Title

QUESTuAN ON-LINE EVENT-PROCESSING ROUTINE

Authors

Alston, Margaret H.
Braley, James E.
White, Peter.

Publication Date

1962-08-08

UCRL-10401

University of California

Ernest O. Lawrence Radiation Laboratory

Berkeley, California

UCRL-10401

UNIVERSITY OF CALIFORNIA

Lawrence Radiation Laboratory
Berkeley, California

Contract No. W-7405-eng-48

QUEST--AN ON-LINE EVENT-PROCESSING ROUTINE

Margaret H. Alston, James E. Braley, and Peter White

August 8, 1962

QUEST--AN ON-LINE EVENT-PROCESSING ROUTINE

Margaret H. Alston, James E. Braley, and Peter White

Lawrence Radiation Laboratory
University of California
Berkeley, California

August 8, 1962

ABSTRACT

An on-line computing system called QUEST can be used by a physicist to analyse unusual bubble-chamber events. The production version of a program called PACKAGE has been modified so that the physicist can control the progress of a specific event through the subroutines of PACKAGE. Feedback from the computer to the operator permits him to decide what hypothesis he should try next, depending upon the results already obtained. Because the physicist requires time to think what to do next, the QUEST system has been designed so that it can interrupt and then restore other programs.

QUEST--AN ON-LINE EVENT-PROCESSING ROUTINE[*]

Margaret H. Alston, James E. Braley, and Peter White

Lawrence Radiation Laboratory
University of California
Berkeley, California

August 8, 1962

At the Lawrence Radiation Laboratory a data-processing system is used in which the tracks of nuclear particle events occurring in a bubble chamber are photographed and analyzed. First the pictures are scanned; then if an interesting event is found the film is placed on a measuring projector (MP), and coordinate points along each charged track on three stereo views are measured. These measurements are converted to magnetic tape and, after some reorganization in a program called PANAL,[1] are processed by a program called PACKAGE. This program, written for the IBM 709 or 7090, is a combination of two programs already described: PANG[2] and KICK.[3]

PACKAGE consists of many very complicated subroutines which are the same for all types of events. The progress of each event is controlled by an event-type subroutine which calls the various subroutines in the main program. In general an event-type subroutine has to be coded for each topological event type in an experiment. The PACKAGE program is currently about 27,000 words long without the experiment-dependent event-type coding, which usually contains about 2000 words.

The PANG part of the program reconstructs the tracks in space from the coordinate measurements on the two best stereo views, and calculates

---

[*] Work done under the auspices of the U. S. Atomic Energy Commission.

the momentum at the center of the track and the azimuth and dip angle at each end of the track. Also it inserts and reconstructs the neutral tracks from vertex-point measurements. These calculations consist of two parts:

(a) a mass-independent part, in which a parabola is fitted to the measured points on the track images, and each track is reconstructed in space, without the loss of energy of the particle being taken into account;

(b) a mass-dependent part, in which the energy loss of the particle is taken into account, and the best space variables are obtained for one or more particle-mass hypotheses for each track.

The KICK part of the program applies conservation of energy and momentum at each vertex of an event by assuming a definite mass interpretation for each track taking part in the fit. If the fit is sufficiently overdetermined, the program calculates the χ^2 (probability) for this particular physical interpretation of the event. In this way many physical interpretations of an event can be tried, and the χ^2 of any particular hypothesis can be found.

This PACKAGE system works well for experiments in which there are many events of a certain topology, and an event-type subroutine trying all the most likely physical hypotheses can be programmed. However, for unusual events it is very time-consuming to program a special event type that tries all possible physical interpretations of the event.

This paper describes a modified version of the PACKAGE program, called QUEST, which has been written to allow a physicist on-line to the 709 computer to decide what hypothesis he wishes to try for a particular unusual event. Communication to and from the computer is by means of a typewriter. The feedback from the computer to the operator is most important because he he can decide what hypothesis he would like to try depending upon previous results. Because the physicist requires time to think what to do next and does not wish to be hurried by the knowledge that an expensive computer is idle, the program has been made so that it can temporarily interrupt some of our production programs. The QUEST program can also be used to process events which, although they have a usual topology, have failed for no apparent reason to fit in the production event type; QUEST can easily be used to try an unusual hypothesis for which the ordinary event type is not programmed.

THE PRESENT PROGRAM

General.

The present QUEST program is a reorganized version of PACKAGE to give a physicist the facility for controlling the flow of an event through the PACKAGE program on-line to the 709 computer. Information is inserted by the physicist by means of a typewriter connected to the direct data channel of the 709. The computer replies, also on the typewriter, with the results of calculations, or prompts the operator to insert further information. From the results of the kinematic fits already obtained, the operator can decide what fit he wishes to try next.

Different types of information are required to define an event type. Firstly, the topology of the event is required so that the measured tracks can be labeled and unmeasured tracks can be inserted. Secondly, the mass identities for each track have to be defined so that the mass-dependent part of the fitting procedure can be performed. Thirdly, in the kinematic analysis the tracks taking part in any vertex fit have to be specified.

Figure 1 shows a block diagram of the program. The left-hand part of the diagram represents PACKAGE, and the right-hand part represents routines that have been written to control the progress of the event through the processing. At several points the main program control is transferred to the operator, who can then transfer back to any one of many different points in the program. The insertion of information has been considerably simplified so that a physicist who is unfamiliar with the subroutines in the PACKAGE program can still operate QUEST. One important modification is that only one mass interpretation is assigned to each track at any time. In this way the track number gives directly the address of the storage locations of the track variables. According to this procedure, if a second mass hypothesis for any track is required, the operator must insert the new hypothesis and return to the mass-dependent fit in the PANG part of the program before starting any kinematic fit for the new mass hypothesis. This means that the physicist usually should try one complete physical interpretation for all the vertices in an event before continuing to another hypothesis.

At present, interesting events are measured on a "Franckenstein" measuring projector and run through the PANAL program. The PANAL output tape is then used as input to QUEST. Soon we will have a measuring projector connected to the direct data channel of the 709 and will be able to remeasure specific tracks or additional tracks if the fits obtained are unsatisfactory.

Control Routines for QUEST.

The two most used control routines in QUEST are IDENT and VERTEX. To insert the topology of an event the operator calls a routine named IDENT. After the computer has indicated the vertex and future vertex (terminal locations) of a track, the operator enters the charge and mass assignment. (For unmeasured tracks the operator has to enter all the information.) When identification has been given to all tracks, program control is returned to PACKAGE. The program then labels the measured tracks, inserts neutral tracks, and makes the mass-independent fit for each track specified.

The particle identities specified by the operator are then translated into mass code words, and the mass-dependent fit in PACKAGE is made.

If the operator requires the program to change some mass assignments to explore other physical hypotheses for the event, he can call the SPMASS (Special Mass) routine, which inserts the new mass codes and then returns control to PACKAGE for the mass-dependent fits.

When a vertex fit is required, the operator enters the word "vertex" followed by the type of vertex fit he requires, (decay, production, two-vertex) the number of measured tracks in the vertex, and the track numbers of the tracks that are involved in the vertex. The VERTEX routine then generates the complicated calling sequences required by the track-setup (ZSET) and vertex-fitting (ZVERT) routines. After the fit has been attempted, some information is printed out on the typewriter. This includes the χ^2 of the fit and the number of constraints, so that the operator can decide what to try next. In addition, the program outputs onto a tape, for subsequent processing, a complete binary record of the fitted variables of each track taking part in the fit. If the operator requires additional track information to be output onto the typewriter, he can request the relevant information by calling the REPORT routine.

Other routines that can be used by the operator are:

(a) VERTMM, calculates the mass missing at a vertex when the operator specifies which visible tracks are to be used;

(b) SWIM, which propagates the fitted variables at one end of a track to the other end and stores them so that they can be used in a subsequent fit;

(c) HUNT, which finds and reads in a particular event at the request of the operator;

(d) STORE, which stores fitted variables, if they are required for subsequent fits;

(e) DESTROY, which destroys stored fitted information when it is no longer required;

(f) VERTXX, which uses the "Extend" subroutine in KICK to slightly alter the angles and the errors of a neutral coming from a zero prong (the end of the zero prong being poorly defined because of gappiness in the track) and then goes into the normal VERTEX routine;

(g) INSERT, which allows the operator to change the variables of a track (for example, the mass value or the value of the error on a particular variable);

(h) BYPASS, which can be used by the operator if he finds that he has two measurements of the same event on an input tape, or that he has several events of the same topology (in this case it wastes time to enter the identifications of the tracks at the start of each event because they can be saved from the preceding event; the BYPASS routine reinserts the identification already used in the preceding event);

(i) TARGET, which permits the operator to change the identity of the target nucleus in a vertex fit.

In addition there are small self-explanatory routines END (end event), REWIND (rewinds input tape), and BSR (Back Space Record, is one event, on input tape).

There is also a CHANGE routine, which permits the operator or programmer to change any word in core by typing out the new instruction on the typewriter. This facility is used mainly for debugging the program. These control routines, including the typewriter decoding routines, occupy about 1000 words of storage.

Input-Output Control

The IO (Input-Output) routines in the QUEST program have been programmed in such a manner that the Direct Data Connection appears to the machine as a tape unit. On-line inputs are read into successive locations either from the typewriter or from the Franckenstein until the ENTER button on the console is pushed; this causes an End-of-Record (EOR) and End-of-File (EOF) code to be inserted. The EOF pulse to the data synchronizer terminates the channel operation. The EOF code (76) and EOR code (77) are inserted to enable the programmer to determine, on review, how the data have been entered with respect to records and files. This allows the programmer flexibility of input format.

Several other additions have been made to the BCD codes:

15 Upshift
16 Downshift } Case control for typewriter
35 Carriage return
36 Tab

At present we have found it possible and convenient to limit all entered comments to six or fewer characters.

Any unassigned BCD codes can be used for "formatting"--i.e., filling out blanks, etc. A word contains six BCD characters and at least one full word must be transmitted at a time. Therefore, if one wishes to print only one character on the typewriter, the word in core contains the character desired and five unassigned characters. The typewriter prints only this character with no additional spaces.

Interrupt.

Since the physicist must spend some time interpreting the output data presented to him, he has the option of allowing the machine to process another program during this interval. He presses a button labeled "continue," which initiates a direct-data interrupt and traps to an interrupt-control program. The control program then writes the contents of core memory onto tape, inputs another previously entered program into core, and starts to process the entered program. When he desires to return to QUEST, the physicist presses "access QUEST", and the procedure is reversed.

The interrupt-control program saves all registers, lights, channel information, and instruction locations--returning them to their "original" status for the subsequent program. Therefore, a program may be interrupted at any point and returned without the program's knowledge that it has been interrupted. The only restriction on the standby program is that it does not use the first 500 words (interrupt-control program) of core except for the trap locations.

At present--owing to tape-writing speeds--it requires 14 seconds to interchange programs. If a double-core 7090 or 7094 were available, this time would be reduced to milliseconds, and then all IO from the typewriter could be interrupted. The typewriter types at a rate of 11 characters a second, and much computation time is lost because of the series aspect of information transmission from operator to machine.

The interrupt-control program also has provisions to start a new standby program at any time--e.g., termination of old standby, error in standby program, etc. The provision is made also to either clear the core or not upon entering a new standby program. It had been feared that there might be some trouble with tape reading while programs were being transferred into and out of core. No errors have yet been observed. Since the interrupt program is available, the physicist does not feel that he is rushed by the computer sitting idle. It is our opinion that this, in itself, makes the interrupt feature highly desirable.

HARDWARE

The hardware consists of an IBM input-output writer, a set of telephone lines between the control station and the computer, and the necessary logic circuitry to provide for the flow and interlocking of data and control signals. In the near future a measuring projector (Franckenstein) will also be put on-line to the 709. The circuitry is shown in Fig. 2.

Input to 709.

Depression of the "Access QUEST" button on the control console establishes the logical connection of the system to the 709. A "Direct-Data Interrupt" pulse is initiated into the direct-data channel of the computer, causing a transfer of control to the interrupt-control program. The program, in turn, reserves the proper data channel and executes a read sequence initiating "ones" into the "Read Select" and "Channel Ready—Read" control lines. The coincidence of these two signals enables the input function of the system and turns on the "QUEST Proceed" light on the control console.

The data originating from the Franckenstein or typewriter are transmitted over the telephone lines six bits at a time and enter a thirty-six-bit register near the 709, through the data distributor. When the register is filled, a "Direct-Data Demand" pulse, initiated into the direct-data channel, causes the computer to accept the entire thirty-six-bit word.

The system is logically disconnected from the computer by a pulse into the "End of File" control line. This pulse is initiated by either an "Event Complete" code from the Franckenstein or by depression of the "Enter" button on the control console.

Output from 709

Execution of a write sequence by the control program initiates logical "ones" into the "Write Select" and "Channel Ready-Write" control lines. The coincidence of these two signals enables the output function of the system and turns on the "Output Active" light on the control console.

The computer provides output data in thirty-six-bit words by establishing dc levels on its output data lines. The data lines are scanned by the data collector, which feeds the data to the typewriter in six-bit codes. When all thirty-six bits have been accepted by the typewriter, a "Direct-Data Demand" pulse is initiated into the direct-data channel, causing the computer to provide the next output word.

Termination of output is under program control and occurs with the reset of the "Write Select" and "Channel Ready-Write" control lines.

Error Checking.

All data transmitted over the telephone lines are accompanied by a parity bit and undergo a parity check after their transmission. The occurrence of a parity error during either the input or the output operation turns on the "Parity Error" light and inhibits any "Direct-Data Demand" pulse into the computer. This occurrence during the output operation causes the typewriter also to stop.

The system can be re-enabled by depression of the "Erase" button (during the input operation) or the "Repeat" button (during the output operation). The former clears the thirty-six-bit register and re-enables the input function, while the latter re-enables the output function and causes the data on the data lines to be typed out again.

METHOD OF OPERATION

Figure 3 shows an example of the present typewriter format, together with a diagram of the event being processed.

Step 1.

The program reads the input tape identification, prints this out on the typewriter, and goes to "Control."

Step 2.

Since the selected event is not the first one on the input tape, the HUNT routine is called. If the event had been the first event on the tape, the READ routine would have been used. In either case the program reads in one event, returns to "Control" and prints out the serial number of the event in hand, together with the target identification--in this case PR for proton.

Step 3.

To identify the topology and mass of each track the operator calls the routine IDENT. First he specifies the total number of tracks in the event (11) and also the number of tracks that were not measured (4)--i.e., the neutrals. The typewriter then gives the LSZ number, vertex, and future vertex[4] of the track, and the operator types in the charge and also the mass identity he wishes to give to this track. This is repeated for all measured tracks. For unmeasured tracks the typewriter types "Z," indicating an unmeasured track. The remaining information is inserted by the operator so that the program can generate the track measurements. After all tracks have been entered, the computer returns control to PACKAGE, and the mass-independent and mass-dependent space fits are made. The program then returns to "Control" for the next instruction after printing some identifying information for the event.

It also writes PANG information for each track onto the binary output tape.

Step 4.

The operator starts to request vertex fits in the order he considers most appropriate. In this particular example he first tries the Λ decay [vertex 4, i.e., a decay with three tracks] and swims the fitted value of the lambda's variables (track 9) for use in vertex 3. He then does the Ξ^- decay at vertex 3 and swims the Ξ^- (track 4) to vertex 2. At vertex 2 he requires the interaction of a Ξ^0 with the hydrogen in the chamber. To obtain this he enters PROD (for production) in the vertex type and the number of tracks taking part in the vertex. In this particular case he calls the VERTXX routine, since he wishes to "Extend" the zero prong at vertex 1 and change the variables of track 8 because gappiness in the tracks prevents one from obtaining accurate measurements for the actual end point of track 1. After the fit, the fitted variables of track 8 at vertex 2 are stored and VERTMM is called to allow a calculation of the invariant missing mass of track 2 (π^+) and 4 (Ξ^-). This missing-mass calculation does not change the values of the variables of track 8 at vertex 2, therefore track 8 can now be swum to vertex 1. Finally vertex 1 is fitted. This is a production vertex with three tracks. In this case the K^0 track is neutral and not observed, and track bank 15 is defined and given the identity of a K^0. Also since track 1 (the beam track) is specified, the program gives the operator the option of beam-averaging its momentum.

Step 5.

Since all the fits in this event are good, the operator can END the event. However, he could also try the Λ as coming from vertex 1, 2, or 3; all these fits proved bad.

After the event is finished the operator can either READ the next event on the tape, or HUNT for a definite event.

USE OF PRESENT PROGRAM AND FUTURE PLANS

The program has been in operation since July 1st and has so far been used for ZOONS (unusual events) in the current experiment in which K^- are incident on the 72-inch bubble chamber filled with hydrogen. Several interactions and decays of Ξ^0 have been analyzed as well as some K^0 interactions. The program has proved to be very useful for this type of event. A typical event takes 10 to 20 minutes of 709 time to analyze. So far the project has cost about $3000 for hardware and 3 man-months of programming effort.

In the future we plan to put one of our large Franckensteins (MPIID) on-line to the 709 (see Fig. 2). This will allow the physicist to measure additional track if he cannot find the explanation of an event. Also, if he is having trouble obtaining a fit, he can look at the picture and remeasure one or more tracks in the event if he desires. This procedure should greatly increase the flexibility and usefulness of the project.

ACKNOWLEDGMENTS

The authors wish to thank Frank T. Solmitz and M. Lynn Stevenson for their advice and encouragement on this project. We also thank Robert Harvey for his assistance in disentangling some of the intricate details of the PACKAGE program.

REFERENCES

1. M. H. Alston, J. P. Berge, J. E. Braley, G. H. Campbell, R. J. Harvey, M. Hutchinson, and T. C. Schneider, IBM Program PANAL, Lawrence Radiation Laboratory, Alvarez Group Memo 358, November 1961 (unpublished).
2. W. E. Humphrey, Description of the PANG Program, and Supplement No. 1, Alvarez Group Memos 111, September 1959, and 115, October 1959 (unpublished).
3. Reference Manual for KICK IBM Program, edited by A. H. Rosenfeld, Lawrence Radiation Laboratory Report UCRL-9099, May 1961 (unpublished); J. P. Berge, F. T. Solmitz, and H. Taft, Rev. Sci. Instr. 32, 538 (1961).
4. LSZ number indicates whether the track was measured as: (a) leaving the chamber (L), in which case the momentum is determined from the curvature of the track; (b) stopping (S), in which case the momentum is determined from range; (c) of zero length (Z), which indicates unmeasured. "Vertex" is the vertex number from which the track comes. "Future vertex" is the vertex number to which the track goes.

FIGURE LEGENDS

Fig. 1. Overall flow diagram for QUEST program.

Fig. 2. Circuit diagram of the on-line typewriter and measuring projector (M.P.) for QUEST.

Fig. 3. An example of the QUEST program. The photograph and corresponding sketch show an unusual event from an experiment. The following reactions take place (track number in parenthesis):

Vertex ① $K^-(1) + p \rightarrow \Xi^0\ (8) + K^0\ (15)$
Vertex ② $\Xi^0\ (8) + p \rightarrow \pi^+\ (2) + p\ (3) + \Xi^-\ (4)$
Vertex ③ $\Xi^-\ (4) \rightarrow \pi^-\ (5) + \Lambda^0\ (9)$
Vertex ④ $\Lambda^0\ (9) \rightarrow \pi^-\ (6) + p\ (7)$

Tracks 1 through 7 are measured; the neutral tracks 8 through 11 are inserted by the program from vertex measurements. The on-line typewriter output from the event is shown in b. The underlined letters and numbers are those inserted by the operator; the rest of the printout is written by the computer.

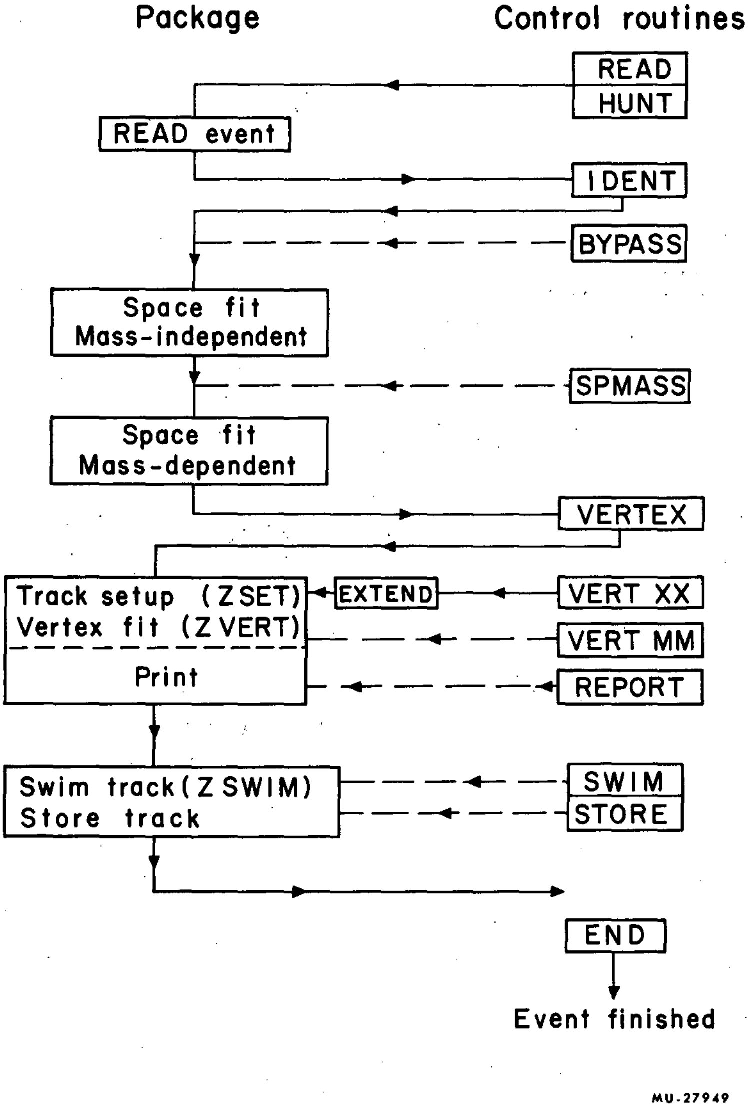

Package
Control routines
READ
HUNT
READ event
IDENT
BYPASS
Space fit
Mass-independent
SPMASS
Space fit
Mass-dependent
VERTEX
Track setup (ZSET)
Vertex fit (Z VERT)
Print
EXTEND
VERT XX
VERT MM
REPORT
Swim track (Z SWIM)
Store track
SWIM
STORE
END
Event finished
MU-27949

Fig. 1

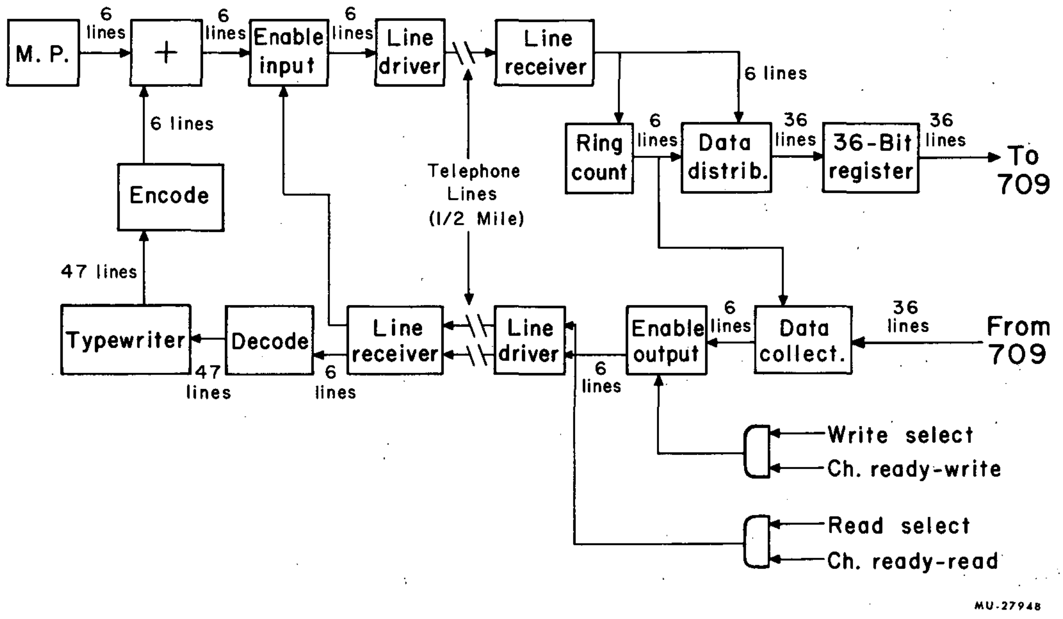
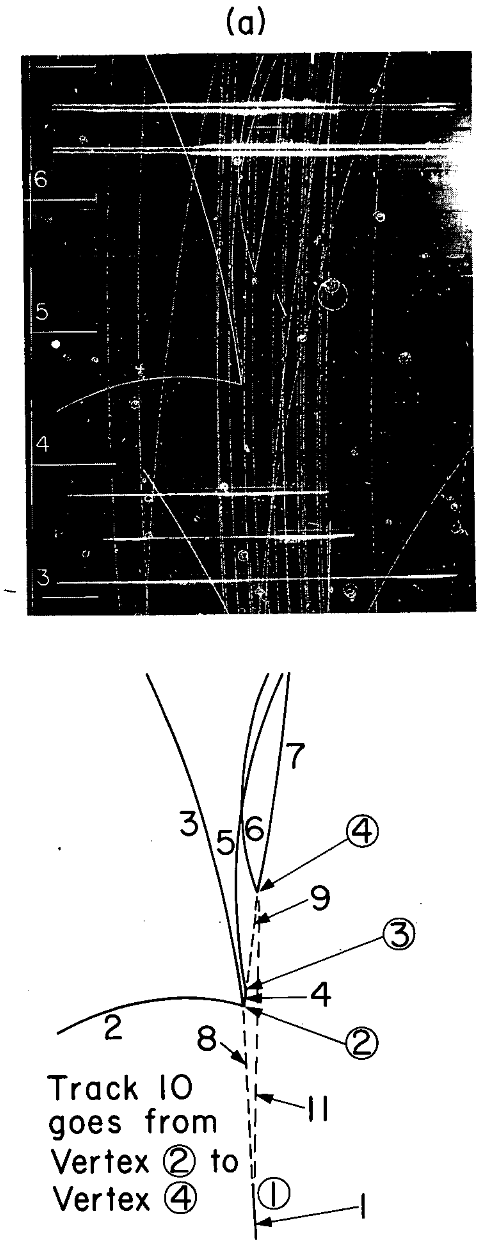
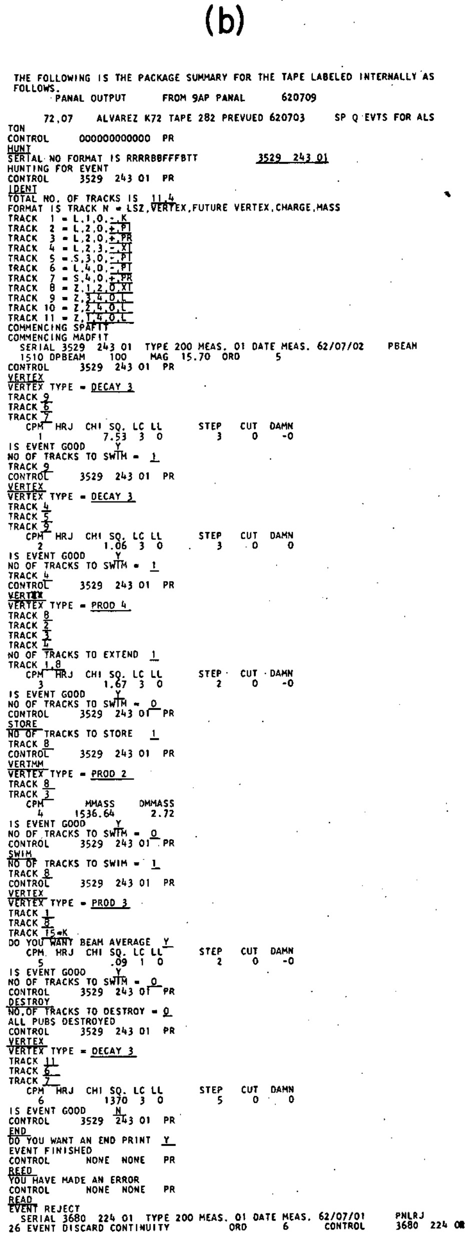

M. P.
6 lines
+
6 lines
Enable input
6 lines
Line driver
Line receiver
6 lines
Ring count
6 lines
Data distrib.
36 lines
36-Bit register
36 lines
To 709
6 lines
Encode
47 lines
Telephone Lines (1/2 Mile)
Typewriter
Decode
47 lines
Line receiver
6 lines
Line driver
6 lines
Enable output
6 lines
Data collect.
36 lines
From 709
Write select
Ch. ready-write
Read select
Ch. ready-read
MU-27948

Fig. 2



Track 10 goes from Vertex ② to Vertex ④



```
THE FOLLOWING IS THE PACKAGE SUMMARY FOR THE TAPE LABELED INTERNALLY AS
FOLLOWS.
     PANAL OUTPUT      FROM 9AP PANAL      620709

    72,07   ALVAREZ K72 TAPE 282 PREVUED 620703    SP Q EVTS FOR ALS
TON
CONTROL     000000000000  PR
HUNT
SERIAL NO FORMAT IS RRRRBBFFFBTT          3529  243 01
HUNTING FOR EVENT
CONTROL     3529  243 01  PR
IDENT
TOTAL NO. OF TRACKS IS  11,4
FORMAT IS TRACK N = LSZ,VERTEX,FUTURE VERTEX,CHARGE,MASS
TRACK  1 = L,1,0,-,K
TRACK  2 = L,2,0,+,PI
TRACK  3 = L,2,0,+,PR
TRACK  4 = L,2,3,-,XI
TRACK  5 = S,3,0,-,PI
TRACK  6 = L,4,0,-,PI
TRACK  7 = S,4,0,+,PR
TRACK  8 = Z,1,2,0,XI
TRACK  9 = Z,3,4,0,L
TRACK 10 = Z,2,4,0,L
TRACK 11 = Z,1,4,0,L
COMMENCING SPAFIT
COMMENCING MADFIT
  SERIAL 3529  243 01  TYPE 200 MEAS. 01 DATE MEAS. 62/07/02    PBEAM
  1510 DPBEAM    100    MAG  15.70  ORD      5
CONTROL     3529  243 01  PR
VERTEX
VERTEX TYPE = DECAY 3
TRACK 9
TRACK 6
TRACK 7
   CPM  HRJ  CHI SQ. LC LL       STEP   CUT  DAMN
    1           7.53  3  0          3     0   -0
IS EVENT GOOD      Y
NO OF TRACKS TO SWIM =  1
TRACK 9
CONTROL     3529  243 01  PR
VERTEX
VERTEX TYPE = DECAY 3
TRACK 4
TRACK 5
TRACK 9
   CPM  HRJ  CHI SQ. LC LL       STEP   CUT  DAMN
    2           1.06  3  0          3     0    0
IS EVENT GOOD      Y
NO OF TRACKS TO SWIM =  1
TRACK 4
CONTROL     3529  243 01  PR
VERTEX
VERTEX TYPE = PROD 4
TRACK 8
TRACK 2
TRACK 3
TRACK 4
NO OF TRACKS TO EXTEND  1
TRACK 1,8
   CPM  HRJ  CHI SQ. LC LL       STEP   CUT  DAMN
    3           1.67  3  0          2     0   -0
IS EVENT GOOD      Y
NO OF TRACKS TO SWIM =  0
CONTROL     3529  243 01  PR
STORE
NO OF TRACKS TO STORE   1
TRACK 8
CONTROL     3529  243 01  PR
VERTMM
VERTEX TYPE = PROD 2
TRACK 8
TRACK 3
   CPM        MMASS     DMMASS
    4      1536.64        2.72
IS EVENT GOOD      Y
NO OF TRACKS TO SWIM =  0
CONTROL     3529  243 01  PR
SWIM
NO OF TRACKS TO SWIM =  1
TRACK 8
CONTROL     3529  243 01  PR
VERTEX
VERTEX TYPE = PROD 3
TRACK 1
TRACK 8
TRACK 15=K
DO YOU WANT BEAM AVERAGE  Y
   CPM  HRJ  CHI SQ. LC LL       STEP   CUT  DAMN
    5            .09  1  0          2     0   -0
IS EVENT GOOD      Y
NO OF TRACKS TO SWIM =  0
CONTROL     3529  243 01  PR
DESTROY
NO.OF TRACKS TO DESTROY = 0
ALL PUBS DESTROYED
CONTROL     3529  243 01  PR
VERTEX
VERTEX TYPE = DECAY 3
TRACK 11
TRACK 6
TRACK 7
   CPM  HRJ  CHI SQ. LC LL       STEP   CUT  DAMN
    6           1370  3  0          5     0    0
IS EVENT GOOD      N
CONTROL     3529  243 01  PR
END
DO YOU WANT AN END PRINT  Y
EVENT FINISHED
CONTROL       NONE  NONE   PR
REED
YOU HAVE MADE AN ERROR
CONTROL       NONE  NONE   PR
READ
EVENT REJECT
  SERIAL 3680  224 01  TYPE 200 MEAS. 01 DATE MEAS. 62/07/01     PNLRJ
26 EVENT DISCARD CONTINUITY           ORD      6      CONTROL     3680  224 01
```

MUB-1291

Fig. 3