in parenthesis are to be located on magnetic tape rather than on disk. In this case the user must also supply additional information on the tape reels to be used. This information is contained on additional punched cards, called "Tape description cards". One tape description card is required to associate the name of one data set with a particular tape reel and specify the protect status and density of the tape reel.

Loading of Programs

Programs at CERN are usually written in FORTRAN, or alternatively in an assembly language. The complete program may be subdivided into several program units (subprograms) in accordance with established FORTRAN conventions. The FORTRAN compiler and the assembler translate one program unit at a time from its source language into a relocatable binary program. Once all program units (other than library subprograms) of an executable program are available in relocatable binary form, they may be loaded into central memory, linked together, and initiated by the SCOPE loader. Depending on the method of loading used, two types of executable program may be distinguished.

The simpler and more frequently used type of program has all its program units in central memory throughout the execution of the program. These programs are called "normal programs". Alternatively, the program units of an executable program may be grouped into sections ("overlays") which may replace one another in central memory during execution. Such programs are called "overlay programs".

Normal programs consist of exactly one main program and possibly one or more subprograms. If, after translation from source language, all relocatable binary programs are e.g. stored on some data set, the program may be loaded and initiated by a control statement such as

```
ALUSHTA.
```

where ALUSHTA is the name of the data set on which the relocatable binary programs are stored. In this situation the loader performs the following tasks in succession:

The relocatable binary programs are loaded and relocated one at a time. The loader establishes the necessary tables of cross references between program units in parallel with the loading process.

When all program units supplied by the user have been loaded and relocated, and program linkage is required, the loader will automatically search the SCOPE program library for missing subprograms. Those library programs required and found will be loaded.

Next, all relocatable binary programs which have been loaded will be linked together by means of the loader tables to form an executable program. Any unsatisfied cross references which exist at this stage will be filled with out-of-bounds memory addresses. Thus, the executable program may be executed even if some of the subprograms referred to have not been supplied. However, if any of the unsatisfied cross references is required during program execution, an arithmetic error condition will occur.

When the relocatable binary programs have been linked, a map of the memory area assigned to the job is provided. Finally, the loader will transfer control to the executable program.

One or more formal parameters may be defined for an executable program. These parameters appear as parameters of the main program. In particular, the formal parameters of a FORTRAN main program are the names of all data sets implied by the input/output statements used. For example, a FORTRAN program which refers to logical tape unit 13 would imply the use of a data set named TAPE13; FORTRAN statements which specify card reader, printer, or card punch operation imply the use of data sets named INPUT, OUTPUT, and PUNCH, respectively. The header statement of a FORTRAN main program must list the names of all data sets which the executable program implies; e.g.

```
PROGRAM DUBNA (TAPE13, OUTPUT, TAPE2)
```

If, for a particular execution of the program, some other data sets should be used with the program, these may be specified in the control statement which initiates execution of the program. For example, if the program itself is stored on the data set ALUSHTA in relocatable binary form, then the control statement

```
ALUSHTA (IGOR, PUNCH)
```

would cause program DUBNA and its subprograms to reference data set

IGOR instead of TAPE13

PUNCH instead of OUTPUT

and TAPE2 as before.

Overlay programs consist of exactly one "basic overlay" which is held in central memory throughout program execution, a number of "primary overlays" which replace one another under program control, and possibly some "secondary overlays". Each secondary overlay is associated with some primary overlay. Only one of the secondary overlays may be in memory at any given time, and only together with its associated primary overlay. (See Figure 4).

Each overlay consists of exactly one main program and possibly a number of subprograms. Once all program units of the overlay program have been translated into relocatable binary form and stored on some data set, the overlay program is initiated in two steps: Firstly, all program units of an overlay are loaded into central memory, linked together to an absolute program, and then the "absolute overlay' recorded on a data set. This step is first performed for the basic overlay, then for a primary overlay, then for each of the secondary overlays associated with this primary overlay, then for the next primary overlay, and so on, in order that the location of each overlay in memory can be properly determined. This process is called overlay generation. It is controlled by "overlay directives" which must be interspersed with the relocatable binary programs and by appropriate loader control statements.

Once the absolute overlays of an overlay program are available on a data set as the result of overlay generation, the program may be initiated by the same type of control statement as described above for normal loading. The formal parameters of the program must be specified in the main program of the basic overlay.

Any executable program which is available either in relocatable binary form if it is a normal program, or as a set of absolute overlays if it is an overlay program, may be added to the SCOPE library. In this case the name of the main program is recorded in the Library Directory of the Central Memory Resident together with the location of the program on disk or on drum. Any job may then reference the program by its name just as if this name would be the name of a data set on which the user had stored the library program. To revert to the example above: If program DUBNA is added to the SCOPE library, then the control statement

DUBNA (IGOR, PUNCH)

has the same effect as had

ALUSHTA (IGOR, PUNCH)

as long as the program was stored on data set ALUSHTA as the user's private program.

References :

1. Control Data Corporation, 6400/6500/6600 Computer Systems Reference Manual, Pub. No. 60100000, 1967.

2. Control Data Corporation, 6400/6500 Computer Systems SCOPE Reference Manual, Pub. No. 60173800, Sept. 1966.

3. CERN, CERN Computer 6000 Series Principles of Operation, 1.3.1968 (loose-leaf).

4. CERN, CERN Computer 6000 Series SCOPE General Reference (loose-leaf).

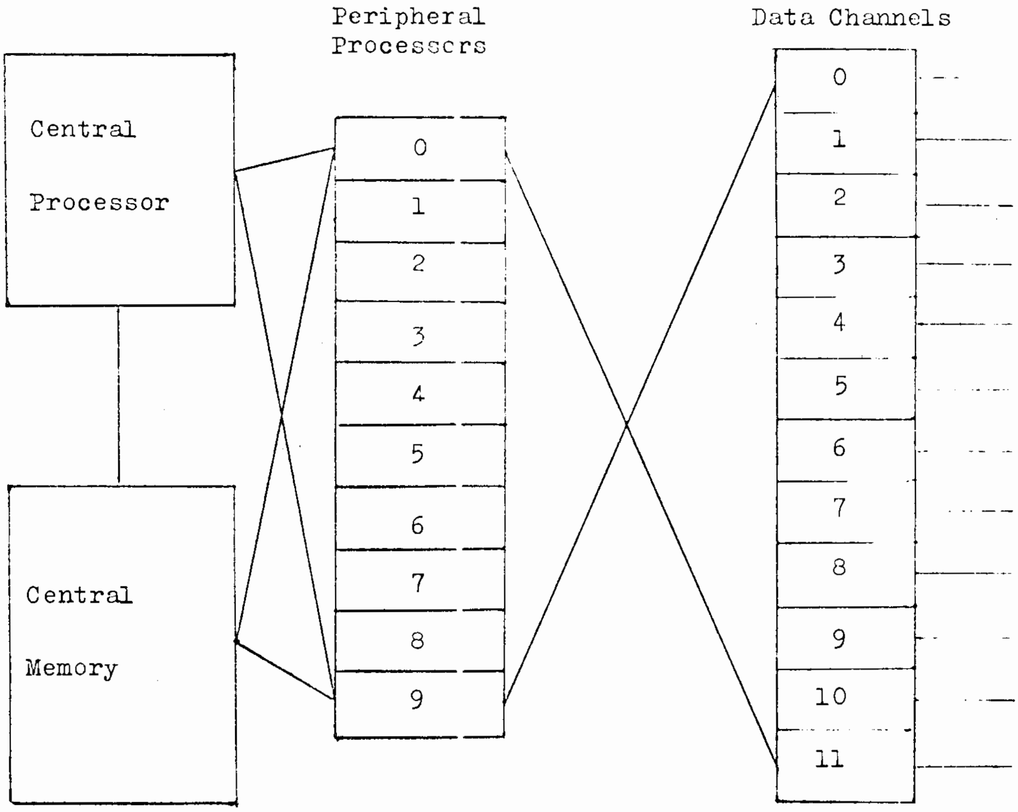


Figure 1 : 6600 and 6400 Main Frame

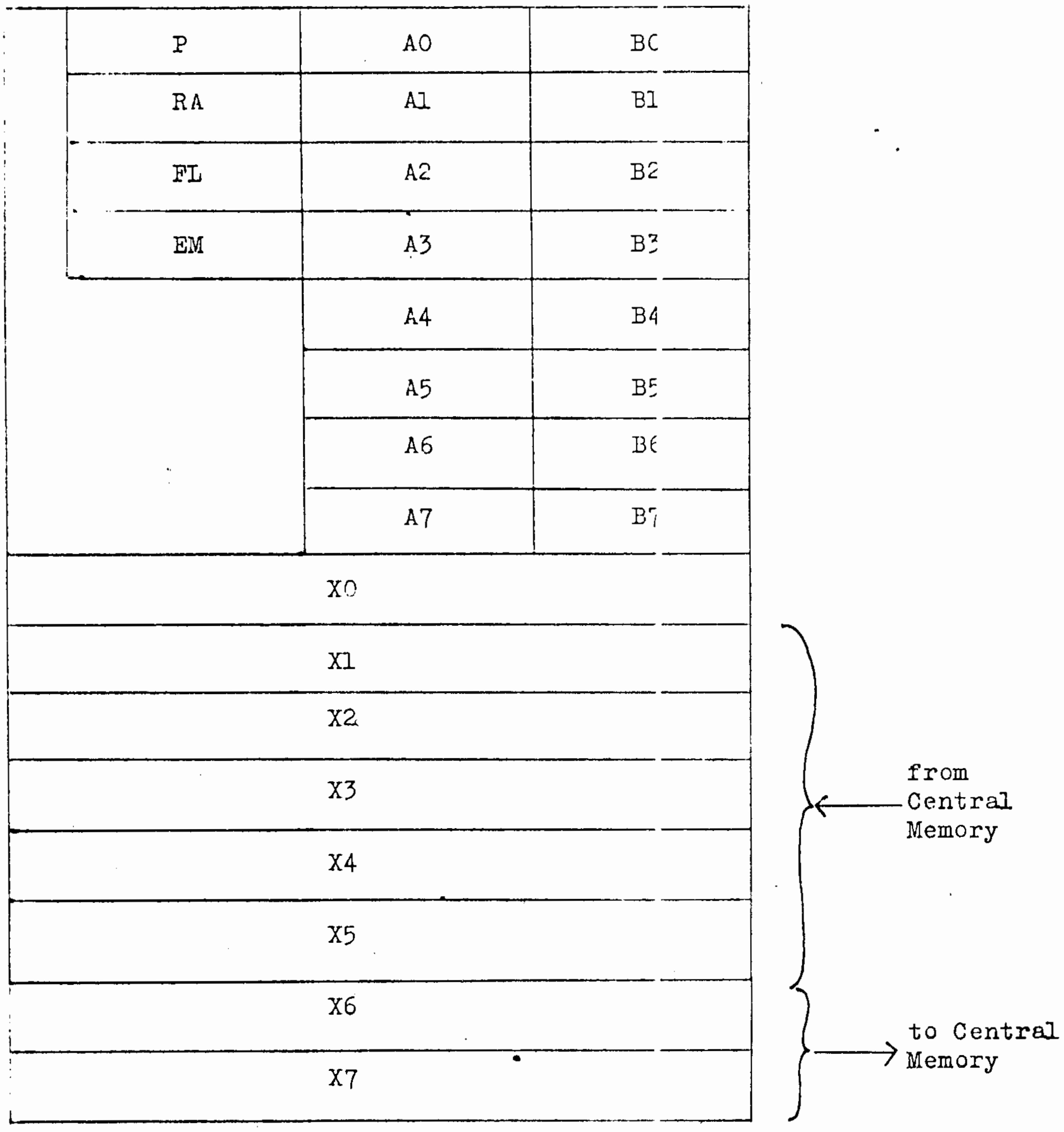


Figure 2 : Central Processor Registers

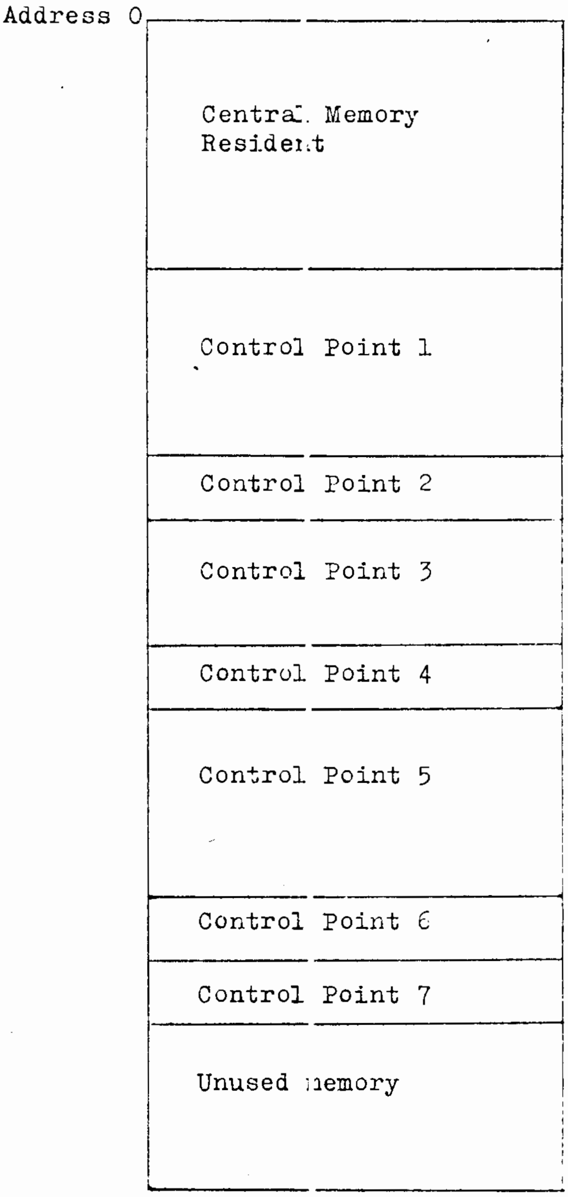


Figure 3 : Central Memory Utilisation

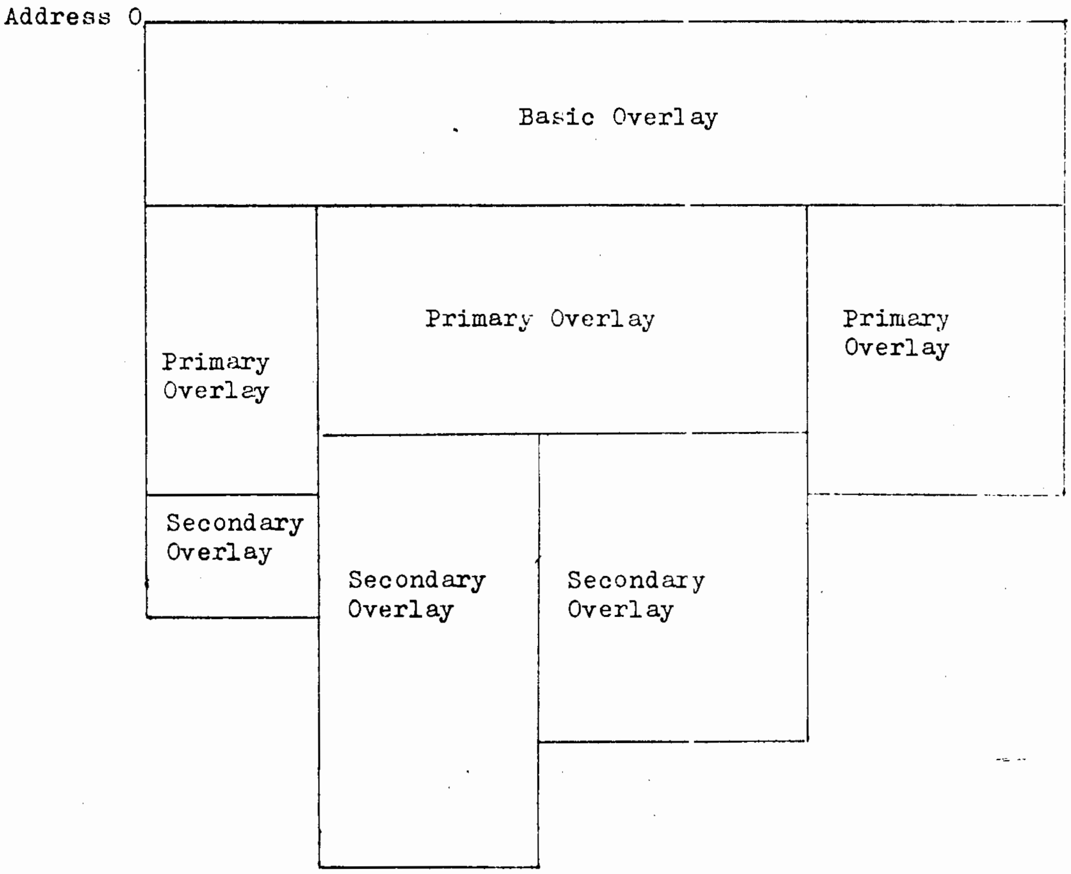


Figure 4 : Overlay Structure

Review of On-Line Computers and Their Uses in High Energy Physics

D. Lord x)

1. Introduction

Over recent years the use of on-line computers has developed and expanded to such an extent that they are now involved in almost every aspect of the execution and analysis of High Energy Physics experiments.

A reasonabley comprehensive list of the uses or applications of on-line computers would be :

Control of accelerators and/or storage rings.

Control of beam transport and "switch-yard" systems.

Control of Bubble Chambers.

As control and data acquisition systems for spectrometers.

As analysers in multiparameter counter experiments.

As the data acquisition and control sections of measuring systems for spark and bubble chamber photographs.

As data acquisition and recording systems for wire and sonic spark chamber experiments.*

As the interface and control between a number of computer systems xx).

In addition to these applications, there are a number of more general applications which effectively involve the user of on-line computers, such as : multi-console schemes for the preparation, modification and testing of programs; display systems to aid in the examination of results from the processing of data produced in high energy physics experiments. As this class of applications involves problems in computer technology (i.e. software as well as hardware) that are not specific to the field of High Energy Physics and in addition requires the use of medium to large computers, I will not be considering them further in this review.

Examining the above list of applications, it can be seen that they each involve all or some of a number of basic tasks for the on-line computer. These basic tasks are :

<u>Data Input (or Acquisition)</u> - The transfering directly into the computer of data from an experiment, or accelerator, or measuring device etc.

x) CERN

xx) As these applications are essentially covered in the two papers of Dr. Zanella, it is unnecessary for me to discuss them further in this paper.

Data Output (or Recording) - The writing of data on some recording medium such magnetic tape, magnetic disks, etc.; or the direct transmission over data-links to another and larger computer for further processing.

Validity Checks on the Data - The data read into the computer must be checked to see that it is reasonable, e.g. whether it lies within specified limits. In this way malfunctions or maladjustments of the system can be detected.

Data Selection - In applications where the input data represents measurements on interactions (events), it is often required that selection criteria be applied immediately to the data so that only events of specific types are passed on for further processing or recording.

Data Processing or Analysis - This is of course very dependent on the application. It may for example be the complete processing of a sample of the incoming events (the processing of the remainder being done off-line at a later date), to provide a set of more stringent tests for the data and a means of monitoring the progress of an experiment. In contrast it may merely be the addition of counts to memory locations representing channels in a multi-channel analyser.

Displays - Operator Communication - In most applications, it is essential to provide the operators of the system (here I include the experimental physicist) with information on the state of the system, the processing of the data etc. and the means of readily changing programs and program parameters.

Control - In many applications, the computer is required to control by means of digital words, control pulses, analogue signals etc., the equipment in the system e.g. the currents in magnets of a beam transport system; the movement of film from one photograph to the next in a measuring machine.

Before going on to discuss in more detail some of these applications and the tasks they involve, it is desirable to provide some background on the characteristics of the on-line computer and of the types of computers that are available to us at present.

2. Characteristics of on-line Computers

In general one thinks of on-line computers as being small - and in many applications this is indeed the case. In principle the size of the on-line computer required for a particular application depends on such factors as :

The peak and average input data rates.

The average data recording rates of the output medium or data links.

The number of data sources.

The precision with which the data is to be represented.

The amount of checking, selecting and analysing that must be done on the data, to provide the real time responses to the rest of the system and its operators.

The time scale in which these responses are required.

The complexity of the control that computer has to exercise over the system.

The complexity of the communications between the computer and the operator, number of displays, typewriters, special control panels etc.

The number of separate applications that are to be run simultaneously in the computer.

In addition to these considerations a number of other things have entered into the choice of on-line computers such as :

A desire to provide the computing power etc., represented by an on-line computer in the most economical way possible by using part of a very large general purpose computing facility; the necessity of using "what was available". Hence we find at CERN, three automatic film measuring machines on-line to the CDC 6600; at Brookhaven and Berkely similar measuring machines are connected on-line to IBM 7094's.

Therefore when attempting to choose the on-line computer for a project, it is necessary examine the various characteristics (memory size, channel structure, interrupt system, cost etc.) of the available computers in the light of the above factors, the basic tasks required and the project's budget.

I would like now to take the various parts of a computer and examine them from the point of view of what the computer manufacturers will now provide and what is desirable for on-line operation.

2.1 Main Memory

As this is usually one of the most expensive elements of a computer, then for economic reasons in most on-line systems, one wishes to keep the memory down to a reasonable size. So it is important to ensure that memory space is not unavoidabley "wasted" or used up in unexpected ways. Examples of how this can occur are :

a) If the precision with which the data must be represented is such that one data word has to be stored in two memory words, then this is evidently wasteful especially as it implies most arithmetic operations must be done in double precision - lengthening the program. Thus where ever possible it is desirable to match the data precision with the length of the computer word.

b) Computers with short words, have an addressing problem. For example a computer with a 12 bit word might only have 8 bits available for addressing, so an instruction can only address 256 words directly. Various methods are used to circumvent this problem such as double length instruction words, indirect addressing. All of these essentially "waste" memory space. Computers of only 12 bits have an addition problem, the full 12 bits can only address 4096 words, so if memories of large capacity are required, some device like "blank switching" must be adopted. Computers with 18 or 24 bit words usually avoid these problems. For many applications a 16 bit word probably represents a reasonable compromise between expense and wasted memory space.

c) System programs, library routines etc. frequently demand unexpectedly large amounts of core store, for example a set of floating point routines (for a machine with only fixed point arithmetic) can easily take 1.500 words; I/O routines for a disk 500 to 1.000 words and so on.

d) If the recording rate of the output devices (e.g. magnetic tape unit) chosen are inadequate, or the computing speed is inadequate, then it will be necessary to expand the various data buffers in the memory.

Thus memories with capacities between 4K* and 16 K words of 16 or 18 bits represent a reasonable choice for many applications. Memories with word lengths of 24 of 32 bits are expensive unless the computer has provisions in its instruction repertoire for working easily with half-words. It is also desirable to be able to expand or choose the memory capacities in reasonable increments of say 4096 words.

Memory speed is clearly related to the required input data rates and to some extent desired computational speed. Many small computers are available with memory cycle times between 800 nsec and 2 µsecs. To obtain extreme speeds, memories are available, which are organized in separate blocks whose accesses can be interleaved. Also if it is desired to reduce the interaction of input and output transfers on the Central Processor Unit (CPU), it is possible to have memories also organised into separate blocks, but with the additional facility that one block can be working with the CPU while another is operating independently with an I/O-channel.

Many computers are available now with memory protection in various forms. At one extreme, each memory location has an associated bit that when set, blocks writing into that location or execution of an instruction in that location; at the other extreme, one area of the memory may be protected. This feature is of great importance where the computer is being used concurrently in a number of applications. It is also a very useful debug aid.

* K = 1024

From the point of view of reliability parity checking and non-volatility (i.e. preservation of memory contents during a power failure) are desirable and available.

2.2 Arithmetic Unit

For very many on-line applications, it is not necessary to employ a computer with hardware for floating point arithmetic. However for those systems that require analysis of the data in the on-line computer by such programs as a three dimensional reconstruction program, this feature would be essential. Many of the small computers do not even provide hardware (except perhaps as an option) for fixed point multiply and divide. In those applications where the required computation is relatively simple and it can proceed slowly, a computer of this type could perhaps be chosen. Even when the hardware for multiplication and division is provided there is considerable variation in the complexity of the algorithms used in the hardware, giving a considerable range in the execution times.

Although indirect addressing is universally available, some machines do not contain index registers, or when they do these may be in fact memory locations rather than "flip-flop" registers (this is also true even for the accumulators in some machines). It is hard to believe that index registers would not benefit any on-line system both from the point of view of speed and programming techniques.

2.3 Interrupt System

This is a most important part of an on-line computer. This type of computer is expected to respond to external stimili, coming in an essentially random* manner from a number of sources. The relative importances of these stimili will be expected to influence the responses to them. Often the speed of response to a stimulus is critical. The response to an interrupt (i.e. stimulus) is that the computer discontinues its current activity, executes a program varying in length from one to several hundreds of instructions and then resumes its former activities.

The currently available computers offer a wide range of interrupt features, the more important are as follows :

Nature of Response

In general a computer responds in one of two ways to an interrupt. Either it is forced to execute a directly or indirectly addressed instruction of the Branch and Store type or it is forced to execute a single instruction (usually an add to memory type) and

---

* In this case random is meant to imply that the stimuli are not synchronized with the operations of the computer and one does not wish to assume any rigid time relationship between the various stimuli.

then immediately continues with the "interrupted" program. The second type of response is particularly useful in applications where the computer carries out simple counting operations, e.g. multichannel analysers, film transport systems, etc.

Interrupt Identity

Interrupts are identified by assigning to each interrupt or a group of interrupts a different memory location for its "forced Branch and Store" instruction. A given interrupt within a group has to be identified by the program examining some sort of status word or by "polling" a number of sense-lines.

Interrupt Priority

If all the interrupts in a system had the same priority then once a program had been started in response to one of these interrupts, then all other interrupts would have to wait until this program had been completed. In most systems however, this is not the case and a response is required to some interrupts in spite of the fact that computer is executing a program in response to an earlier interrupt. This feature is provided by assigning interrupts or groups of interrupts to a number of priority levels and allowing those of higher priority only to interrupt the processing of an interrupt response.

Speed of Interrupt Response

The speed with which a computer responds to an interrupt depends on many factors. Taking these in sequence from the time at which the interrupt is given to the computer, they are :

- Time for the computer to complete the execution of the current instruction. (In some computers this can be very long because, for example, a response is required from a peripheral device before the completion of an I/O instruction).

- Time for the priority system to decide the relative priority of this new interrupt and that of the interrupt currently being serviced.

- If necessary, time to complete the service programs for all higher priority interrupts and possibly for some interrupts on the same level.

- Time to execute the "forced Branch and Store" instruction sequence.

- Time to preserve status information, working registers contents, etc. of the "interrupted program" and to replace them by information required by the program to service the interrupt. Various features are available that are intended to reduce this time and the previous one; such as automatic exchange of Program Control

Register contents (i.e. register which contains address of next instruction, status information, interrupt mask, etc.); automatic exchange of the active working registers (i.e. groups of "flip-flop" registers used as accumulators and index registers) from one group to another.

Selective Enabling of Interrupts

Features are available that enable individual interrupts or groups of interrupts to be independently made sensitive or insensitive to interrupt signals, by setting suitable bit patterns in control words (i.e. interrupt masks) within the interrupt system.

Interrupt Triggering

This feature enables the computer itself to apply a signal to any one of its interrupt lines. This has a number of uses, such as in simulation programs that are designed to simulate a piece of equipment not yet attached to the computer.

2.4 I/O-System

The I/O-System of a computer can be divided into three main areas, Channels, Status Sensing and Device Control.

a) Channels

The basic characteristics of an I/O-Channel are : its width (i.e. whether it can transfer one bit at a time, or the six bits of a character in parallel, or a full computer word in parallel, etc.); its maximum transfer rate; whether it contains a buffer register and whether it can carry out autonomous transfers*.

Another most important feature of a channel is the way it can be connected to a number of devices and the manner in which it handles the transfers to and from these devices. A selector channel can be connected to a number of devices but at any given time only one of these devices will be selected and carrying out transfers. A multiplexor channel also has a number of devices (or device controllers) attached to it, but all of them can be selected and carrying out transfers at the same time. In its most complicated form the multiplexor channel provides for each device (or device controller) a data buffer register, a word count register and a memory address register, so that an autonomous transfer can be simultaneously carried out for each device.

* An autonomous channel, once the program has instructed it as to the number of words (or characters) and their location in memory, can independently carry out the specified number of I/O transfers, leaving the CPU free to continue execution of a program.

b) Status Sensing and Device Control

For both the standard peripheral equipment (e.g. card readers, magnetic tape units, etc.) as well as the special devices attached to the computer, it is important that the computer should be able to provide the program with the means of obtaining status information about the devices, channels and interrupt system and with the means of controlling the devices and interrupt system.

Status information is either obtained by reading in a special word via for example a channel or by a system of sense lines attached to the various devices and each of which can be individually tested by an instruction of the Sense and Skip type.

Device control is usually provided either by sending special words to the device via a channel or by providing a system of individual control lines, on which pulses can be generated by the execution of an Output Control Pulse type instruction, the address of which specifies the line to be pulsed.

2.5 Standard Peripheral Devices

Besides the normal peripheral devices necessary for program loading, compiling, testing, controlling, etc. (i.e. paper tape punch and reader, or card punch and reader, typewriter or teleprinter), it is often necessary to have a peripheral device for recoring output data. This is obviously a device capable of much higher transfer rates (50 to 150 Kc/sec) and to date this has been usually 1/2" magnetic tape decks writing at 200 or 556 bpi in "IBM compatible format".

A very interesting new development in the peripheral equipment for on-line computers are the smaller disk stores with capacities up to about half a million words (i.e. 8 million bits). Besides offering an alternative to magnetic tape as the output device of a system, because of their random access capabilities they provide a number of advantages from the programming point of view :

With a disk it is possible to extend the size of program used beyond the size of the core store, by overlaying unrequired parts of the program with new parts that are held on the disk.

The existence of a disk on a computer can facilitate the assembly, compilation and loading of programs. Because the assemblers and compilers can be kept on disk, only one pass of the object program is required. In addition all of the subroutine library and user programs can be held on the disk ready for loading.

Another peripheral device that is becoming more common in on-line systems is the CRT Display with Light-pen and Keyboard. With this device it is possible for the programs to rapidly display messages to the system operator; to display interesting data in graphical form and via the keyboard and light-pen allow the operator to input control information and orders in a convenient manner. For example the program could display on the screen of the CRT a list of alternative actions open to the system and ask the operator to choose one; which he would do by pointing to that action with the light-pen.

2.6 Software

It is essential that an on-line computer should be delivered with an Assembler, Loader and a library of routines for I/O, Arithmetic functions, floating point etc. In applications where the memory space is available and timing is not too critical, Fortran is very valuable. If the computer has a disk then the manufacturer should provide a full Batch Monitor system.

2.7 Miscellaneous

A number of other special features are available, such as :

Real Time Clocks - These are very useful in applications where recording the time of events is important or where it is necessary to initiate events on the basis of time.

Logic Modules - In almost all applications it is an enormous advantage to be able to purchase from the computer manufacturer the modules from which the computer is built. If the equipment attached to the computer is then constructed from these modules, interfacing is greatly simplified etc.

3. Applications of On-line Computers

3.1 Control Applications

Returning now to the list of applications given in section 1, we see that the first four items all involve the use of the computer primarily as a control element in the system. Many systems of this type are in construction or operation. Some examples are :

Control System for the Argonne ZGS (CDC 924 A) and the CERN PS (IBM 1800).

Beam Switchyard at SLAC (SDS 925).

Control System for the CERN 2m Hydrogen Bubble Chamber (PDP-9).

Spectrometer Data Processing System at SLAC (SDS 9300).

In these systems the basic tasks of the computer can be redescribed as follows :

Supervisory Function

The computer monitors a large number of 'go-no/go' signals, e.g. interlocks, vacuum valves open or closed, water flow relay detectors, etc., so as to ensure that all parts of the system are in a condition to allow the successful and safe operation of the system. Various methods for continuously scanning these inputs can be adopted, for instance a special scanning unit can be built from reed relays which examines each of these inputs in turn, presenting to the computer the status of the input and its number, so that a short program can be used to initiate this scanning every accelerator pulse, say, and to note any inputs that indicate a failure. The exact choice of the means of examining these inputs, is one of economics, reliability and speed.

Data Logging

Here the computer is required to measure (by means of an analogue to digital conversion) a number of important parameters in the system, often with respect to time. Examples of these measurements are :

Beam position measurements by pick-up electrodes in accelerators and storage rings.

Vacuum pressure in the interaction region of storage rings.

Currents in the coils of magnets both in accelerators and beam transport systems.

Magnet positions in beam switch-yards and spectrometers.

Variation of pressure with time in a bubble chamber.

Radiation levels.

Interaction Rates,

etc.

The exact way in which these various analogue signals are measured by analogue to digital converters and read into the computer depends on the number of signals and the speed and precision with which they must be converted.

Equipment Control

The computer can be given direct control over many parts of the equipment, for example by digital to analogue converters, or stepping motors driving helipots it can adjust magnet currents; by actuators, or electric motors it can open and close valves, rotate magnets etc.

Interruption

The interrupt facility of the computer is used in a number of ways, for example : if the start of the accelerator output pulse causes an interrupt, this can be used to initiate a sequence of measurements on the pressure in a bubble chamber or a supervisory scan of the system; certain critical signals can be connected to interrupts so that if a dangerous failure occurs, e.g. a sudden rise in pressure in a vacuum chamber, then the computer can take immediate steps, closing valves, turning off power supplies etc., to protect the system. Often a clock is used to produce interrupts at regular intervals so that system checks or parameter measurements can be initial at regular intervals.

Operator Communications

As usual this is a most important feature of these on-line systems. The simplest form of communication is by printing messages and numerical information on a typewriter and by the operator typing in responses. This is not very satisfactory because it is slow and it may require more highly trained operators than can be usually provided. Hence CRT displays are being introduced to give the operator graphical information and in some system special control and display panels are being built so that an operator familiar with the conventional means of controlling the system will have a minimum to learn when he starts operate the computer controlled system.

3.2 Spectrometers and Analysers

In these types of system, in addition to having to provide some of the features described above (3.1), the computer has now to read-in from the equipment data for Nuclear and High Energy Physics Experiments. A typical example of what this involves is given by the Spectrometer system at SLAC and the associated program system called Spectre. This system was intended to provide :

An event triggered system for data acquisition, filtering and recoring.

Data logging system (magnet currents etc.).

Display system to help detection of equipment failure and aid in the set-up and monitoring of experiments.

A set of equipment checking procedures, operated concurrently with the experimental data taking.

The system basically consists of an I/O multiplexor linking the computer to the conventional equipment of a spectrometer experiment, so that the computer can monitor and control the spectrometer and read in data from pulse height analyzers, coincidence circuits etc.

Event detection by the coincidence circuits can be used to generate one of twelve interrupts. The computer is an SDS 9300 with 32K core store, 2 million character disk, 2 magnetic tape units, CRT display with light-pen, teleprinter etc.

The program system Sceptre has been constructed so as to give the physicists running an experiment on the system a considerable degree of flexibility in rearranging his programs during the running of the experiment. The physicist has first of all to prepare a set of subroutines - written in assembly language or Fortran, which are held on the disk in binary relocatable form. The Sceptre system then allows him to form sequences of these subroutines, called 'Scripts'. When a Script's definition has been typed in, it is loaded into the core store in preparation for execution. Scripts are then executed in response to one of three events : a particular interrupt that was associated with the Script in its definition; asking for its execution by a typed in message; after a specified time interval.

The CRT display system is arranged so that the physicist can select one of several possible displays that he has already defined and introduce parameters by means of a set of thumbwheel switches.

In its simplest form a multi-channel analyzer system with a computer, merely consists of a conventional set of electronics containing amplifiers, discriminators, coincidence detectors and a pulse height analyzer. The coincidence circuits are connected to the computer's interrupt system and the pulse height analyser's output is either connected to an input channel or to a single execute interrupt system. When a coincidence is detected and the pulse height analyzer has prepared its output to indicate which 'channel' in the experiment must have a count added to it, this number is either read in by a short interrupt routine via an Input channel and the appropritate 'channel' count is incremented, or the number is used as an address in the core store so that a single execute interrupt can increment the 'channel' count directly.

More complicated systems are in operation involving several pulse height analyzers - for multiparameter experiments. In many systems the interface between the conventional electronics and the computer can be quite elaborate, so that for example all events can be input to the computer and recorded, but only a certain sub-set of the possible channels are analyzed immediately.

3.3 Film Measuring Systems

On-line computers are employed in a number of different types of bubble chamber and spark chamber film measuring systems :

Manual Measuring Machines and Scanning Tables - e.g. Franckensteins, IEP's, Image Plane Digitizers, Miladies etc.

In this kind of system the on-line computer plays its usual role of accepting data from the machines, carrying out simple tests on each measurement and then recording this data onto some output medium. This single activity alone is very valuable because it replaces the individual and unreliable card punches or paper tape punches that used to be used with these machines. In addition, when they were used off-line from a computer, their output data was usually only processed after a delay of many hours, so that machine or operators' errors could go undetected for some time, which often lead to the loss of as much as a days production.

To further reduce the number of rejects caused by these errors, the computers in this type of system are used to apply detailed tests to the data on all of the tracks measured; to check the sequence of operations carried and to provide the operator with facilities to correct errors.

In almost all systems the data relating to a track is tested by checking its smoothness (i.e. fitting a curve through the measured points and examining their scatter about this curve). In some systems a more stringent test is applied, that of trying to reconstruct the tracks in 3-dimensions from the measurements in the various stereo-views.

Again in this type of system communication between the operator and the computer is important, to inform him of his errors, what he should do next, etc. So far communication has been by means of typewriters, illuminated message panels, numerical displays, lights etc.

In a number of systems the computer is also used to control the measuring device. It is given control over the film transport system and is used to position the machine close to each of the fiducial marks and tracks that have to be measured, leaving the operator to complete the final positioning for the measurement and in addition ensuring that operator measures the correct number of fiducials and tracks.

Semi-Automatic Measuring Machines - e.g. Spiral Reader, SMP. etc.

This category of measuring machines contains various devices that are essentially developments of the manual machines, which attempt to raise the measuring rate and reduce the work of the operator. All of these machines involve much more data and higher date rates than in the previous category.

The data produced by these machines contains many measurements on tracks that are close to or crossing the tracks actually being measured; it is therefore necessary to filter out these spurious measurements from the data. In the case of the SMP this filtering is done in the on-line computer so that if the resulting data for the track is unsatisfactory, the operator can be instructed to remeasure the track.

In the Spiral Reader system the filtering is done off-line in a large computer; the on-line computer does very little with the data except check the fiducial measurements by examining their separations and possibly a simple reconstruction test on the vertex measurement. However the on-line computer is deeply involved in the control of the device. It is used in the stage motion servo-system and is used to control and check the automatic measurement of fiducial marks. The communications with the operator are relatively simple and are carried out mainly by means of a typewriter.

Automatic Measuring Systems

A great variety of measuring systems of this type have been constructed, the tasks of the on-line computer vary very considerably from the relatively simple situation of an HPD system (as for example at Brookhaven and College de France) to systems like PEPR where pattern recognition programs are executed in the on-line computer.

When an HPD is connected on-line to its own computer, besides controlling the HPD (stage movement, scan mode selection, film transport etc.); reading in the data, carrying out consistency checks and recording the data on magnetic tape; the computer usually has a number of other important tasks. When an HPD is operating in the so called "full-guidance" system, the first stage in processing the data is to eliminate all measurements that are not close to the expected positions of the tracks and fiducial marks, (the so called Gating operation). This is a convenient operation to do in the on-line computer as it greatly reduces the amount of data that has to be written on the output tapes. In many HPD systems the number of each frame appears as coded marks on the film which are actually measured by the HPD; the recognition and decoding of these marks must be done in the on-line computer. Its other tasks can be : the selection from the output data of measurements on fiducial marks; provision of various displays of the "gated" coordinates of the tracks etc.

Measuring systems such as PEPR, SPASS, POLLY etc., besides actually measuring the events recorded on the film also in various ways attempt to recognise where the tracks are in each picture and which of them belong to interesting events. These systems essentially use the film as a random access store which is read out by some sort of CRT scanner. Operating in their on-line computers are various forms of pattern recognition program that find tracks and fiducial marks and follow along tracks. The basic cycle of operations for these programs is that they instruct the scanner to examine a small region of the picture; then some combination of electronics in the scanner and software in the computer decides whether there is a line element (or spark) in this area and if so where it is and what is its direction. This line element may then be checked to see whether it can be matched to any previous found elements. Depending on the results of this the programs will then instruct the scanner to examine another area of the photograph, where for example the programs expect to find another part of a track.

The amount of guidance provided to these systems varies from none to rough coordinates of vertices (measured during prescanning of the film to find frames containing events) and to having an operator "on-line" to the system, providing help if the programs get into difficulties (e.g. POLLY). In this last type of system the complexity of the pattern recognition programs can be greatly reduced; however for the operator to be able to rapidly provide the necessary help, it is essential that the system has very good displays, showing what has to be measured and how the programs have interpretated it.

PROGRAM SUMX

J. Zoll[x)]

Introduction

These last days you have heard a lot about the processing of individual events. You have been presented with the various more or less automatic instruments for measuring coordinates on bubble chamber pictures. You have heard about THRESH which combines these measurements for each track to yield an estimate of its 3 -dimensional momentum. You have heard about the next link in the data-processing chain, GRIND which combinesthe momenta of all tracks of an event to test the various physical interpretation of each event. For the physically possible interpretation, GRIND does an adjustment of the momenta by the method of leastsquares and determines any missing quantities, such as the magnitude of the momentum of a V^0 or the momentum of an unseen neutral. The result of GRIND is the complete description of each event, that is the 4-momentum vector of each track, with error estimated, written into magnetic tape. This tape goes into SLICE, the next link in the chain. In the program the physically interesting quantities are computed, such as effective masses of particle-combinations angles in various Lorentz-frames and whatever you need. The result of SLICE in the Data Summury Tape for a bubble chamber experiment, containing event-for-event the quantities needed for the statistical analysis of the expe-

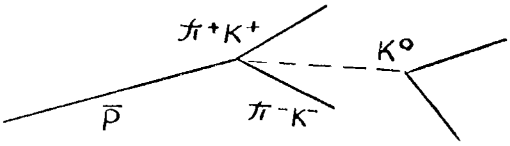
---

[x)] CERN

riment. By statistical analysis we mean the compilation of the simple-event data into histogram of I or 2 dimensions, the determination of mean and variance of some quantity over the whole experiment, the compilation of lists and ordered lists of event-data. This statical analysis is the task of the program SUMX, the last link of the data-processing chain at CERN. Although this is logically and historically the place where CERN SUMX has been written, it is not logically tied to the bubble-chain, but it is a self-contained general program. You could use it to evaluate the census of Russia, it has in fact been used to analyse the school-problems of the CERN children, and less exotic, it is in current use by the spark chamber and counter people at CERN. Further it is used in most high-energy laboratories all over the world, even in places where programs other than THRESH and GRIND are used for the bubble data-processing.

Example:

To let started we shall look at an example: the annihilation of antiprotons at rest into 2-prongs $+V^0$:



This topology contains the following reactions:

		hyp #
$\bar{p}+p \rightarrow K^0+K^++\pi^-$	$(+\pi^0)$	1 (101)
$K^0+\pi^++K^-$	$(+\pi^0)$	2 (102)
$K^0+\pi^++\pi^-+K^0$		3

With SLICE we have made ourselves a tape, the Data Summary Tape or DST, which contains all the events, may be I0000, I event per record, with the following information:

word I	Event number
2	Hypothaesis number (I, IOI, 2, I02, 3)
3	χ^2
4-8	P_x, P_y, P_z, P, E of K^0
9-I3	– " – of +ve particle
I4-I8	– " – of -ve particle
I9-23	– " – of unseen neutral, if any
24	M^2 of combination (K^0, +ve part)
25	– " – (K^0, -ve part)
26	– " – (+ve, -ve part)
27	– " – (unseen neutr. K^0)
28	– " – (– " – +ve part)
29	M^2 of combination (unseen neutr. -ve part) if any.

From this sea of data we may require SUMX to produce the following distributions:

for hypothesis I: the Dalitz-plot $(K^0\pi^0)$ vs $(K^+\pi^-)$

for hypothesis 2: – " – " – $(K^0\pi^+)$ vs $(K^-\pi^+)$

for hyp. IOI: the histograms of the masses $(K^0\pi^0)$ and $(K^+\pi^-)$

I02: – " – " – " – " – $(K^0\pi^0)$ and $(K^-\pi^+)$

for hyp. 3: the angular distribution $(K^0 K^0)$ and a histogram of the mass $(K^0\pi^+)$

Let us write down a formal plan of requirements which looks English but is easy to translate into computer-chinese:

SUMX pass: Alushta special

Tape: DST39 of Montanet on unit II

Selection : Event belongs to

subset IO if word 2 = I

subset II if word 2 = IOI

subset 20 if word 2 = 2

subset 2I if word 2 = IO2

subset 30 if word 2 = 3

subset 33 if event in subset 30 and if word $3 <$ IO

Compute

I) for subset 33:

set word $35 = \vec{P}_1 \cdot \vec{P}_2 / P_1 \cdot P_2$, $\vec{P}_1$ is in words 4-6

$\vec{P}_2$ is in words I9-2I

Histograms

I) for subset II: '$(K^0 \pi^0)$ eff. mass in hyp. IOI'

histigram word 27, 25 to 4 GeV^2, IOO channels,

2) for subset II: '$(K^+ \pi^-)$ eff. mass in hyp. IOI'

25 to 4 GeV^2, IOO channels,

3) for subset 2I: '$(K^0 \pi^0)$ eff. mass in hyp. IO2'

hist. word 27 . . .

4) for subset 2I: '$(K^- \pi^+)$ eff. mass in hyp. IO2'

hist. word 28 . . .

5) for subset 30: '$(K^0\pi^+)$ eff. mass in hyp. 3'

histogram word 24, . . .

and also word 28,

6) for subset 33: '(K^0K^0) angular distribution, $\chi^2 < 10$'

histogram word 35, -I to + I,40 channels

Dalitz-Plots

I) for subset IO:'$(K^0\pi^-)$ vs $(K^+\pi^-)$ in hyp. I' 60 channels in x ,

60 channels in y.

Particle masses: .5,5,.I4 GeV, total energy I860 MeV,word

25 is X , word 26 is Y.

2) for subset 20: '$(K^0\pi^+)$ vs $(K^-\pi^+)$ in hyp. 2'

60 channels . . .

Particle masses . . .

word 24 is X , word 26 is Y .

All done

You should note the following very clever idea which comes from BERKLEY SUMX: the selection of subsets of events is taken out of the rest of the histogramming procedure. On first-thought impulse one would have written for the first histogram:

Histigram

I) if word 2 = IOI : '$(K^0\pi^0)$ eff. mass in hyp. IOI' etc. On second thought you wish notice the enormous advantage of the separation. The criteria as to whether or not a given event belongs to a given subset are written down once. Later on, when the specification for the operations are given, a single number is needed to specify the wanted subset, no matter how complicated the selection criteria may be. This idea is in fact of utmost importance and is largely responsible for the versatility of SUMX.

There is one other thing I would like you to notice when looking at this set of specifications: to accumulate a histogram or a Dalitz-plot, both are standard operations. They can be programmed into SUMX once-for-all, and the user simply calls for these standard operations with his control-cards. The same is true, to a less extent, for the selection of sub-sets. To do this, a piece of program in SUMX reads, interpretes and executes control-cards, which specify relationships between numbers, such as $=$, $>$, $<$. To do the same for the stuff called for under the heading of "compute" is impracticable. If one were to try, it would amount to writing a computer. SUMX has an elegant solution to this problem. The user can specify any non-standard computations he needs in the form of subroutines, called CHARM1, . . . CHARM9 which are compiled by the FORTRAN compiler. These become part of the program. On the control-cards the user calls for these computations for the wanted sub-set of events.

The Structure of the Program.

Now, I think, we have a fairly clear idea of what we want the program to do; so we may put down its outline specifications. The first thing we need, are 2 vectors, call them BOUT ans TEST.

BOUT holds the current event, as it comes from the DST. In our example, we had the χ^2 in word 3, in the program. This will be used as BOUT (3). The vector TEST (n) contains YES-NO information, indicating whether or not the current event belongs to sub-set n These 2 vectors contain the basic information needed everywhere in SUMX. Of course, they must be in common storage. Next we need separate pieces of code, which execute the various operations.

Each such piece of program we call a processor. Some of the existing processors in SUMX are :

block 6	makes histograms
block 7	makes 2-dimensional histograms
block IO	finds means, variance, minima and maxima
block I4	compiles lists of event-data
block I5	compiles ordered lists
TAPE	reads the DST
SELECT	checks whether or not an event belongs to the defined sub-sets
CHARM	controls auxiliary computations defined by the user by means of his Charm-routines.

There are some more standard processors in SUMX, with less obvious functions, but this list will be enough for you to get an idea. A processor consists of 3 parts, corresponding to the 3 stages of SUMX: during the first stage SUMX reads the control-cards on which the user specifies the detailed operations he wants.

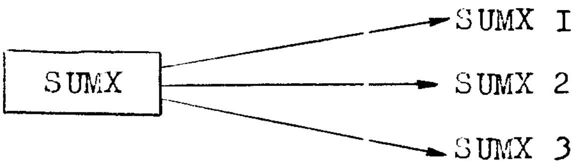
The cards look about like that:

```
* NEW PASS
Alushta special
* Tape
II          DST 39  Montanet
* SELECT
TEST        IO
            2     EQUAL I
TEST        II
            2     EQUAL IOI
...
TEST        30
            2     EQUAL 3
TEST        33
            3     SMALLER IO
AND         30    TRUE
* CHARM
ANGLE       33     I        4          I9
* BLOCK     6
'K⁰π⁰ eff. mass., hyp. IOI'
 IOO   25          4       II
 27
'K⁺π⁻ eff. mass, hyp. IOI'
...
```

```
* BLOCK 7
  '(K°π⁻) vs (K⁺π⁻), hyp 1'
test I0        I     60       60
 .5      .5     .I4      I860
25         26
  '(K°π⁺).....
. . .
* ALL DONE
```

The information from the control-cards is somehow encoded and stored away as parameters for the second stage when the DST is read and the various processors extract the information from the events. During the 3rd, the last, stage the compiled distributions are printed out for the user.

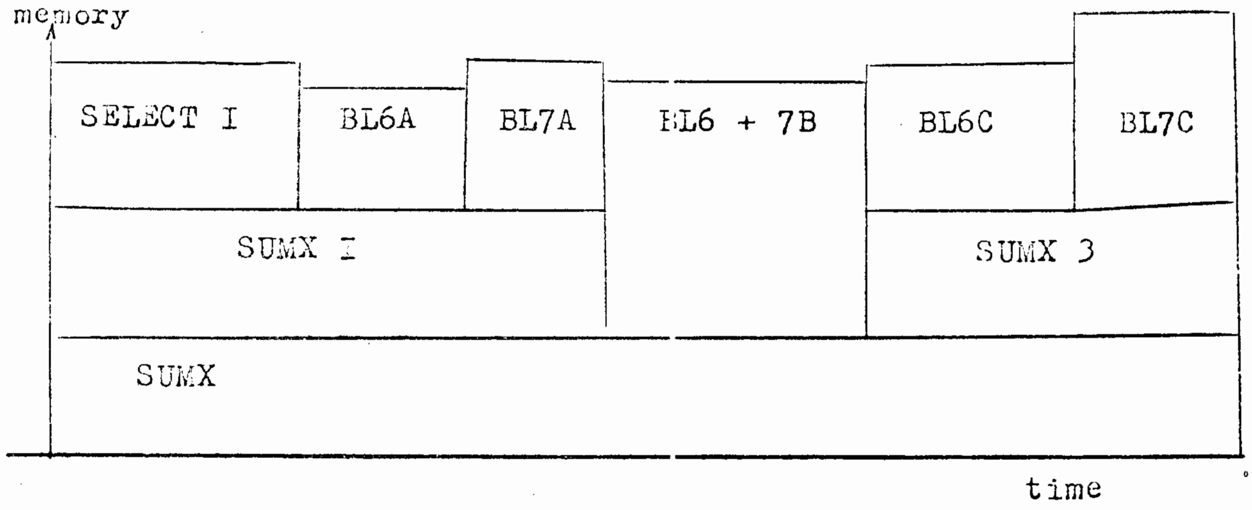
Let us look at this in somewhat for details:
the SUMX chief calls the 3 sub-chiefs in succession:



there by initiating stage I,2 and 3. During each stage the appropriate sub-chief has overall control, which it returns to the big boss when its work is done.

SUMX I starts reading the control-cards. It first takes note of the title of the pass. On finding the request-card * TAPE it jumps to part I of the processor TAPE, the FORTRAN subroutine TAPE I. This reads the control-cards to TAPE, in this case a single card, noting that the DST is to be read from unit II. When TAPE reads the 2nd card, it finds an asterisk in col.I. and returns control. SUMX I looks again at this card and finds * SELECT. So it calls SELECT, the part I of the processor SELECT which now reads all its control-cards. The process continues until the card * ALL DONE, when SUMX I returns to the master. During this stage I the order of requests is established: TAPE, SELECT, CHARM, BLOCK 6, BLOCK 7. This order is used in stage 2: for each event SUMX 2 calls first TAPE 2 to read the event into BOUT, it then calls SELEC 2 to establish whether or not the event belongs to the sub-sets defined, it next calls CHARMB which in turn calls the user's routine CHARMI to do his non-standard computations. Finally, it calls BL6B and BL7B to add the event into the various histograms. This process continues for all events. On reaching the end-of-file mark, SUMX switches to stage 3 by calling SUMX 3, which in turn calls the parts 3 of all processors for them to print their results.

Let us point out some of the properties of this set-up. Firstly, the various processors are independent of each other as a result each processor is a plug-in unit which you may put in or leave out. This does, of course, not mean that there is no communication between the processors, but this is strictly formalized: communication is only permitted via the 2 vectors BOUT and TEST, and further more it is completely under the control of the user. Secondly, this set-up is ideal for using the overlay technique. On the first level of overlay, only one logical third of the program is needed at any stage. Switching of the stage is very rare, only 3 program exchanges are needed for a normal simple run. For stage I and stage 3 we may use second-level overlay, exchanging the processors. This gives us the following core-storage map; which is schematic in that some pieces have been left out, but for the amount of memory needed by the various pieces it is about right.



So, most of the program resides on back-up storage, and only the needed pieces are brought to core-memory. This helps both for small computers and for multi-processing computers where core-space is usually a serious bottle-neck. Note that the processors are not exchanged in stage 2, which would mean several overlays for each event, which would be terrible.

The Dynamic Store

Next I would like to discuss a technique used in SUMX, which is natural to programs written in machine language, but is strange in FORTRAN. The problem is the following: the processors need some memory into which to put the accumulated information. This memory requirement is variable; for 2 histograms you may need about 200 words, for 100 histograms about 10 000 words. The same variability exists for all processors. The solution is one large common vector used by all processors, called the dynamic store. With our example this store would be used as follows:

1200 words	~40 words for Select	~10 words for CHARM	~100 words for Block 6	~1000 words for Block 7	TEST
BOUT	SELECT	CHARM	Block 6	Block 7	33 words

The storage for block 6 magnified might look like this:

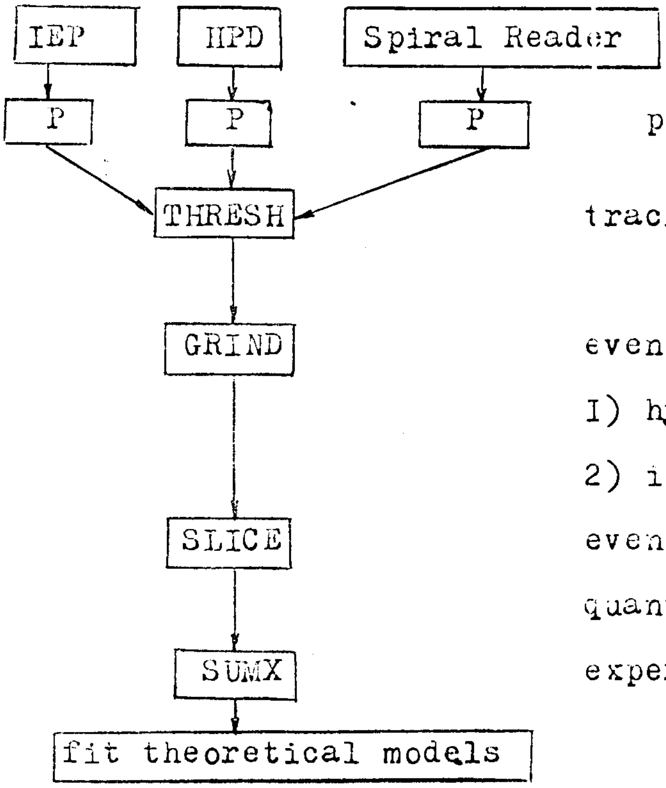
ii	ii of next hist.	
ii+I	multiplicity	
+2	wanted subset	
+3	number of channels	
+4	lower limit	
+5	upper limit	
+6	ii STORE	
+7÷I4	BCD title of histogram	
+I5	LOC I	multiple entries from BOUT (LOC)
+I6	LOC 2	
	. . .	
iist+0	under flow channels	
+I-I00	histogram channels	
+I0I	overflow channel	

ii NEXT	ii of next
+I	Multipl.
	...

Well, this is enough technical detail, so that you can imagine the rest. Of course, one needs some coding which looks after orderly use of the dynamic store.

This dynamic store in blank common solves a number of problems. First, of course, what it has been designed for, namely the efficient use of the precious memory. Secondly, all information which has to be passed within a given processor from stage I to stage 2 and to stage 3 goes via the dynamic store, it is not kept in memory local to the processor. This is essential for program overlay to work. Thirdly, all the information is always accessible. There is a block 4 in SUMX which I have not mentioned yet. It allows the manipulations of histograms, it can add several histograms together, and such like. This could not be done, if the histograms were burried somewhere in local memory of block 6. Historically, it was the use of this technique which enabled the break-through from the old Berkeley SUMX to CERN SUMX.

The chain of data-processing programs



point: bubble-image coordinates

track:reconstruction in space deliver

p for each track

event: kinematic fit

I) hypothesis testing

2) improved p

event: make DST containing derived quantities of physics interest

experiment: distributions

В Школе ОИЯИ по применению ЭВМ в задачах экспериментальной физики, проведенной с 5 по 19 мая 1968 года в г. Алуште, были прочитаны следующие лекции:

М.Г.МЕЩЕРЯКОВ (ОИЯИ) Вводные замечания

I. ЭВМ и программная организация их работы

Г.И.ЗАБИЯКИН (ОИЯИ)
ЭВМ в задачах экспериментальной физики

Е.П.КАЛИНИЧЕНКО (ОИЯИ)
Структура современных ЭВМ

А.А.КАРЛОВ (ОИЯИ)
СДС-1604А и организация ввода-вывода на этой машине

В.Н.ПОЛЯКОВ (ОИЯИ)
Вопросы сопряжения ЭВМ с внешним оборудованием

Д.ЛОРД (ЦЕРН, Швейцария)
Применение малых ЭВМ в экспериментах на линии

С.С.ЛАВРОВ (ВЦ АН СССР)
Состояние и перспективы развития математического обеспечения ЭВМ

Г.М.КАДЫКОВ (ОИЯИ)
Характеристики ЭВМ класса БЭСМ-4

П.ЗАНЕЛЛА (ЦЕРН, Швейцария)
Система машин ЦЕРНа

В.П.ШИРИКОВ (ОИЯИ)
Язык ФОРТРАН и программирование на нем

Н.Н.ГОВОРУН
Система математического обеспечения ЭВМ БЭСМ-6

В.А.РОСТОВЦЕВ (ОИЯИ)
Монитор для БЭСМ-6

Х.ЛИППС (ЦЕРН, Швейцария)
Операционная система СКУОП для СДС-6600

II. Вопросы обработки спектрометрической информации

И.ЗВОЛЬСКИ, Ю. ОСТАНЕВИЧ, В. ПРИХОДЬКО (ОИЯИ)
Применение ЭВМ в спектрометрических экспериментах ядерной физики

Й.ТОМИК, В.Р.ТРУБНИКОВ (ОИЯИ)
Осциллограф со световым карандашом как средство связи человека с ЭВМ

Л.С.НЕФЕДЬЕВА (ОИЯИ)
Программы обработки спектрометрических данных

III. Устройства и программы обработки информации с пузырьковых камер

Ю.А.КАРЖАВИН (ОИЯИ)
Полуавтоматические и автоматические устройства обработки фотографий

В.Д.ИНКИН (ОИЯИ)
Прибор для автоматического измерения фотографий с пузырьковых камер на базе механического сканирующего устройства типа "бегущий луч" (НРД)

В.Г.МУРХЕД (ЦЕРН, Швейцария)
Измерение фотографий с помощью НРД

Ж.К.ГУАШ, Ж.ТРЕМБЛЕ (ЦЕРН, Швейцария)
Спиральный измеритель

Г.Н.ТЕНТЮКОВА (ОИЯИ)
Математическая обработка фильмовой информации с пузырьковых камер ОИЯИ

Ж.ЗОЛЛ (ЦЕРН, Швейцария)
Программа САМЭКС

А.Ф.ЛУКЪЯНЦЕВ (ОИЯИ)
Программа ГРАЙДН

В.Г.ИВАНОВ (ОИЯИ)
О программе ПАЙТОН

Г.В.МАЙЕР, Д.ХАММЕР (ГДР)
Автоматический перевод программ с одной версии ФОРТРАНа на другую.

IV. Обработка информации с искровых камер

П.ЗАНЕЛЛА (ЦЕРН, Швейцария)
Обработка данных экспериментов, использующих искровые камеры

В.Н.ШКУНДЕНКОВ (ОИЯИ)
Сканирующий автомат на электронно-лучевой трубке

И.А.ГОЛУТВИН, Ю.В.ЗАНЕВСКИЙ (ОИЯИ)
Методика искровых камер для работы на линии с ЭВМ

И.М.ИВАНЧЕНКО (ОИЯИ)
Организация системы программного обеспечения экспериментов на линии с ЭВМ

С.С.КИРИЛОВ (ОИЯИ)
Измерительный центр Лаборатории высоких энергий

В.А.НИКИТИН (ОИЯИ)
ЭВМ в опытах по упругому рассеянию РР и Р-ядро в интервале энергий 1÷70 Гэв

А.Г.ГРАЧЁВ (ОИЯИ)
Система автоматической регистрации данных проволочных искровых камер на ферритах

V. Большие пузырьковые камеры

А.ЛЕВЕК (Сакле, Франция)
Проблемы измерения фотографий с больших пузырьковых камер

Б.П.КУЗНЕЦОВ (ИФВЭ)
О проекте пропан-фреоновой камеры СКАТ