SLAC-PUB-255
January 1967

SYSTEMS FOR THE HUMMINGBIRD AT SLAC[*]

C. Dickens
Stanford Linear Accelerator Center
Stanford University
Stanford, California

(Presented at the International Conference on Flying Spot Devices at the Max Planck Institute, Munich, Germany, January 18 - 20, 1967)

[*] Work supported by the U. S. Atomic Energy Commission.

I. INTRODUCTION

Late in 1965 a joint research effort between IBM's Scientific Research Center, Palo Alto, California, and the Stanford Linear Accelerator Center was begun. Included in this effort was an IBM 360/50 provided by the Scientific Research Center. The Graphic Study Group (hereinafter denoted GSG) consisting of both SLAC and IBM personnel with W. F. Miller of SLAC as principal investigator initiated several projects.

One of the projects to reach early fruition was the building and subsequent incorporation into the IBM 360/50 system of the Hummingbird I scanner.

After looking at several scanning systems, we decided to follow as closely as possible the following general principles:

(1) Ease of Use

(2) Simplicity

Keep both the hardware interface and software as simple as possible to provide reliability and minimize maintenance and costs.

(3) Generality

To provide for future scanners and other peripheral devices with additional or completely different characteristics.

(4) Rapid Adjustment or Feedback

By users during developmental stages or production runs.

(5) Maximum Programability

Or to state it another way, let the computer perform adjustments to the scanner and in fact make all the decisions.

II. GENERAL CONSIDERATIONS

Adherence to the above principles led us to our first major decision that of making the scanner interface look like any standard peripheral I/O device. This as it turned out saved a tremendous amount of reprogramming in the systems area and allowed us to use all the software systems facilities provided by the manufacturer. The word format was thus dictated as an integral number of 8-bit bytes, and all communication,including interrupts, goes through a standard data channel.

The second consideration was the elimination of major decision making in the scanner or control unit. We wanted only one decision box namely the CPU of the Model 50 itself. This included the frequent adjustments normally required of a technician. Thus the scanner does not find data boxes nor does someone have to make minor adjustments to the digitizing threshold.

Thirdly, we decided to use the highest level programming language available to us, unfortunately Fortran IV and not PL/I, and at the same time to completely modularize the programs. As a result new functions may be added or old ones replaced or modified in the shortest possible time.

The last consideration was to provide immediate feedback to the user and relate his consequent request back to the scanning system. For this we have included a display scope as an integral part of the system. For example, we can look at raw scanner data and input new parameters to the scanner or program for a re-try at the same picture.

III. GENERAL DESCRIPTION

The scanning system consists of the Hummingbird I, an IBM 360/Model 50 with tapes and disks, and IBM printer and card reader, a 2250 Model 1 display scope with keyboard and light pen, an IBM 2701 Control Unit with a parallel data adapter, OS-360 Software System provided by IBM, and an open-ended set of Fortran subroutines to tie the scanning system together. The scanner is dealt with in detail elsewhere (1), (2), as are the IBM components (3).

The set of Fortran routines which give the scanning system its dynamic structure is completely open ended and starting from a basic scanner control nucleus (4) may be developed for a specific application into a complete analysis system. The basic program nucleus may be used as a stand alone program or merely as a subroutine for acquiring data and controlling the scanner.

IV. BASIC PROGRAM ORGANIZATION (5)

The basic program centers around a control driver through which the program flow is controlled by the user. Various functions such as writing an output tape are performed by a Fortran 'CALL' to a tape-writing-subroutine from the driver. Each subroutine is thus independent from the others, and may be easily replaced or modified. The option to bypass functions is also built into the driver via a series of switches which may be set by the user. As a consequence there is no chaining of subroutines and a return to the driver is required after each function performed in order to determine what is to be done next. All data transfer, for the same reason, among subroutines is done in 'COMMON'.

Generally the subroutines forming the basic package fall into three categories, scanner control and raw-data acquisition,data formating and output, and acquisition of user supplied control information and parameters.

V. BASIC PROGRAM CONTROL

We use three methods for inputing controls and parameters to the scanning system. The card reader is left open during a run and at user specified points new information is read in. Secondly, using the 2250 display unit with light pen and keyboard the same controls and parameters may be entered. Fundamentally the difference lies in the convenience of viewing all the settings and values at one time on the face of the scope, whereas using the card reader alone the user must remember the values he has input. The third method is setting parameters in the program itself, and since all parameters lie in 'COMMON', it is simply another subroutine added to the program by the user and 'called' from the driver. driver.

The parameters and controls entered fall into three areas; those parameters that control the read-in of subsequent parameters or terminate the read-in itself. Thus we may switch back and forth reading first one parameter or more from the card reader and then displaying those values on the scope and making further modifications. Those switches that determine the execution of various functions, and those that give program variables values.

Figure 1 shows the control page for the basic scanning program on the face of the display scope. As can easily be seen the mnemonics are for the initiated only and are actually the names in most cases of the variable in 'COMMON'. This is merely an expedient and could easily be modified for users unexperienced in the use of the program and expanded to the point where, if desired, you could teach someone to use the basic scanning system.

An example of each type of parameter is:

SCAN EXIT — terminates the input of parameters and proceeds with program execution

IWRTAP — if yes outputs a raw data tape, if no skips this function

NPIC — specifies the number of iterations the scanner operates with the other parameters unchanged.

VI. BASIC OUTPUT

At this time we have three modes of output of raw-data with two formats. The two formats are 16 bit words or 32 bit words with the leftmost 16 bits zeroed. The latter conforms to the minimum word size of early OS-360 Fortran programs.

For subsequent analysis of raw-data at a later time we may output each picture on tape, and for hand analysis of data we use printer records. However, for immediate response the raw data display on the display unit has proven most convenient. With this we can look at the results of certain parameter settings and change them accordingly. All this without moving from the display console.

VII. I/O INTERFACE

The I/O interface is described in full detail elsewhere (1) (6); however, all software was written as a GSG project. A new version is now being completed which follows all the standard IBM OS-360 conventions and will allow the scanner and analysis programs to execute in a multiprogramming environment.

VIII. ADDITIONAL PROJECTS

Following the same program organization and development principles, several allied projects are in progress. Note that the first two have both a strong similarity and a distinct difference. One is designed for completely automatic processing with on-line debugging while the other includes a human scanner using a display scope for normal processing. Both systems, however, use the same fiducial and spark recognition sub-routines and differ in the automatic phases at the control driver level.

IX. CBX SYSTEM

For the colliding electron beam experiment (7) a set of Fortran routines was written and designed to fit into a multi-programmed environment with the scanning system, which will automatically reduce the film data. There are three levels of tape output, the raw data, transformed spark coordinates, and physics output in the form of 3-D reconstruction. This system has been tested on simulated data, and we are presently waiting for film.

X. N-P SYSTEM

(SLAC, Michigan, Princeton collaboration)

The data from this experiment is recorded on two strips of film. Automatic processing of the proton film will yield output in the same form as a manual measuring machine would. This output is used to edit or gate the data on the neutron film. Automatic processing will continue through the fiducial and spark recognition phase.

At this point the spark data will be displayed on the 2250 for human interaction using the light pen. This is designed for finding vertices of neutron showers. Again, output will look like that from a manual measuring machine and the aggregate will go into a separate 7090 program for reconstruction and kinematic fitting. This system is now in the process of debugging on the first 200-frame test strip.

XI. ON-LINE SUMX

A control and display feature has been added to SUMX which uses the 2250(8). Thus the user may, by sitting at the console, control using both the scope and card reader the various passes through a SUMX tape and view the plots generated on the face of the scope. Presently this feature is available with Berkeley SUMX but will in the future be added to our copy of Cern SUMX.

XII. PERIPHERAL EFFORTS

A set of routines was organized to reduce a small set of bubble chamber photographs on the Hummingbird I as a test of the hardware (9). For looking at raw data and linked data fiducials or combinations of these another set of display routines, called BTEST, was put together. Additionally a display program for looking at the union and difference of two separate scans has proven useful for studying the digitizing threshold and stability of the scanner (10), and there has been some activity in the area of automatic bubble chamber scanning using Hummingbird I (11), (12), (13).

XIII. ACKNOWLEDGEMENTS

We would like to mention the people involved in the GSG effort, who have had a hand in the design and implementation of these various projects. In particular A. Leino (SLAC) has played the major role writing the control drivers for the spark chamber experiments, modified many of the subroutines, and helped in the general systems design; H. Lee (SLAC) for writing most of the later scope routines; M. Montalbano (IBM) who collaborated on the early scope routines (14) and with Leino developed the O.S. systems interface for the later routines.

W. Lucas (SLAC) has refined the spark linking and mapping routines. R. Levine's (SLAC) digitizing-association and fiducial-finding routines are still in use, and W. McGee (IBM) has been responsible for the On-Line SUMX.

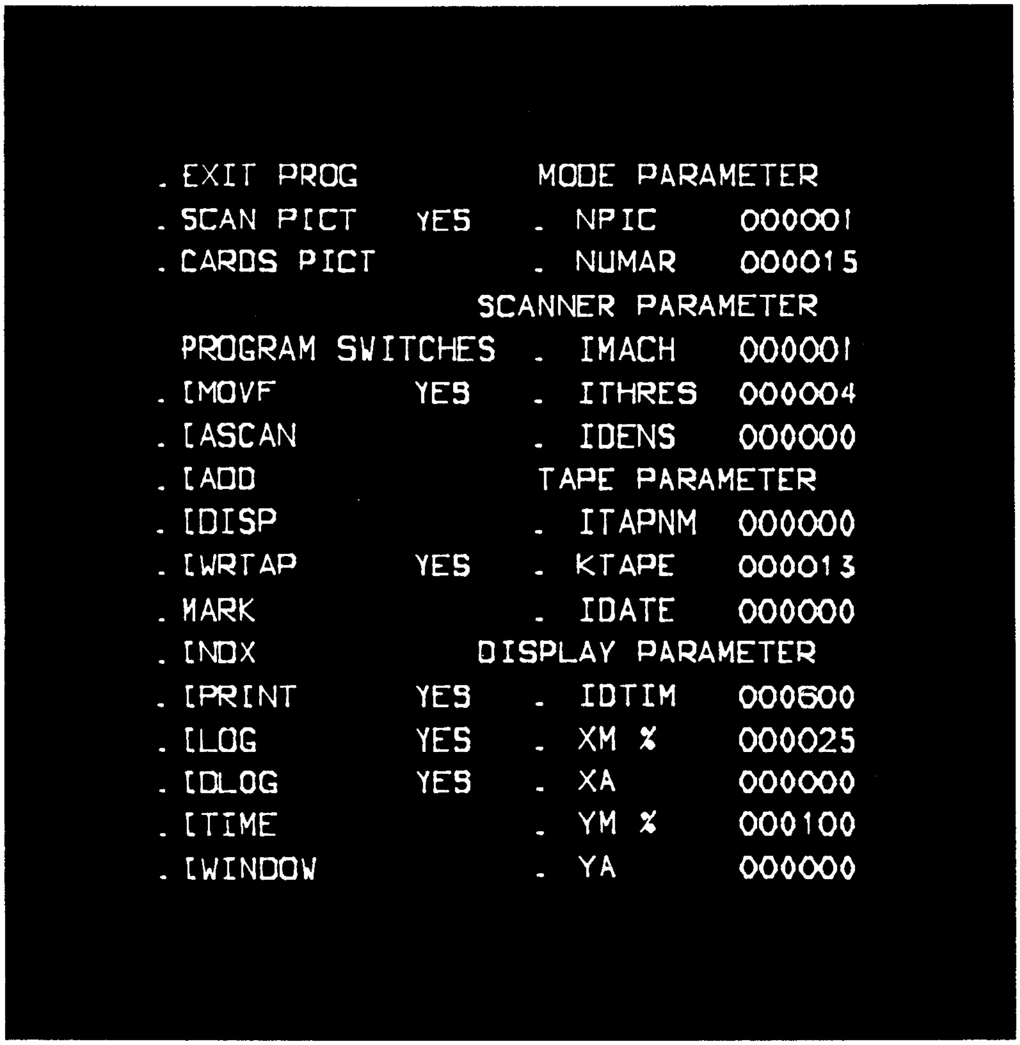
S. Torok (SLAC) has experimented with bubble chamber pictures, studied threshold levels and stability of the scanner and worried about data boxes.

Using OS-360 has been difficult at times and without the support of our system's people, L. Bastian (IBM), J. Cook (SLAC), J. Reed (IBM), we could not have proceeded very far. J. L. Brown (SLAC) has directed the activity in the CBX and N-P systems and worked with calibration (15) and transforming of data. W. F. Miller has given us the valuable advice and continued encouragement so vital in developing a system (16).

REFERENCES

1. J. Van der Lans, paper presented at this conference.
2. J. Van der Lans, GSG 100 and 101, Stanford Linear Accelerator Center.
3. IBM System/360 Bibliography (A22-6822-6);
   IBM Systems Reference Library.
4. C. Dickens, GSG 20, Stanford Linear Accelerator Center.
5. C. Dickens and A. Leino, GSG 200, Stanford Linear Accelerator Center.
6. C. Dickens, GSG 102, Stanford Linear Accelerator Center.
7. C. Dickens, GSG 7, Stanford Linear Accelerator Center.
8. W. McGee, GSG 19, Stanford Linear Accelerator Center.
9. J. L. Brown, GSG 16, Stanford Linear Accelerator Center.
10. S. Torok, GSG 17, Stanford Linear Accelerator Center.
11. W. McGee, GSG 8, Stanford Linear Accelerator Center.
12. S. Torok, GSG 9, Stanford Linear Accelerator Center.
13. S. Torok, GSG 18, Stanford Linear Accelerator Center.
14. C. Dickens and M. Montalbano, GSG 10, Stanford Linear Accelerator Center.
15. J. L. Brown, GSG 5, Stanford Linear Accelerator Center.
16. W. F. Miller, GSG 21, Stanford Linear Accelerator Center.

Note that all GSG papers are internal and tend to be rough hewn; consequently application may be made directly to the author for copies. Rather than list the incredibly long array of IBM 360 references, we mention only the bibliography.



```
. EXIT PROG               MODE PARAMETER
. SCAN PICT    YES     .  NPIC       000001
. CARDS PICT           .  NUMAR      000015
                       SCANNER PARAMETER
 PROGRAM SWITCHES      .  IMACH      000001
. [MOVF        YES     .  ITHRES     000004
. [ASCAN               .  IDENS      000000
. [ADD                    TAPE PARAMETER
. [DISP                .  ITAPNM     000000
. [WRTAP       YES     .  KTAPE      000013
. MARK                 .  IDATE      000000
. [NDX                 DISPLAY PARAMETER
. [PRINT       YES     .  IDTIM      000600
. [LOG         YES     .  XM %       000025
. [DLOG        YES     .  XA         000000
. [TIME                .  YM %       000100
. [WINDOW              .  YA         000000
```