

CGTM #29
October 25, 1967

THE SLAC CENTRAL COMPUTER

by

R.T. Braden and W.F. Miller

Computation Group
Stanford Linear Accelerator Center
Stanford, California

PREFACE

In June 1967, the Stanford Linear Accelerator Center entered into a contract with IBM for a large central computer -- a 360/91 system, with a 360/75 as an interim machine. This contract was the culmination of a selection and negotiation process that required a year and a half, starting with a Request for Proposal in December 1965. To satisfy the requirements of Bureau of the Budget Circular A-54 on acquiring ADP equipment, a comprehensive document on this acquisition was prepared in June 1966 for review and approval by the AEC. This document, known as the "A-54 Study", analyzed the need for a central computer at SLAC, the principal applications for this computer, the projected work load, the requirements and selection criteria, the cost/performance characteristics of the proposals which were submitted for SLAC's evaluation, and made a final recommendation in favor of the IBM 360/91 proposal. The initial 360/75 system was installed at SLAC in June 1967.

The present document incorporates much of the material in the A-54 study which might be of general interest. Since the study was written 15 months ago, we have made changes and additions to update it. Also, we have added some later material in Section 4.

We are much indebted to Myron Ruderman for his editing of the present document; with great finesse he has removed many rough spots from the prose of the original A-54 study without changing its content or style.

Table of Contents

	<u>Page Number</u>
1. Introduction	1
2. The Need for a Central Computer at SLAC	3
2.1 The Nature of the Computational Load-- A Discussion	3
2.2 Application Areas	6
2.2.1 Engineering Design Calculations	6
2.2.2 Theoretical Model Calculations	6
2.2.3 Chamber Physics Data Analysis	7
2.2.4 On-Line Analysis and Control	10
2.3 The Magnitude of the Load-- A Projection	12
2.3.1 Rates for Spark Chamber Film Analysis	16
2.3.2 Rates for Bubble Chamber Film Analysis	16
2.3.3 Computer Time Requirements for Chamber Data Analysis	17
3. Selection of the Computer System	18
3.1 Hardware Requirements and Specifications	18
3.2 Software Requirements and Specifications	21
3.3 A Summary of the Selection Process	23
3.4 Technical Considerations -- Comparison of CDC 6600 with IBM System/360 Models 75 and 91	24
3.4.1 Computation Speed and Throughput Capacity	24
3.4.2 Performance on Kernels	25
3.4.3 Instruction Set	27
3.4.4 Main Memory	28
3.4.5 Interrupts	28
3.4.6 Memory Protection and Relocation	29
4. The System Selected -- Summary and Retrospect	31
4.1 The Hardware	31
4.1.1 2075I and 2091K Central Processors	34
4.1.2 2860 Selector Channels	36
4.1.3 2702 Model 1 Terminal Controller	37
4.1.4 2321 Data Cell Drive	37
4.1.5 2820 Drum Controller	38
4.1.6 2701 - External Device Interface	38
4.1.7 2501 Card Reader	39
4.2 The Software	39
Appendix: Additional Computers in Use at SLAC	41
References	43

The SLAC Central Computer

1. Introduction

As physics research probes deeper into the fundamental nature of matter, experiments become increasingly complex, and the interpretation of data becomes more difficult. The volume of data to be analyzed and the complexity of the operations performed on the data are so great that experimenters must make extensive use of computers in their data analyses. Thus, a major concern of the contemporary high energy physicist is the development of suitable computing facilities at his laboratory.

The computing system under study in this report serves as the central computing facility for the Stanford Linear Accelerator Center (SLAC). We envisage a computer complex consisting of a central computing facility utilizing a powerful general-purpose computer as well as a number of smaller peripheral computers located throughout the laboratory and dedicated to particular data analyses and control tasks. [1, 2]*

The peripheral computers will generally be located at or near the locations of the experiments or apparatus which they control and service; thus, they will be physically remote from the central facility. Some of these remote computers will be functionally independent of the central facility. However, many of them will communicate with the central machine on a regular basis, either directly via cables (on-line), or indirectly via magnetic tape (off-line). Where system reliability is important, operation will be decentralized so that a remote machine can perform its primary function even if the central machine is not available.

In addition to these remote computers, we envisage "satellite" computers located close to and intimately tied to the central machine. Such a satellite computer would be programmed to act as a sophisticated input/output controller for the central computer. It could format the input data for the central computer or control a complex output device. For example, such a satellite computer will form the nucleus of a "graphics station", driving an incremental plotter, a display, and a microfilm recorder.

*Note: Numbers in square brackets refer to references which will be found at the end of this report. Refer to the Appendix for a list of the peripheral computers currently installed at SLAC.

When we speak of the "power" of a computer system, we imply that there is some analogy between its information processing capacity and the mechanical power which can be obtained from an engine or motor. The central computer system selected to meet SLAC's requirements will be the information-processing analog to the "prime mover" common during the period of the Industrial Revolution: centralized, fixed in place, and powerful; the central computer will provide the bulk of the computer "horsepower" for the Stanford Linear Accelerator Center. It is feasible (and for economic reasons, desirable) to acquire now a computer with enough "horsepower" to support the projected SLAC load at least through 1972.

We use the term "computation speed" to signify the maximum amount of computational and symbol-manipulation power which the central processor can deliver, assuming that suitable input/output devices are provided. As with mechanical horsepower, measurements of computation speed vary depending upon the nature of the load to which the "engine" is coupled. Thus, computation speed depends upon the access times, transfer rates, and capacities of input/output and on-line devices connected to the central processor.

The current generation of computers, usually referred to as the "third generation"*, has reflected a greater emphasis on input and output than was typical of previous machines. Recent developments in direct man-machine communication [4] will have a deep impact on future computing practice at SLAC.

The choice of input/output and on-line devices, the "power transmission" system, is sensitive to the details of the computer requirements of SLAC. Initially, SLAC should acquire a basic set of input/output (I/O) devices which will unquestionably be important throughout the lifetime of the central processor. It is clear, however, that developments and changes in the SLAC experimental program during the next five to ten years will make necessary the acquisition of other types of I/O and peripheral devices, some not even anticipated now. [2]

* Note: First generation usually means the electron tube machines such as UNIVAC 1 and IBM 704. The second generation usually means the solid state (more reliable and higher speed) machines such as the CDC 1604, IBM 7090 and their immediate successors. The third generation usually means the multi-programmable machines such as GE 645, IBM 360, CDC 6600, etc.

A modern computing system provides more than just rapid calculations to its users. [3] Therefore, characteristics other than computing speed are important. Previous memoranda [1] emphasized that one must also consider the quality of the results (accuracy and reliability), the timeliness in producing results, and the ease of utilization of the system.

In summary, it is clear that one must carefully weigh many characteristics of a computing system in planning a laboratory computing complex such as SLAC's facility. While "horsepower" rating is an important consideration in choosing a prime mover, what ultimately matters is the power delivered to the user. Also, it is important to consider not only future computing loads but future modes of operation. One must examine carefully the question of the impact of current research in computer science, making certain that the system is capable of continuing development.

2. The Need for a Central Computer at SLAC

2.1 The Nature of the Computational Load - A Discussion

Digital computation played an important role during the successful design and construction phases of the SLAC project. Extensive engineering design calculations were made on beam optics, bending and focusing magnets, and radiation shielding, using the facilities of the Stanford Computation Center. With the accelerator now in operation and experiments in progress, we expect computation directly related to the experimental program to increase rapidly and soon exceed past SLAC computer usage by an order of magnitude. Projections for the rate of use of the proposed central computer facility are given in Section 2.3 of this report.

On the basis of experience at SLAC and other high-energy physics laboratories, we foresee four major applications areas for the large computer facility :

- (1) Engineering design calculations,
- (2) Theoretical model calculations,
- (3) Chamber physics data analysis,
- (4) On-line data reduction and device control.

Areas 1 and 2 involve general scientific computation of small to moderate magnitude. Such programs are typically written in FORTRAN or AIGOL with an expenditure of from one man-day to one man-year of programming effort.

Area 3, the analysis of data from bubble chambers and spark chambers, represents one of the major components of the projected computing load at SLAC. These data analysis codes are large and complex; the Berkeley bubble chamber analysis codes alone represent twenty man-years of programming effort.

Area 4, on-line data reduction and device control, represents an area of rapidly growing importance to high-energy physics, which has been experiencing an "information explosion" over the past five years. Large-scale computerized data analysis, computer control of measuring tables, automatic measuring devices, spark chambers, streamer chambers, and wire chambers all represent steps toward escalation and automation of data taking. Since computers are closely tied in with the whole process of experimental physics at SLAC, it is no safer to predict the course of computation at SLAC five years from now than to predict the course of experimental physics. It seems likely, however, that the information explosion will continue and that, as a consequence, on-line data reduction and device control will grow to dominate the other three application areas.

Each of these four application areas is discussed more fully in Section 2.2 of this report.

Before proceeding to a discussion of these applications areas, it is useful to further examine the criteria used for judging the effectiveness of a computing system. The important characteristics of a system are

- (1) Speed of the computation,
- (2) Quality of the computation (especially precision and reliability),
- (3) Prompt delivery of results to the user,
- (4) Convenience for the user.

Items (3) and (4), promptness and convenience, are of special concern in the context of the high-energy physics laboratory. The experimental physicist looks at vast amounts of data with his computer programs. The programs are often large and complex, requiring a long time to develop and debug. After programs are debugged and completed, it is quite common for them to be revised to improve their efficiency, correct flaws, or to make them more sophisticated. Thus promptness and convenience in getting complex programs and large amounts of data processed and in and out of the system are crucial factors.

The properties of input/output equipment and its mode of operation are extremely important to a large facility like SLAC. We are acquiring a computer with a central processor between one and two orders of magnitude faster than a 7090, but it is not possible to get input/output equipment which is even one order of magnitude faster than equipment available on the 7090. Implications of this fact for physics bubble chamber data reduction are discussed further below, but in general new program organizations are required, and some subtle and difficult programming problems arise as a result.

Furthermore, the new modes of computer operation contemplated for the large SLAC computer - on-line devices, on-line remote consoles and displays - require considerable complex programming. Fortunately, modern computer technology provides at least a partial solution to the "programming bottleneck" with time-sharing, multiple-access, and on-line remote consoles. On-line consoles will play a key role in the SLAC central computation facility.

It is anticipated that consoles, in conjunction with on-line storage of program and data files, will ease two other potential bottlenecks: output printing, and expediting of magnetic tapes. In physics laboratories today, one 7090 usually requires one full-time 600 lpm (lines-per-minute) printer, while a 7094 II requires between two and three such printers for off-line output. Scaling this up by the machine speed ratio, an IBM Model 91 would require at least 20 1000 lpm printers. This is operationally absurd, and shows clearly that new modes of use must be established. We propose to write large-volume "in-case-I-need-it" output such as post-execution memory dumps onto a direct access storage device and allow the user to subsequently query this file from a remote console. He will then be able to request (via console) that selected portions of the file be printed, if necessary. Such dump files will be automatically purged by the system in 24 hours, unless the user specifically requests that they be saved.

The remote consoles could be completely justified by their effectiveness in breaking the programming and operational bottlenecks. However, once available, they open up many possibilities for new kinds of service involving direct man-machine interaction.[4] A simple example would be the use of a remote console as a sophisticated desk calculator. A far more difficult (but feasible) project would be the creation of a complete on-line information system for collection, organization, and dissemination of the whole spectrum of information flowing through SLAC - from budget figures to scientific preprints.

To summarize, we expect that there will be several important modes of computer usage at the SLAC central facility:

- (1) Batch processing;
- (2) On-line data acquisition and control computation, in conjunction with measuring machines, experimental devices and peripheral computers at SLAC;
- (3) Program development, debugging, and output querying via consoles; and
- (4) Man-machine interaction situations, including information retrieval and symbolic computations.

2.2 Application Areas

2.2.1 Engineering Design Calculations

The laboratory will continue to have a substantial demand for engineering design calculations connected with development of experimental devices, such as bubble chambers, spectrometers, bending magnets, waveguides, etc. One of the important computational problems in a particle physics laboratory is the calculation of magnetic fields for magnets of different designs and the calculation of the trajectories through these magnetic fields. Examples of the latter type are the much-used code, TRANSPORT [5] and the special-purpose code "Solenoid Optics". [6] The magnet design calculations are of a much more special character than the trajectory calculating codes, but a typical example of such a code is "CONFORMAL MAP". [7] Such codes permit magnet design with a great deal less experimentation than would otherwise be necessary.

The calculation of energy loss and ionization yields in electron-photon cascade showers is of importance in radiation shielding design as well as in studies of background levels in counter systems. These calculations, characteristically have been carried out by means of the Monte Carlo method and are relatively demanding of machine capacity, both CPU time and memory space. Examples of such work are the calculations of Zerby and Moran [8] and of H. Nagel. [9]

There will be a continuing need for general-purpose engineering calculations-- evaluation of complex functions, numerical quadrature, parameter searches by curve fitting [10] and factor analysis, numerical solution of algebraic equations, and minimization of functions of many variables. [11]

There is no need to go into greater detail here, as every large research laboratory has a substantial background of numerical calculations of importance in the engineering design of the equipment for the laboratory. We expect a mixture of long problems such as particle trajectory problems and cascade calculations, and small general calculations such as quadrature and minimization.

2.2.2 Theoretical Model Calculations

The exploration of new theoretical models and comparison of these models with experimental results are essential to high-energy physics. Comparison of theoretical models with experiment often involves a substantial amount of numerical computation. In such work, the programs typically solve the classical or the relativistic wave equation, calculate eigenvalues of matrices, compute solutions of simultaneous non-linear algebraic equations, or calculate various statistical distributions. [12, 12a, 13]

A computer application that will probably become more important for high-energy physics is symbolic or non-numerical computation, in which the computer is used as a symbol manipulator to yield algebraic (as opposed to numerical) results. Many tedious algebraic calculations in group theory and in parts of classical analysis are now being carried out on machines.

Indeed, since many theoretical calculations in particle physics involve complicated algebraic manipulations, computer application to such tasks should become more important at SLAC. The first step has been taken by Hearn, who has developed a LISP program [14] to perform the tensor algebra to reduce Feynman graphs. This program has already demonstrated the practical usefulness of symbolic computation for particle physics. An extended discussion of several non-numerical computer applications has been given by Miller. [3]

Symbolic computation will require list-processing software; on the hardware side, these programs typically require a very large central memory. Most symbolic computational problems could in principle be performed in modest-sized memories (e.g., 65K) through suitably clever programming. However, this is not a practical solution because programming complexity is one of the most formidable barriers to the general use of symbolic computation. The importance of a large directly-addressable memory in lowering this "complexity barrier" cannot be overemphasized.

2.2.3 Chamber Physics Data Analysis

The analysis of bubble chamber and spark chamber events combines the large-scale information storage and updating problems of commercial data processing with the complex numerical computations characteristic of scientific problems. The complete processing of measured events involves the successive steps of sorting, geometrical reconstruction, non-linear least-squares curve fitting, file updating, and hypothesis testing. [15, 16, 17]

This sequence of steps has typically been carried out by large and complex programs which operate in several passes. Each pass was originally a 32K core load on a 7094 and ran tape-to-tape. At each step, a number of "bad events" are found which require recycling through all preceding steps.

As the number of events to be processed increases, the operational complexity and difficulty of current processing techniques get out of hand. Each processing step typically requires a physicist's time to request a run, job, a dispatcher's time to schedule the job, and an operator's time to prepare the computer. Reports from another high-energy physics installation indicate that they require a minimum of ten days to pass one batch of events through the entire processing sequence. Many intermediate tapes are created and saved, and one large bubble chamber installation has a library containing over 20,000 tape reels. Expediting these tapes to and from the computer requires a number of clerical personnel whose salaries are a significant part of the processing cost.

Finally, it should be noted that the projected chamber event processing requirements at SLAC exceed the highest rates now achieved at other laboratories, making all the problems mentioned above even more acute. New techniques for chamber data analysis are therefore under consideration for the large SLAC computer.[18] The changes involve:

- (1) The use of a large central memory,
- (2) The use of direct access storage devices for data,
- (3) Reorganization of data records.

As noted earlier, the balance between processor and I/O speed for the large central computer at SLAC is considerably different than for earlier computers; processor speed has increased much faster than I/O speed. Fortunately, it will be possible to eliminate many of the I/O operations currently used to write and read intermediate results between overlays of programs into a 32K memory. This will be achieved by having a central memory large enough to contain the complete sequence of processing routines in one core-load. This is a particular instance of the general proposition that central memory can be traded against I/O operations. Improved core memory technology has brought the price of memories steadily down, and it is reasonable to specify a memory of 128K words for event-processing programs.* It will become clear that SLAC will need a large central memory for a number of purposes other than bubble chamber data reduction, but the need for large memory will nearly always turn out to be a consequence of the CPU-I/O balance mentioned earlier.

* Note: It has recently (September 1967) proved possible to fit all three Berkeley codes TVGP, SQUAW, and ARROW at once into 62,000 32-bit words, using the H Level FORTRAN compiler and combining COMMON blocks. Single (relatively simple) events have been processed through all three in an average time of two seconds apiece on the IBM 360/75.

Conventional event processing uses sequential storage (magnetic tape) on which a single record cannot be altered without recopying the entire tape. Whenever a few events need to be added to or modified in the event library, the entire library must be copied, causing a large number of unnecessary I/O operations. With events stored on direct-access devices (that is, disks or Data Cells), records can be updated individually.

Once the events are available on-line, it becomes feasible for the physicists to process small samples of events --or even single events -- on demand from remote consoles. Furthermore, several physicists can then be simultaneously processing such events. We believe that the SLAC physicists will use a combination of demand processing of a few events at a time with batch processing of moderate or large numbers of events.

No matter what the size of the batches, we anticipate that with remote consoles the persons involved in the traditional set-up cycle mentioned above --physicist, expeditor, dispatcher, operator --can be replaced by the physicist alone, working in or near his research area. Runs that he requests from a terminal can be set up, scheduled, and executed by the system. The resulting reduction in overhead should result in manpower savings and a reduction of errors common in handling a large volume of jobs and large tape files of data. Another advantage would be that physicists would be closer to and have more control over their data.

One million events, projected by SLAC bubble chamber physicists to be one year's output by 1969, requires on the order of 10^{11} bits of on-line storage. This falls into the range between the Livermore photo-digital store (10^{12} bits) and IBM Data Cell Drives (3×10^9 bits). There is currently no device in this intermediate range. Thus, SLAC will be forced to use a multi-level storage arrangement with "old" events "trickling" from Data Cells to tape for long-term storage. If the output of events increases further, acquisition of a photo-digital store will become desirable for SLAC.

It is important to choose very carefully the format of data records used for storing intermediate and final event-processing data. Minimizing the number of characters in each record will reduce the input/output time required by the program and increase the number of events which can be kept on-line.

Data-structuring techniques such as relative track measurements, variable length tables, and pointers can reduce record size to essentially the minimum possible. Furthermore, the high speed of the central processor will often make it more economical to recompute seldom-used quantities rather than to store every conceivable result in the record in case it might be needed.

These considerations lead immediately to a number of specific requirements for the SLAC central computer:

- (1) A large main memory, at least 128K words.
- (2) Direct-access I/O devices with very large capacity and reasonably short access time. It is desirable to have a number of independent access mechanisms, as well as removable disk packs, so that event libraries can be moved on and off the shelf.
- (3) A central processor with a command structure suitable for handling a variety of data formats, both binary and BCD, and for easily and efficiently formatting, packing, and unpacking information in data records.
- (4) Computer hardware and software suitable for remote consoles.
- (5) Computer hardware and software to make the data analysis codes reentrant, so that several physicists can use the same copy of the code simultaneously.

2.2.4 On-Line Analysis and Control

The last few years have seen considerable progress in computer hardware and software that permit on-line data analysis and control. Rapid analysis of data and return of the results to an experimenter during the setting up and conduct of the experiment have proved to be of great importance in both low-energy [19,20] and high-energy physics [21].

As described in the Introduction, the SLAC computer complex will include a number of remote "peripheral" computers in addition to the central facility. The peripheral machines are expected to exhibit the full spectrum of dependency on the central machine. One class of small computers will function quite independently of the central facility. Other peripheral machines will communicate with the central machine via direct cable and will be "on-line."

An example of the independent special purpose computer might be the SDS 9300 computer in the Counting House of End Station A. This machine acquires counting data from the 1.6 Gev, 8 Gev, and 20 Gev spectrometers to be stored on magnetic tape. The 9300 is programmed for each experiment to perform a running analysis of the data and to display a summary of the results [31]. After an experimental run is finished, there is still considerable computation to be performed on the data, off-line from the experiment. The 9300 is busy around-the clock, either monitoring an experimental run and acquiring data, or debugging physicists' programs for later on-line runs. Therefore the central computer will be required to perform the off-line analysis of data from the 9300. The data will be brought to the central computer on magnetic tapes.

An example of the on-line machine would be the IBM 1800 which is the computer component of a data acquisition system in a wire spark chamber experiment [32]. The 1800 does the bookkeeping and preliminary processing of the data. It also controls a display scope on which a summary of the data is displayed. The 1800 will be connected via cable to the central computer, so that the 1800 can periodically send blocks of data to the central computer for lengthier calculations. The central computer will in turn drive a remote display located near the 1800 in the experimental area.

System reliability is a necessity for the overall computer system at SIAC. However, ultra-high reliability requirements on a large central computer are very expensive to satisfy. To guarantee continuous availability of the central computer would require SIAC to acquire hardware equivalent to more than two complete computers. Since this kind of reliability in the central computer is economically unfeasible (and technically difficult), SIAC has taken the approach of reliability through decentralization of function. Each of the peripheral computers for which reliability is important will be capable of performing its primary function (though perhaps in a reduced or inconvenient manner) even if the central computer is not available. As a consequence, experimental work -- although it may benefit very greatly from the availability of the central computer -- will not depend upon it at all times.

For example, consider a peripheral data acquisition computer such as the 1800 discussed earlier. This computer will be capable of "logging" the data from the experiment -- recording it on magnetic tape, for example -- even when

the central computer is not available to receive the data immediately over communication lines. The logged data can then be processed later, when the central computer becomes available. On the other hand, when the central computer is available to receive the data over a communication line as it is acquired, substantial computation can be done immediately to give the experimenter more rapid feedback on the current state of his data and equipment. Another example is provided by the control computer which is now installed in the beam switchyard of the linear accelerator. The beam switchyard (BSY) can be operated manually, but it can be operated much more conveniently with the aid of the SDS 925 BSY control computer [33]. At some time, the BSY control computer will be connected via communication line directly to the central computer. However, use of the central computer by the BSY computer will be limited to functions which are auxiliary to the primary BSY operation. Therefore, if the central computer is unavailable, accelerator operations can continue using the BSY computer alone.

The necessity for on-line communication with a number of peripheral computers imposes certain requirements upon the central facility. The requirements include the following:

- (1) A fast and flexible interrupt scheme on the central processor.
- (2) A large main memory (or a moderate-sized main memory with extremely fast swapping capability), to hold the processing programs for peripheral computers with high rates of demand.
- (3) Adequate hardware and software to provide storage allocation and memory protection for these programs.
- (4) Direct-access devices for storage of programs and data used by the peripheral computers.

2.3 The Magnitude of the Load -- A Projection

This section is intended to supplement and update the data presented in the memoranda of two previous years, 1964 and 1965 [1, 22]. Estimates of SLAC's future needs are based on an extrapolation of past use and on the experience of other high-energy physics laboratories.

The principal use of computers at SLAC has heretofore been for engineering design and for analysis of experimental data from other laboratories. Now that the accelerator is operating and experiments are in progress, demands for computer time are rising sharply.

Figures 1 and 2 show past computer utilization in equivalent IBM 7090 hours. The summary includes the use of the Stanford Computation Center's IBM

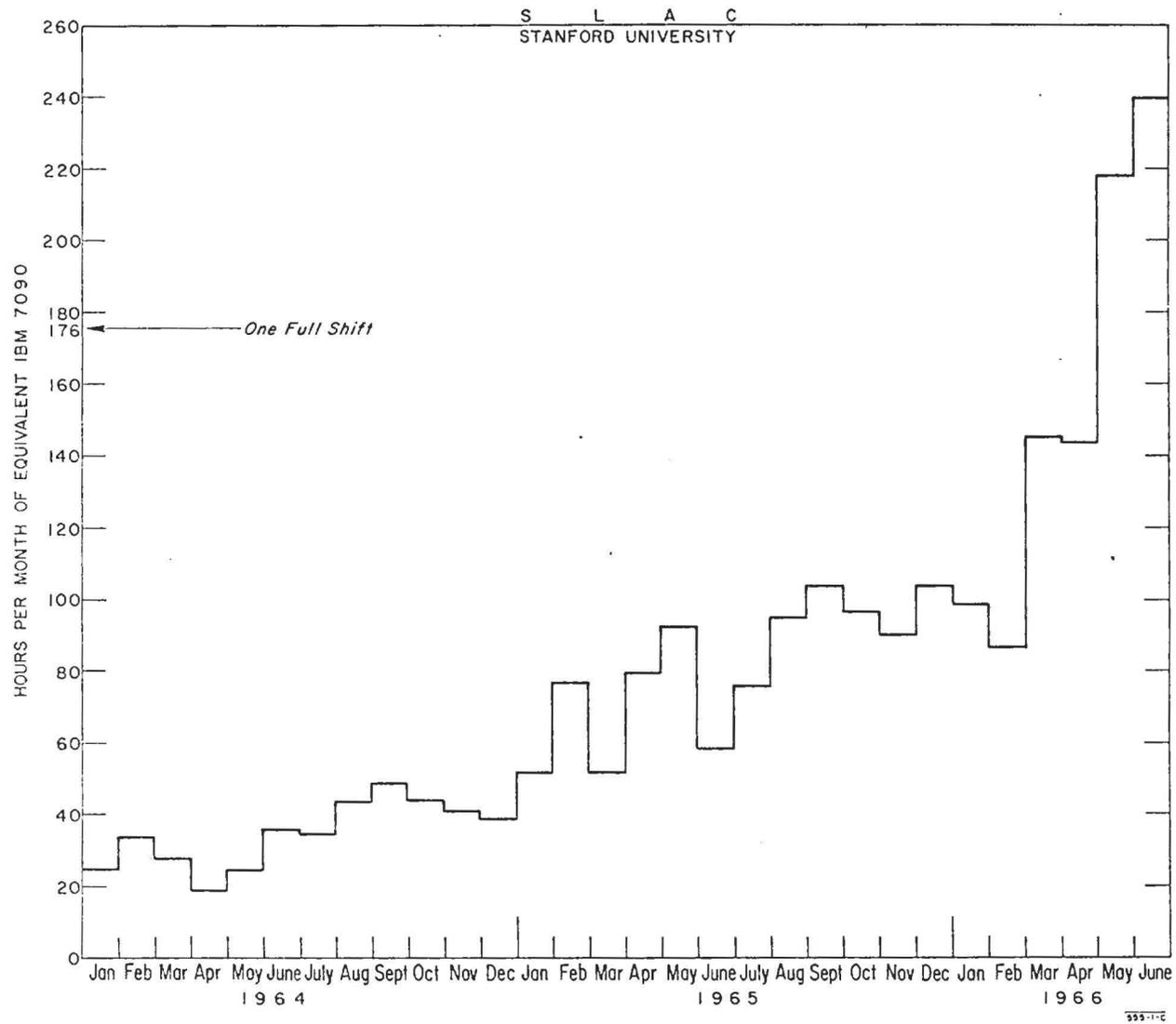


Fig. 1 CURRENT COMPUTER USE BY SLAC IN TERMS OF HOURS OF EQUIVALENT IBM 7090 TIME

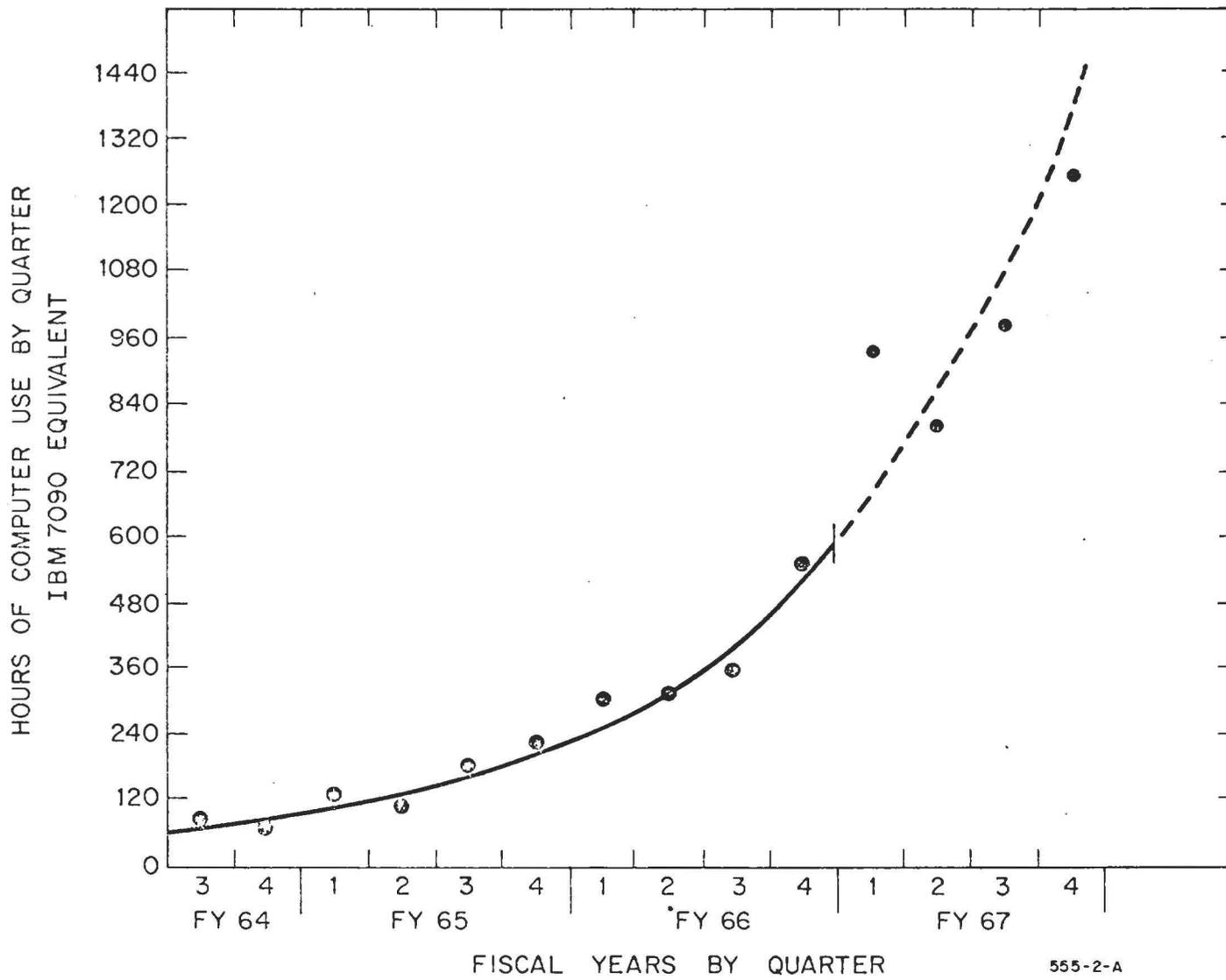


Fig.2-QUARTERLY SUMMARY OF COMPUTER USE AT SLAC

555-2-A

7090 and Burroughs B5500 and the IBM 360/50 in the SLAC-IBM Joint Study on Graphic Data Processing. Early projections indicated that SLAC would be using in excess of one shift of IBM 7090 equivalent time by June of 1966. The actual amount of equivalent IBM 7090 time used in June of 1966 was 239 hours or 1.35 shift months.* Also, earlier projections forecast a rapid increase in computer use when preparations for experiments with the SLAC machine commenced. Figure 1 supports this earlier forecast in that one sees a very rapid upturn in the use curve near the end of FY66.

In the 1965 Memorandum [1], available computers were categorized into speed classes on an arbitrary relative scale. More quantitative information has become available in the interim which necessitates some minor revisions of the classification. The IBM 360/92 is no longer available; nor is the CDC 6800. The fastest IBM machine still available is the IBM 360/91. The fastest announced CDC machine is the CDC 6600. However, CDC has accepted orders for a machine called the CDC 7600, a redesign of the 6800. While for purposes of this discussion a greater refinement of classification is not needed, it should be pointed out that the CDC 6600 and the Burroughs B8500 could be classified in a separate category intermediate between Class 1 and Class 2. It was argued in the earlier memorandum that SLAC's computing needs would require a machine in Class 1. The data presented in this report support that argument.

In 52 weeks there are 8736 hours. A smoothly operating machine that is not undergoing any hardware changes or additions can be available about 6000 hours per year under optimum conditions. The remaining time is used for preventive maintenance, unscheduled maintenance, equipment changes, and the "saturation effect". By the "saturation effect" we mean that the average use must always be less than or equal to the maximum possible use.

One might reasonably expect to devote half of the available machine time to the analysis of experimental data. The other half would be used for theoretical calculations, code preparation, design calculations, and software preparation and modifications.

*Note: We take a "shift" to mean 8 hours per day; then:
 a "shift week" is 40 hours,
 a "shift month" is 174 hours, and
 a "shift year" is 2088 hours (approximately).

2.3.1 Rates for Spark Chamber Film Analysis

The available information on the time required to process spark chamber data indicates that measuring and track-linking times on automatic devices will average about 8 seconds per stereo pair, depending on the complexity of the events [23, 24]. The geometrical reconstruction and kinematic analysis programs take about 10-12 seconds per stereo pair. The times quoted here are for the CDC 3600 and the IBM 709⁴ II. The combined time of measuring plus geometrical reconstructions and kinematics amounts to about 20 seconds for complex events. Simple spark chamber events can be expected to take about half that time.

This gives a total processing rate of 180 to 360 stereo pairs per hour. In 2000 hours, about one shift year, one can process about 360,000 to 720,000 stereo pairs on an IBM 709⁴ II. This would amount to about 120,000 to 240,000 stereo pairs per shift year of IBM 7090 time.

Later estimates for the IBM 360/75 indicate that 750,000 to 4 million spark chamber events can be processed per shift year. The higher figure contains no allowance for reruns, etc.

2.3.2 Rates for Bubble Chamber Film Analysis

One estimate of the time required for geometry and kinematics calculations is about 20 seconds per event on a DCD 3600 or an IBM 709⁴ II [25, 26, 27]. For the purpose of this estimate, we assume that scanning and measuring of bubble chamber film has been carried out on conventional tables. Any automatic scanning and measuring that requires CPU time will add to the estimate. In addition to the geometry and kinematics, one spends about an equal time in event analysis, for example in SUMX. Forty seconds per event means that one can realistically expect to process only about 180,000 events with one shift year of a 709⁴ II.

An estimate from LRL Berkeley indicates that the average time (counting reruns, etc.) for geometry, kinematics and event analysis is about one minute per completed event on an IBM 709⁴ II. This is a global average, obtained for an entire experiment and including the effect of reruns and "bad" events. One minute total per event would mean about 120,000 events per shift year. Fowler [28] has given the following times for the Berkeley codes [17] TVGP (geometry) and SQUAW (kinematics):

	<u>709⁴II</u>	<u>360/75</u>
TVGP	3 sec/event	1.3 sec/event
SQUAW	6 sec/event	4.2 sec/event

Assume that PANAL, SUMX, administration, etc., together take computer time equal to the time for TVGP and SQUAW; then the total time for bubble chamber data reduction and analysis is approximately:

	<u>7094II</u>	<u>360/75</u>
Analysis of one event:	20 sec/event	10 sec/event

These figures do not include allowance for reruns.

For the A54 study, we used an estimate of between 120,000 and 180,000 events per shift year on a 7094 II, or between 36,000 and 55,000 completed events per shift year on an IBM 7090. This corresponds to a total time per completed event on a 360/75 of 16 to 40 seconds.

The difficulty of determining realistic figures is illustrated by the fact that actual measurements on the 360/75 at SLAC have shown an average one-pass time of 2 seconds per event for TVGP, SQUAW, and ARROW on one experiment; this may be compared with 5.5 found by Fowler for TVGP and SQUAW alone. The difference is presumably due to a difference in event complexity.

2.3.3 Computer Time Requirements for Chamber Data Analysis

SLAC physicists are currently engaged in several spark chamber and bubble chamber experiments at other laboratories: Berkeley, Brookhaven, and Stanford High Energy Physics Laboratory. Small amounts of film data are being generated and analyzed now. We anticipate a rapidly increasing demand during the remainder of FY68; early increases have come from spark chamber film data, later increases will come from bubble chamber film data.

During FY68 two bubble chambers are being brought into operation at SLAC: the 72" Berkeley Chamber and the SLAC 40" Hydrogen Bubble Chamber. Planning ahead is very difficult, but (as a rough estimate) SLAC physicists expect from 300,000 to 500,000 events from the two bubble chambers in FY69. From timing estimates in the preceding sections, this much film will require the equivalent of 6 to 10 shifts of IBM 7090 time - about one shift on a 360/75.

By FY69 SLAC can expect to have at least three spark chambers taking from 1,000,000 to 1,500,000 stereo pictures per year. The analysis of these pictures including automatic scanning and measuring time would amount to about 5 to 7.5 shift years of equivalent IBM 7090 time in FY69 -- again about one 360/75 shift year.

Thus, the experimental program using bubble chambers and spark chambers is expected to account for 11 to 17.5 shift years of equivalent IBM 7090 time in FY69.

Although the computer will be heavily used for analysis of experimental data, the computer time required for code preparation, software changes, hooking up of equipment, etc., will also be substantial. We expect code preparation and software changes to take from 1/4 to 1/2 of the production time, that is from 5 to 10 shifts of equivalent IBM 7090 time. This added to the bubble chamber and spark chamber use gives a range of 16 to 27.5 shifts of equivalent IBM 7090 use. Assuming that some groups will use nearly their maximum time estimates and others will use more nearly their minimum, it is most reasonable to assume a use level within this range. Thus we assume a use of 20 shift years of equivalent IBM 7090 time for FY69.

The use curve in Figure 3 projects a doubling each year of computer use during the first few years. It shows a tripling of use in FY68 and FY69 when the experimental program is getting underway. The growth then slows down to doubling and finally to 50% increases in the later years of the planning period.

3. Selection of Computer System

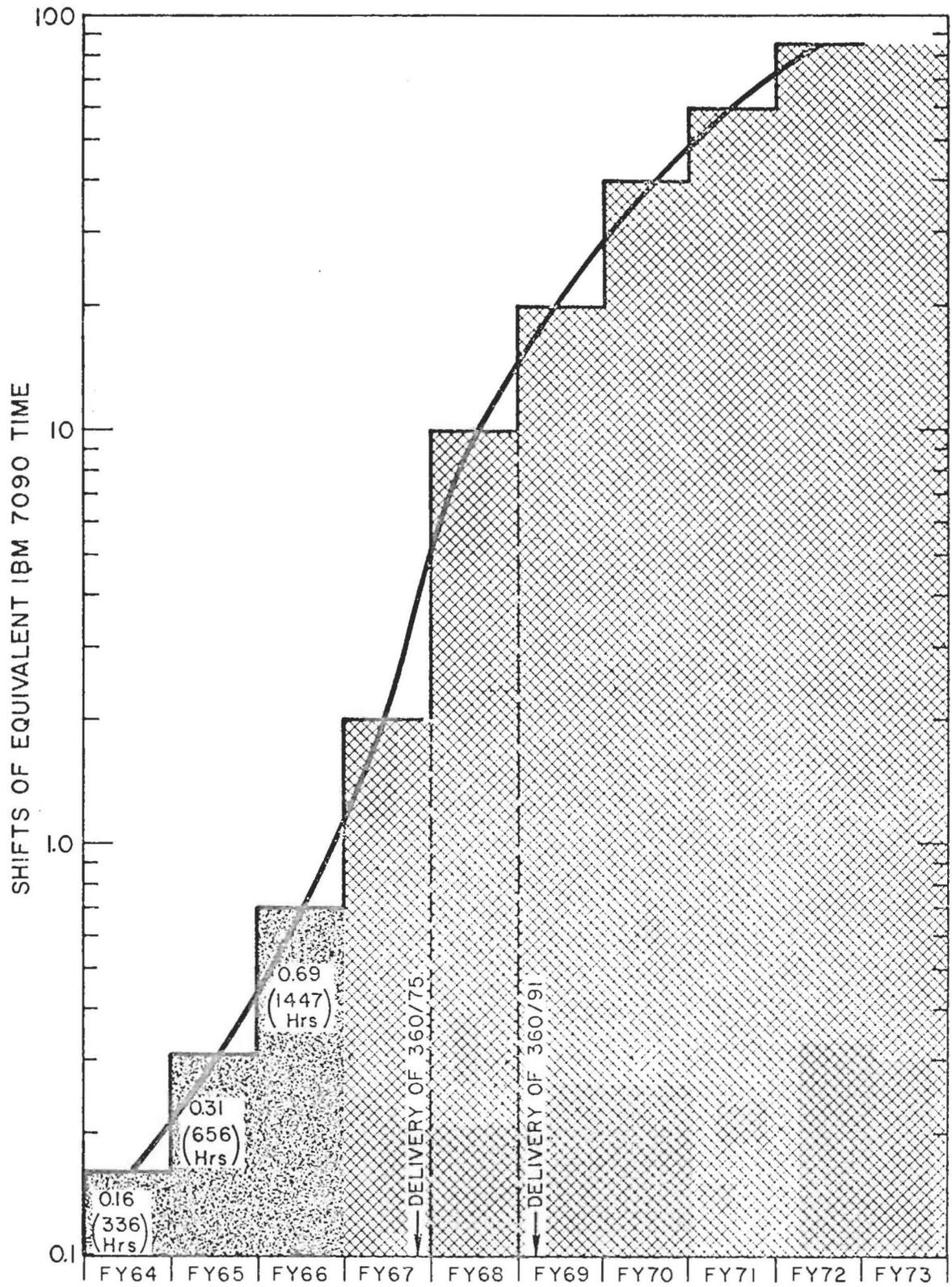
Section 2 has described the nature and magnitude of the work load projected for the SLAC central computer. The information developed in Section 2 led us to the requirements and specifications used in the Request for Proposal and the A54 Study for the central computer. These requirements are stated in the following sections (Section 3.1 for hardware, Section 3.2 for software).

3.1 Hardware Requirements and Specifications

Certain general requirements stand out from the discussion in Section 2:

- (1) The total computational load projected to 1972 requires a computer system with a computational speed at least 10 times that of an IBM 7094 II or a CDC 3600 (see Section 2.3.3).
- (2) The central processor must be a good high-speed numerical computing "engine", and also have good facilities for symbol manipulation, bit and character manipulation, and for handling a variety of data formats and word sizes.*
- (3) The system must have a very large directly-addressable central memory.
- (4) The system must be well equipped to handle a large number of peripheral and on-line devices using autonomous data channels.

*Note: This requirement may be met either through the hardware instruction set alone, or through a combination of hardware and software facilities. In any case the corresponding software must have provisions for bit and byte manipulations, data packing and unpacking.



555-3-B

Fig. 3 PROJECTED COMPUTER NEEDS FOR SLAC

- (5) The system must be equipped with one or more types of direct-access I/O devices, to satisfy a spectrum of requirements ranging from system overlays and program swapping to on-line storage of event libraries. It is unlikely that any single type of I/O device could satisfy the full range of requirements.
- (6) The hardware (and software) should be highly reliable and easy to maintain, consistent with the latest computer technology.
- (7) The CPU must have a flexible and fast interrupt mechanism.
- (8) The hardware, perhaps in conjunction with software, must provide a very high degree of protection for the operating system, other system programs, and users' programs and data, whether they reside in core memory or on direct-access storage.
- (9) CPU design should be suited for the compilation and execution of reentrant programs.

The following list of concrete specifications was included in the SLAC Request for Proposals from manufacturers for the large computer acquisition. These were specifications, not absolute requirements, and were intended as guidance for the computer manufacturers.

- (1) An average access time for the main memory of no more than $1/4$ microsecond.
- (2) A minimum word length of 48 bits with a minimum amount of directly addressable high-speed memory ($1/4$ μ sec) equal to 131,000 words.
- (3) A fast-transfer auxiliary storage for "program swapping", to hold current problems while in a time-sharing mode. A capacity of at least 10^6 words is required with a transfer rate of several words per microsecond.
- (4) An auxiliary drum or disk with a capacity of about 10^7 words.
- (5) An extremely large capacity storage facility that may be "read only" with a capacity of about 10^{12} bits.
- (6) Matching input/output equipment including card readers, card punches, printers, graphical recording equipment with photographic output and displays, remote consoles and real-time communication equipment, and magnetic tapes.
- (7) An accounting clock with Month, Day, Hour, Minute.
- (8) An interval timer with resolution of 100 microseconds.
- (9) Read-write memory protection for individual user programs as well as the operating system.
- (10) A dynamic relocation register or registers available only to the operating system.

3.2 Software Specifications and Requirements

Quality of results, prompt delivery of results, and convenience for the user are affected more by the system's software than by the hardware. These three standards should be used to evaluate software provided by the manufacturers making proposals. Unfortunately, software is not sold like hardware, so requirements and specifications for software do not have the same force as do those for hardware. On the other hand, while it would be impractical for SIAC to modify the manufacturer's hardware, it is feasible to modify and augment his software. To state some of the desired features more specifically,

- (1) Simplicity: The set of control cards necessary to run ordinary batch jobs on the computer should form a well-defined programming language. Furthermore, the "ritual" associated with running simple jobs should be correspondingly simple or nonexistent. There should be an absolute minimum of arbitrary conventions, rituals, and complicated rules for using the system.
- (2) Control by User: The programmer should be able to exercise a great deal of control over his program and over its system environment. For example, he should be able to get control asynchronously from any error interrupt, such as an arithmetic overflow, memory protection violation, or exceeding of time limit. Furthermore, the user should be able to employ the hardware protection features within his own protection sphere.
- (3) System Discretion: On the other hand, for the casual or experienced user who does not choose to exercise this control, the system should use its own judgment and make all the decisions necessary. For example, the user should be able to create and delete files routinely without considering space allocation or protection on the direct-access device.
- (4) Restart Ability: Whenever an error occurs at execution time for which no recovery is provided, the system should automatically check-point the job so that the programmer can restart it later, probably from the console.
- (5) Debugging: Debugging facilities should be provided, particularly for higher-level languages. This implies that the symbol table should be available at execution time.

The following software packages will be required for the SIAC computer:

(1) Operating System (Monitor)

This system should be capable of scheduling and supervising I/O operations, job sequencing and accounting, controlling multiprogrammed tasks, interrupting low priority jobs with higher priority jobs, spooling of input and output files, processing interrupts, passing software interrupts to user programs, terminating jobs which exceed time, memory, or output estimates, loading programs, allocating core memory, and communicating with the operator.

(2) Time-Sharing Executive Program

This program must be capable of running a batch processing operation with the facilities listed under (1) as a background, with a foreground operation permitting multiple access through remote consoles, including facilities for writing and editing programs, initiating jobs in foreground or background, debugging, post-mortem examination, file querying, and selecting output mode.

(3) Macro Assembler

(4) FORTRAN Compiler

The source language should include ASA FORTRAN IV as a subset. String and bit manipulations should be convenient. The compiler should provide clear diagnostics and forgive irrelevant errors. Both compiler and object program should be reentrant. Source-language debugging facilities should be provided.

(5) ALGOL Compiler

The ideal would be a fast ALGOL compiler with a source language compatible with Extended ALGOL for the Burroughs B5500. In particular, the source language should employ a reasonable subset of the ALGOL characters, including at least square brackets, and use reserved identifiers for the ALGOL control words. An alternative to an ALGOL compiler might be a PL/I compiler.

(6) List Processing System

(7) Utility Routines

These should perform a variety of system housekeeping tasks, such as preparing backup copies of files and moving old files to archive tapes.

(8) Linking Loader

A linking loader is used to combine separately-compiled subroutines into a single program that can be loaded and executed. It must be able to bind symbolic global references from one routine to another. It should provide an AIGOL-like block structure for global names, as well as flexibility in the use of overlays and noncontiguous storage space at run time.

3.3 A Summary of the Selection Process

The selection of a large computer to fill SLAC's requirements was made by a special committee under the chairmanship of W.F. Miller. The committee prepared a Request for Proposal containing the hardware and software specifications of Sections 3.1 and 3.2 above. The RFP was distributed in November 1965 to 24 computer companies. Three of the companies returned proposals for large computer systems: Burroughs (8500), Control Data (6800) and IBM (360/91). CDC subsequently withdrew the 6800 and offered a 6600 instead.

During the Spring of 1966 the Computer Selection Committee compared the merits of these three machines according to the criteria:

- (1) capability (that is, computation speed and throughput capacity -- see Section 3.4 below)
- (2) price
- (3) reliability
- (4) support by the supplier
- (5) compatibility with other installations
- (6) ease of SLAC's software development

The Committee concluded that the IBM proposal offered the best buy for SLAC on the basis of price and performance. In addition, the Committee felt that the IBM 360/91 was technically better suited to support the laboratory research needs. A technical discussion of the merits of two machines (IBM 360/91 and CDC 6600) is given in Section 3.4 below. This recommendation was incorporated in the A-54 study submitted to the AEC for approval in July 1966. Approval was given, and in December 1966 SLAC began formal contract negotiations with IBM. The draft contract was submitted to the AEC in April 1967. During the first week in June 1967, the contract was signed and the initial system was installed.

3.4 Technical Considerations -- Comparison of CDC 6600 with IBM System/360 Models 75 and 91

We will not attempt to give here a complete or detailed comparison between the CDC 6600 and the IBM System/360, Models 75 and 91, but rather summarize relevant aspects. Since the Model 75 and the Model 91 of IBM System/360 are nearly compatible, we will sometimes refer to them as one machine, "S/360".

3.4.1 Computation Speed and Throughput Capacity

Two fundamentally different quantities may be used to measure the performance of a computer system: computation speed (discussed in the Introduction) and throughput capacity. Throughput capacity is typically measured by running a "typical" series of complete jobs through compilation and execution, and tests the effective capacity of the total system, both hardware and software.

SIAC attempted to use both measurements in evaluating the machines presented for our consideration. The computation speed was evaluated by hand-timing a set of short "kernels" of code for all of the machines. Considerable care was taken to use congruent assumptions and rules for writing and timing these kernels. In addition, one physics program, PROGRAMMED DAVIDON [11], was used as a benchmark to evaluate throughput capacity.

For several reasons, we believe that computation speed is more relevant and significant for us than is throughput capacity. Firstly, throughput capacity is very sensitive to the current state of development of the software. Our benchmark revealed that both CDC and IBM were having difficulties with software performance. Both manufacturers made considerable progress in solving these difficulties over the ensuing year. In any case, it would indeed be surprising if, over the next five years, both manufacturers were not able to improve their software to the point that their throughput capacity approached its theoretical limit, the computation speed.

A second and more fundamental reason to prefer computing speed as a measure of performance for the SIAC system is that throughput capacity is by its nature a batch-processing concept. We expect that a major part of the load on the SIAC computing facility will be on-line and time-sharing use rather than batch processing. In the absence of any simple measure of the capability of a time-sharing machine, we used kernels to evaluate the computation speed; we also considered the qualitative suitability of the two machines, including memory sizes, swapping rates, I/O gear, storage protection, and provisions for reentrant code.

3.4.2 Performance on Kernels

Five different kernels were used to span a range of problem types, from pure computation to pure branching. It was our belief that this set of kernels gave a fair and reasonable evaluation of the computation speed of these computers. The kernels chosen for evaluation are the following.

1. Polynomial Evaluation

$$R \leftarrow (((((A_5 X + A_4)X + A_3)X + A_2)X + A_1)X + A_0$$

2. Floating-Point Arithmetic

$$R \leftarrow A*B + ((C+D)*E)/F$$

3. Matrix Multiply

$$C_{ij} = \sum_{k=1}^5 A_{ik} B_{kj} \quad \text{for } i, j = 1, 2, \dots, 5$$

4. "Clump Finding"

for I \leftarrow 1 step 1 until 1000 do

begin comment Each word of TABLE array contains two fields.
addr and decr are functions to extract these
fields;

if TABLE [addr(LIST[I])] < 0

then A \leftarrow A + decr(LIST[I])

else B \leftarrow B + decr(LIST[I])

end ;

5. IF Statements

begin if A > B then X \leftarrow 2.4;

if (I=J) \forall (K=L) then INDEX \leftarrow 1;

end;

It is important to notice that the System 360 programs for these kernels were very straightforward, being the code which one would normally write for a Model 50 or 75; there is no special optimization included for the 91. In fact, an advantage of the Model 91 is that it does a great deal of optimization in the hardware.

The 6600, on the other hand, has much simpler hardware which is more dependent upon special optimization for efficient use [29]. Therefore each of the 6600 kernels was written twice, first producing straightforward code under a set of rules such as might be used by a FORTRAN compiler which did not optimize for the concurrent structure of the 6600, and a second time with careful hand optimization. These two were averaged for each kernel, with a "compiled" case weighted twice as heavily as the carefully hand-optimized case. This is reasonable on the basis of what is possible within the constraints within which compiler writers operate.

On the 75, we used a weighted average of the times for full-precision (64-bit) and half-precision (32-bit) arithmetic, with full-precision times being given twice the weight of half-precision times. The assumption was that in one calculation out of three, advantage could be taken of faster execution and more compact storage of the half-precision numbers.

Under these assumptions, the following table shows the speed of the machine, relative to the same kernel running on a 7090.

	Computation Speed Relative to 7090			
	<u>6600</u>	<u>75</u>	<u>91K</u>	<u>Weight</u>
1. Polynomial Evaluation	26	9.5	138	1
2. Floating-point Arithmetic	18	9	107	1
3. Matrix Multiply	28.5	10	100	2
4. "Clump Finding"	10	6.5	25	3
5. IF Statement	<u>6</u>	<u>6</u>	<u>18</u>	<u>3</u>
Weighted Average	15	7.6	57	

Finally, to arrive at a single figure for effective computation speed of each machine, we took a weighted average of these kernels, using the weights shown in the last column. We felt that these weights would be at least qualitatively correct for the mix of problems expected at SIAC (see, however, remarks in Section 4).

3.4.3 Instruction Set

(a) General. Both machines have instruction sets which can be described as awkward. The S/360, in its attempt to be the "universal machine", has a large set of instructions and is quite complex to code. The 6600, on the other hand, is awkward to program because the instruction set is so elementary.

Both machines have a set of floating-point registers for general computation and a set of fixed-point registers. For both the 6600 and S/360, the assignment of registers is an important but difficult aspect of coding in machine language or of compiling code.

For both the 6600 and the S/360, the ease of writing assembly language programs could be significantly improved by using a less primitive assembly language [30] than that provided by the manufacturer. This is particularly acute in the case of the 6600, whose assembly language reflects the viewpoint of the machine designer rather than of the programmers who must use the machine.

(b) Floating-Point Arithmetic. It would be expected that machines of this class would be excellent numerical computing engines, if nothing else. Surprisingly, both machines have defects in their floating-point operations, leading to unnecessary accumulation of round-off errors.

The S/360 machines have no provision for rounding arithmetic results. At the time of the A54 study there were three other glaring faults in the S/360 floating-point operations. However, IBM has since announced that all of their machines will be modified to correct these faults. The change will be installed on our 360/75 CPU in the field, and on our 360/91 CPU in the factory.

The 6600 performs only unnormalized floating-point addition, with or without rounding. The sum may be normalized with a subsequent Normalize instruction, and in fact this is generally necessary; if unnormalized operands are used as operands in multiplication commands, leading zeros accumulate and significance is rapidly lost. Unfortunately, it is not possible to get floating-point sums both normalized and properly rounded on the 6600.

Double-precision results can be obtained from the Add, Subtract, and Multiply commands of the 6600, but these double-precision results cannot be used as an operand to any subsequent commands. For example, it is not possible to accumulate the double-precision inner product of two single-precision vectors.

(c) Bit, Character, Half-Word Manipulations. The long word length -- 60 bits -- of the 6600 forces a good deal of packing and unpacking of bit and byte fields. Unfortunately, the instructions for this purpose are very elementary, consisting solely of logical operators and shifting. There is no double-length shift of

a pair of registers. We found that programs require about twice as many instructions when written for the 6600 as for S/360; in some circumstances, particularly where character handling is involved, the ratio can be much higher.

S/360 does include a reasonably complete and efficient set of instructions for byte (8 bits, or one character) and half-word (16 bits) operations, including translation and editing of byte strings. The decimal arithmetic facility of S/360 appears to be of limited usefulness to SLAC. The decimal instructions are executed interpretively by the 360/91, so they will execute extremely slowly.

3.4.4 Main Memory

(a) The largest directly-addressable memory with which the CPU can be equipped is:

CDC 6600	128K 60-bit words	1.0 μ sec
IBM Model 75	128K 64-bit words (main)	0.75 μ sec
	1024K 64-bit words (bulk)	8.0 μ sec
IBM Model 91	768K 64-bit words	0.75 μ sec

(b) The CDC 6600 can be equipped with up to 2048K of 60-bit words Extended Core Storage (ECS). This is not directly-addressable memory, and is better thought of as a very high performance (and high cost) I/O device for swapping.

(c) It is interesting to note that the 0.75- μ sec IBM main memory costs 1/3 as much per bit as the 1.0- μ sec CDC main memory.

3.4.5 Interrupts

Both the 6600 and the Model 91 (and, in certain circumstances, the Model 75) suffer from the "imprecise interrupt" problem. This means that when an interrupt -- or its equivalent, in the case of the 6600 -- is caused by execution of an instruction in the program, the location of the particular instruction causing the interrupt is not indicated precisely. If a jump instruction intervened, then the location of the offending instruction is not determined at all. This impreciseness is a consequence of the parallelism of the arithmetic units of both machines.* Imprecise interrupts will unquestionably cause difficulty in debugging programs, particularly system programs.

The Model 91 designers have provided a switch that puts the machine into non-overlapped mode, in which it runs at about Model 75 speed but with precise interrupts. This switch will be important for program debugging, and it can be set and reset by programming.

* Note: However, it is not a necessary consequence and could have been eliminated with some extra hardware in the CPU.

An S/360 computer has a reasonable interrupt scheme. The hardware makes an initial classification into one of five categories, and then one of five software interrupt handlers saves (stacks) machine status, classifies the interrupt, calls a routine to process it, and finally restores machine status.

The 6600 is perhaps unique among modern machines in not having an interrupt system. Interrupts can be divided into three classes:

- (a) Program-Generated Interrupts (error conditions such as references outside memory bounds or arithmetic overflows, and supervisor calls);
- (b) Input/Output and External Interrupts, whose function is to briefly divert the central processor from problem program execution to perform its supervisory function; and
- (c) Machine Check Interrupts, which result from hardware failure.

The 6600 handles the first category by having the CPU simply halt. One of the PPU's ("Peripheral Processing Units") must be programmed to continually run around a loop monitoring whether the CPU is stopped. Because the PPU is relatively slow and has other duties, traversing this loop requires about 200 microseconds in CDC's Chippewa System. Thus the effect is of an interrupt after a delay averaging 100 microseconds. In some circumstances, this delay could be a serious drawback.

The necessity for Input/Output and External interrupts is avoided in the 6600 by performing the entire I/O control function in the PPU's and interrupting the CPU only to switch problem programs via an Exchange Jump. The PPU programs must in turn continuously monitor the status of the devices they control. While this organization obviates the need for interrupts, it has some undesirable consequences which are mentioned below. Finally, the 6600 performs very little hardware checking and consequently has nothing equivalent to a Machine Check interrupt.

3.4.6 Memory Protection and Relocation

The 6600 provides both Read (fetch) and Write (store) memory protection to problem programs in the CPU. This protection is provided both within Central Memory and within Extended Core Storage, but it has two significant limitations. First, protection is implemented by means of a single pair of bounds registers,

so the accessible region must be a contiguous block of storage. As a result, two or more problem programs in 6600 Central Memory cannot share the same reentrant code, without exposing the users lying between the program and the data to possible destruction. Furthermore, since each problem program must occupy contiguous space, loading new programs into memory may require the monitor to move other programs around in memory.

The second limitation is that central memory is unprotected against programs in the PPU's. This fact, together with the non-interruptibility of the PPU's, implies the following:

- (a) If a PPU program gets "lost", it can destroy the system or cycle endlessly and can be recaptured only by doing a "Dead Start";
- (b) Hence, PPU programs are difficult and risky to debug;
- (c) PPU programs are therefore quite unsuitable for communicating with on-line devices and peripheral computers at SLAC, since these control programs are subject to frequent change and might be written by the users themselves. Again, programming difficulties become the major issue in machine organization.

The memory on S/360 computers includes both fetch and store protection, but fetch protection is not supported in the software. We consider both fetch and store protection to be extremely important for programming and operational reasons.

The S/360 decentralized memory protection scheme, using keys in memory rather than registers in the CPU, is (theoretically) more flexible than the 6600 scheme. In particular, the IBM scheme allows a problem program to use non-contiguous memory areas. Unfortunately, IBM's most advanced multiprogramming software (MVT) does not allow this capability to be used (except within a single job). Using MVT, one generally realizes none of the theoretical advantage of the S/360 scheme, and the user would be better off with boundary registers like those on a 6600.

The CDC 6600 has an important advantage over the S/360 machines: it provides hardware relocation of the (contiguous) block of memory occupied by a user. In the case of S/360, programs cannot be moved around in memory once they have been loaded, since relocation is a software function performed at load time. As a consequence, priority roll-out and time-sharing are harder to program on an S/360.

4. The System Selected -- Summary and Retrospect

4.1 The Hardware

Figure 4 shows and Table 1 lists the computer equipment ordered from IBM for the SIAC central computing facility. Since the Model 91K CPU will not be available until the third quarter of 1968, the contract calls for interim rental of a Model 75I CPU. The 91K will be shipped in August 1968 and installed at SIAC on October 1. At the date of this writing (September 1967) the Model 75 and about half of the peripherals in Table 1 have been installed and are operating satisfactorily. The rest of the peripheral equipment will be installed between November 1967 and June 1968.

The configuration in Table 1, which reflects SIAC's current plans, differs in a number of significant details from the configuration contained in the original A-54 study. The changes were triggered by refinements in our planning, experience with the hardware, better understanding of the software, and new software developments.

We will now discuss certain pieces of equipment in the configuration of Figure 4, particularly those which have been changed since the A-54 study. This retrospective analysis of the SIAC computer configuration is offered in the hope that it will be of use to others who may be faced with similar configuration decisions.

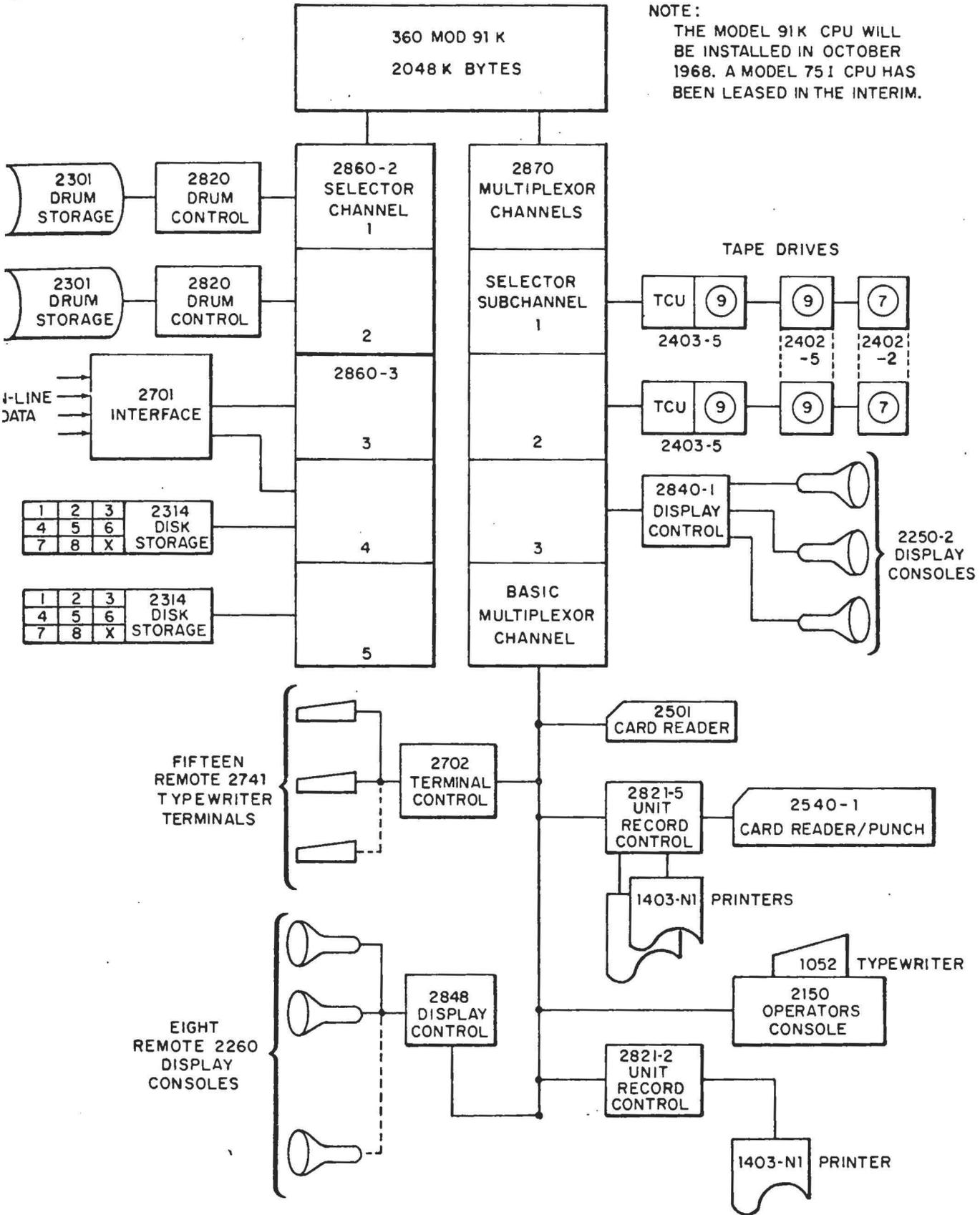


FIG. 4

SLAC CENTRAL COMPUTER, IBM 360/91 FINAL CONFIGURATION

Table 1: SIAC Central Computer, Final Configuration

Description	Quantity	Unit	Model/Feature
<u>Final Processor - October 1968</u>			
CENTRAL PROCESSOR	1	2091	K
PROCESSOR STORAGE (2048K bytes)	1	2395	1
<u>Interim Processor - June 1967 to November 1968:</u>			
CENTRAL PROCESSOR	1	2075	I
PROCESSOR STORAGE (512K bytes total; 256K bytes per unit)	2	2365	3
CONSOLE	1	2150	1
PRINTER/KEYBOARD	1	1052	7
SELECTOR CHANNELS			
(two selector channels)	1	2860	2
(three selector channels)	1	2860	3
MULTIPLICATOR CHANNEL	1	2870	1
Selector Subchannel No. 1	1		6990
Selector Subchannel No. 2	1		6991
Selector Subchannel No. 3	1		6992
M34273 Additional Control Units			RPQ
DRUM STORAGE	2	2301	1
(4 million bytes; transfer rate = 1.25 million bytes/sec.)			
DRUM STORAGE CONTROL	2	2820	1
DISK STORAGE	2	2314	1
(consists of 8 disk drives; capacity = 29.2 million bytes/drive; transfer rate = 312,000 bytes/sec.)			
MAGNETIC TAPE UNIT AND CONTROL	2	2403	5
(9 track)			
Data Conversion	1		3228
Dual Density	1		3471
7 and 9 Track Compatibility	1		7135
MAGNETIC TAPE UNIT (9 track)	1	2402	5
(two tape drives in unit)			
Dual Density	1		3472

Description	Quantity	Unit	Model/Feature
MAGNETIC TAPE UNIT (7 track)	1	2402	2
(two tape drives in a unit)			
Mode Compatibility	1		5122
CARD READER/PUNCH	1	2540	1
(Reads 1000 CPM, punches 300 CPM)			
PRINTER (1100 LPM)	3	1403	N1
Universal Character Set	3		8640
UNIT RECORD CONTROL	1	2821	5
(controls 2540 and 1403)			
Column Binary	1		1990
1100 LPM Adapter	2		3615
Universal Character Set Adapter	1		8637
Universal Character Set Adapter	1		8638
UNIT RECORD CONTROL	1	2821	2
(controls 1403)			
1100 LPM Adapter	1		3615
Universal Character Set Adapter	1		8637
CARD READER (1000 CPM)	1	2501	B2
INTERFACE (on-line data)	1	2701	1
Expanded Capability	1		3815
Parallel Data Adapter	4		5500
Parallel Data Time Out	2		5501
Parallel Data Extension	8		5505
Expansion Feature	3		3855
Channel Interface, 2nd	1		1860
TERMINAL CONTROL	1	2702	1
Data Set - Line Adapter	15		3233
IBM Terminal Control, Type 1	1		4615
2741 Break	1		8055
Expansion Base	1		3853
E46765 Break Recognition	1		RPQ
TYPEWRITER TERMINAL	15	2741	1
Interrupt	15		4708
Typamatic Keys	15		8341
E40681 Break Recognition	15		RPQ
Data Set Attachment	15		9114
Ball Printing Element	15		9571
DISPLAY	3	2250	2
Absolute Vectors	1		1001
Alphameric Keyboard	1		1245
Light Pen	1		4785

Description	Quantity	Unit	Model/Feature
DISPLAY CONTROL	1	2840	1
Absolute Vector Control	1		1003
Buffer	1		1499
Display Multiplexor	1		3351
TEXT DISPLAY STATIONS	8	2260	1
Keyboard	8		4766
TEXT DISPLAY CONTROLLER	1	2848	3
(controls 8 displays, each with 12 lines of 80 positions)			
Display Adapter	4		3357
Display Expansion	1		3857
Line Addressing]	1		4787
Non-destructive Cursor	1		5340
Non-destructive Cursor Adapter	4		5341
Channel Adapter	1		9011

4.1.1 2075I and 2091K Central Processors

Three months of experience with the 360/75 system, combined with additional information from IBM on the 91 CPU, has provided additional insight into the central processors selected for SLAC.

CPU Computation Speeds

The evaluation of the computation speed of the Model 75 using kernels (see Sect.3.4.2.) has turned out to reasonably accurate. For example, large computational FORTRAN codes execute 5 to 10 times faster on the 360/75 than they did on the 7090. The variation is presumably due to differences between 7090 and 360 FORTRAN compilers, and to differences in I/O speeds.

IBM has made some preliminary tests of actual compute-bound job times on a 360/91. The measurements that have been made suggest that (a) the computation speed of the 91 is very sensitive to the details of the code and, even for "heavy floating-point" jobs, may vary from approximately 25 to 60 times that of a 7090; (b) the floating-point kernels of Sect. 3.4.2. are less representative than was hoped, probably because real programs have a much higher percentage of "access-dependent" code (kernels 4 and 5) than we assumed.

Memory Bus Overrun

The relatively high speed of both the Model 75 and the Model 91 CPUs leads to an I/O balance problem, since in the real world of scientific computation there are a great many compilations and debug runs with a great deal of I/O. One solution to this problem is to reduce the amount of I/O by using much larger blocking factors (and hence larger core buffers) and by making all frequently used system routines and control blocks resident in core. In addition, one can multiprogram a number of jobs so as to overlap I/O with computation. Both approaches are necessary to effectively utilize the capacity of a Model 91 (see below), but both reduce I/O waits only at the expense of vastly increased core memory requirements. IBM has adopted both methods to improve the performance of OS/360 on large machines.

However, multiprogramming will help only as long as the average aggregate I/O demand can be satisfied; therefore it is generally necessary to have an array of high-speed I/O devices on the system. This, however, raises a potential hardware problem of whether the memory is designed to handle the necessary aggregate byte rate. The 75I memory does have a theoretical "bandwidth" of eight bytes every .75 microseconds, far in excess of any reasonable I/O demand. Unfortunately, the interface between the memory bus control circuitry of the 75 and the 2860 selector channels is much slower. As originally designed, the Model 75 would overrun if two 2301 drums operated simultaneously with any other I/O activity--even a card reader. However, the priority of the multiplexor channel has been changed to third, below selector channels 1 and 2; as a result it will be possible to run both 2301 drums and both 2314 disk drives simultaneously on the Model 75 and experience only "occasional" overrun.

The overrun problem on the Model 91 is fortunately much less; it will not be possible to produce overrun on the 91 with SLAC's configuration of I/O gear.

Core Memory Size

The contract with IBM calls for the purchase of a 91K CPU. "K" refers to a memory size of 2,048K bytes, i.e., over two million bytes of 500,000 32-bit words. Recent information has indicated that this was a fortunate choice; at present this memory size seems adequate, but we have reason to believe that a Model 91J (one million bytes) would be seriously limited in its capabilities. Considering the system from the batch processing viewpoint, the large memory size is required in order to effectively multiprogram enough jobs so that the CPU will always have something to do in spite of I/O waits. Thus the large memory size is forced by the I/O balance problem discussed above.

Even if a large memory were not required for efficient batch processing, it would be required for time-sharing. The IBM 360 has no fast-swapping capability equivalent to the Extended Core Storage of the CDC 6600. To provide efficient time-sharing, therefore, it is necessary to use the "large-memory" model rather than the "fast-swapping" model. It is important to the success of the TORTOS time-sharing system that some jobs are not swapped out between time slices but rather remain ready in core.

The interim 75 is an "I" machine with 512K bytes. This size was dictated by economic factors, but turns out to be a very serious restriction. The I/O balance problem exists very significantly on the 75, but we do not have enough memory to effectively multiprogram; consequently a great deal of CPU time is spent in "Wait" state. The MFT system with HASP requires 144K or core memory for the system alone. To run the FORTRAN H level compiler requires 228K in the batch partition. This leaves only 140K for graphic work and, in many cases, this is not enough memory to use the 2250 displays effectively for physics codes.

One suggested solution to the memory shortage on the Model 75 was to rent a bulk core memory. Several installations with Model 75s have invested considerable software effort in using bulk core instead of a drum for system residence. This improves performance, particularly the system overhead in short jobs; however, it offered no help to our memory in shortage problem. It is not possible to do 2301 drum I/O in bulk core, for example.

There is some reason to hope that the TORTOS time-sharing system will help solve the memory shortage on the 75 by swapping. The performance in terms of response time on the Model 75 is likely to be poor, but we should be able to operate the 75 much more effectively and schedule graphic and other on-line activities more flexibly using it. It is our feeling that a large scale scientific computing installation using the standard IBM OS/360 software would not be well advised to purchase a Model 75 with less than one million bytes.

4.1.2 2860 Selector Channels

Subsequent to the completion of the A-54 study, IBM pointed out two important restrictions on selector channels: the Model 75 can have at most two 2860 boxes attached, and the Model 91 can have at most five selector channels. Maximizing the number of selector channels on the system has seemed an increasingly urgent requirement because (a) we have found that the design of the S/360 hardware and

and software makes interference among devices sharing a common channel a serious problem, (b) the I/O channel capacity will be a limiting factor in utilization of the full capability of the Model 91 CPU, and (c) the decision to enter on-line data directly into the big machine (through a 2701) without an intervening buffer computer places severe requirements on the I/O channel capability of the machine. As a result of these considerations, it was decided to increase the number of selector channels from the four shown in the A-54 study to five, the maximum number possible under the hardware constraints mentioned above.

The restriction to five selector channels is likely to be a problem in the asymptotic computer configuration. Several installations which have ordered Model 91's have considered RPQ's to add additional selector channels to their machines. At some time (probably two years hence) SIAC may have to consider such an RPQ for its machine.

4.1.3 2702 Model 1 Terminal Controller

The 2702 was originally configured with Limited Distance Line Adapters, which include IBM modems (data sets). Since most of the 2741's at SIAC are expected to be placed within 2,000 feet of the central computer, we decided to use the D.C. hook-up scheme devised at the Medical School Facility of the Stanford Computation Center. This requires the Data Set Line Adapters on both ends (2702 and 2741) of the line, although no data sets are used.

The Break Recognition RPQ allows the computer to regain control of the terminal while it is in type-in state. This feature is required for the TORTOS time-sharing system which we expect to run on the Model 75 and Model 91.

4.1.4 2321 Data Cell Drive

The A-54 study included two 2321 data cell drives, providing bulk storage for 400 million bytes of information. It was subsequently learned that the 2321 is engineered for low duty-cycle applications; with heavy use it may require excessive mechanical maintenance. It was decided that the data cell is not well suited to the SIAC application. As a storage place for user program files under the time-sharing system, the 2321 is likely to be subjected to an excessive duty-cycle. It is not large enough for long-term event storage (see Section 2.2.3), and its slow access and low transfer rate make it poorly suited to SUMX. Therefore we decided to replace the two data cell drives in the original configuration with a second 2314 Disk Storage Unit. The characteristics of the two devices are compared in the following table.

	<u>Transfer Rate</u>	<u>Access Time</u>	<u>Capacity</u>
2321 Data Cell	312,000 bytes/sec.	.1 to 650 ms	400 million bytes
2314 Disk Storage	55,000 bytes/sec.	25 to 160 ms (87 average)	232 million bytes

Another significant advantage of a second 2314 is that the system disk packs can then be split between the two 2314 units and therefore between two channels and two control units. An important property of the S/360 direct access devices is that the channel and control unit are tied up during the rotational latency, an average of 12.5 milliseconds on the 2314. Therefore it is vital to good system performance to have as many separate channels and control units as possible.

4.1.5 2820 Drum Controller

The original A-54 configuration contained two high-speed 2301 drums on one 2820 control unit. We subsequently learned about the TORTOS time-sharing system and decided that we should plan to install it on the SLAC computer. TORTOS dedicates a drum, its controller, and its selector channel to continuous swapping. It keeps a swapping channel running continuously with a loop channel program. In this situation the second 2301 drum without a second 2820 controller was of little use to us. We therefore ordered a second 2820. The second drum may be now used either for system residence or as a second swapping drum; we will need experience with the Model 91 system and with TORTOS to determine which use will improve system performance more.

4.1.6 2701 - External Device Interface

At the time of the A-54 study, our planning for the on-line use of the large computer was in a very early stage. Since then, considerable experience has been gained with the on-line operation of Van der Lans' CRT film scanners connected to the Model 50 through a 2701. These scanners have now been connected to the Model 75. In addition, an IBM 1800 and another small computer are due to be connected to the Model 75 shortly. These considerations led us to re-configure the 2701 with the maximum possible number and width of data paths -- four ports of 32 bits each into the central processor. The initial assignment of the four ports is planned as follows:

- Port 1: CRT Film Scanners.
- Port 2: Graphics Station - A complex including a plotter, a microfilm recorder, a display, and a film digitizer, all run by a small computer.
- Port 3: A 2000 foot cable connection to an IBM 1800 in the experimental area.
- Port 4: A second IBM 1800 to be used for research in graphic data processing.

4.1.7 2501 Card Reader

The programming language PL/1 which we expect to be used with increasing frequency at SLAC requires 60 distinct characters. No manufacturer builds an electro-mechanical machine (like an IBM 407, for example) capable of printing more than 48 distinct characters; however, our users will want to list their program decks with the full character set. Therefore, we intend to use one of the 1403 printers and the 2501 card reader for listing user decks. The 2501 card reader is suitable for this purpose since it is relatively inexpensive, includes an integral control unit, and is of such a design (photoelectric reader) that it should be relatively reliable for hands-on operation by users.

4.2 The Software

Section 3.2 listed a number of desirable properties for the software as well as a basic set of software packages. It was felt that the programming packages specified in Section 3.2 are a minimal set essential to the operation of a machine in the class of a Model 91. It turned out, however, that a significant subset of this minimal software is missing from IBM-supplied software. The facilities which SLAC requested but which are currently absent are discussed in the following.

(1) Operating System: Missing are facilities for interrupting low priority jobs to run higher priority jobs ("roll-in/roll-out"), and terminating jobs which exceed output estimates. In general, OS/360 is very weak in provisions for control or data gathering by the installation management and the computer operator.

(2) Time-Sharing Executive Program: IBM has no official software product for time-sharing a standard 360 in a large-scale scientific environment. There is, however, a system called TORTOS being written for UCLA which SLAC hopes to use. TORTOS is a modification and extension of the full multiprogramming (MVT) version of OS/360.

(3) FORTTRAN Compiler: Although IBM provides two reasonable FORTRAN compilers, known as level G and level H, neither is reentrant, nor does either produce reentrant code. One of them (G level) provides source language debugging while the other forgives irrelevant errors, but neither does both. There is an effort within IBM to improve the H level compiler further.

(4) ALGOL Compiler: IBM does not support their ALGOL 60 compiler for the MVT system. This compiler has an awkward input language -- it requires quotes around reserved words and has no square brackets. However, IBM does supply a PL/1 compiler which we expect to use in place of ALGOL.

- (5) List Processing System: There is none provided by IBM.
- (6) Utility Routines: There are no file maintenance utilities suitable for maintaining source programs, data files, etc., in a time-sharing environment.
- (7) Linking Loader: IBM provides a very comprehensive "linkage editor", but it is designed for small machines ("E Level") and presents a performance problem.

The situation in the software area has changed significantly since SLAC began the selection process. These changes include the following.

- (1) IBM withdrew a number of crucial facilities in their full multiprogramming system. These withdrawals included roll-in/roll-out and non-contiguous memory. IBM also delayed the scaled-down MVT system, but it turned out that the results for SLAC were not serious.
- (2) IBM produced a simple subset operating system called MFT or the "partitioned system." This system runs a batch stream purely sequentially in the lowest priority partition of memory. Long-running jobs such as those which use graphic displays can be scheduled in higher-priority partitions. Experience with the system over three months has shown that it is extremely awkward and inefficient for the kind of load existing at SLAC.
- (3) The situation on the Model 75 was saved by an unofficial IBM software product called HASP (Houston Automatic Spooling Program). This system runs in the highest priority partition of MFT, spools input and output onto a disk, and executes jobs sequentially in priority order. It drives a number of card readers and printers simultaneously with very little CPU overhead, and provides an excellent "warm start" capability and a good operator interface. In fact, HASP is so good that we fear that switching to MVT may be looked upon as a step backwards by our users.
- (4) We learned of the TORTOS time-sharing system being written for UCLA. This is a general time-sharing system giving the user access to all the facilities of OS/360 but providing swapping and conversational interaction.

Appendix: Additional Computers in Use at SLAC

IBM 1800

Date Acquired: June 1967.

Purpose: Used in filmless graphic data analysis of wire spark chamber experiments. The computer performs real-time, first-order analysis on the data as it is gathered and logged onto tape or disk.

Configuration: 32K 16-bit words of 2 μ sec core storage, 9-track magnetic tape unit, disk storage unit, card reader/punch, printer, special I/O channels.

SDS 925

Date Acquired: December 1965.

Purpose: To monitor and control magnets in the Beam Switchyard, complement data collection by the spectrometers, and monitor accelerator beam status.

Configuration: 8K 24-bit words of core storage, card reader, card punch, typewriter, analog-to-digital converter presently with 25 channels, digital-to-analog converter presently with 16 channels; future expansion of the A/D and D/A channels is easy and inexpensive. There is a data link to the SDS 9300 computer.

System Status: The major components of the system are in operation, but some features are being revised. The SDS 925 was used in the initial testing of the accelerator beam.

SDS 9300

Date Acquired: July 1965 by Research Group A.

Purpose: Acquisition and analysis of data produced by the three spectrometers at End Station A. Results are recorded on magnetic tape for subsequent off-line processing.

Configuration: 32K 24-bit words of core storage, three 7-track tape drives, teletype, disk file with two million character capacity, card reader, printer, card punch, plotter, scope, multiplexor with 2000 subchannels, priority interrupt system with 32 interrupt levels.

System Status: Presently 60% of the disk monitor is completed. Pending completion of the disk monitor, the SDS AD-2 tape monitor is still in use.

PDP-8

Date Acquired: September 1966 by Research Group E.

Purpose: The computer is used to analyze data in a μ -p scattering experiment. Data collected by the counters is displayed on an oscilloscope for human observation. Spark chamber photographs are taken of events of interest (as

determined by counter coincidences). The computer lists, as part of its output, the counters in coincidence for each event.

Configuration: 4K 12-bit words of core storage, 12-bit high-speed data channel, teleprinter, oscilloscope display unit, magnetic tape unit.

PDP-9

Date Acquired: October 1967 by Research Group B.

Purpose: To control and to acquire data from the Spiral Reader (a mechanical bubble chamber film digitizer).

Configuration: 8K 16-bit words of core storage, one direct access to memory, four high-speed data channels, four levels of automatic priority interrupts, teleprinter, paper tape reader/punch.

ASI 6020 *

Date Acquired: October 1966 by Research Group B.

Purpose: To control and monitor bubble and spark chamber film measurement.

Results are recorded on magnetic tape (in card image format) for further processing.

Configuration: 8K 24-bit words of core storage, card reader, 10-million bit capacity drum storage, multiplexor with nine channels, 32 scan interrupts, eight teletypes, display lights for each scanning table, Kennedy tape recorder, Datamac tape recorder.

* Note: The name of the company manufacturing this computer has been changed to EMR Computer Division.

REFERENCES

- [1] E. Burfine and W.F. Miller, "Computation Requirements for SLAC", SLAC Computation Group Staff Paper, June 9, 1965.
- [2] R.H. Miller, "Some Considerations Regarding Computers at SLAC", SLAC TN-64-52, June 1964.
- [3] W.F. Miller, "Status and Immediate Future of Computer Development for Nuclear Physics Applications", Proc. Conf. on Automatic Requisition and Reduction of Nuclear Data", Kernforschungszentrum, Karlsruhe, July 13-16, 1964. Published by Gesellschaft für Kernforschung m.b.H. Karlsruhe.
- [4] J.C.R. Licklider, "Man-Computer Partnership", International Science and Technology, No. 41, p. 18, May 1966.
- [5] C. Moore, S. Howry, H. Butler, "TRANSPORT", SLAC Internal Report, May 1963.
- [6] S. Howry, "Solenoid Optics", SLAC Computation Group Program Report, 1964. (This program is a very specialized ray-tracing program for tracing through a series of closely-spaced solenoids).
- [7] S. Howry, "CONFORMAL MAP", SLAC Computation Group Program Report, 1964. (This program computes contour lines as a function of pole face shape).
- [8] C.D. Zerby and H.S. Moran, "Studies of the Longitudinal Development of Electron-Photon Cascade Showers", J. Appl. Phys. 34, p. 2445, August, 1963 and "A Monte Carlo Calculation of the Three-Dimensional Development of High-Energy Electron-Photon Cascade Showers", ORNL-TM-422, December 3, 1962.
- [9] H. Negel, Dissertation, University of Bonn, 1964.
- [10] C. Moore, "CURVE", SLAC Computation Group Program Report, 1964.
- [11] William C. Davidon, "Variable Metric Method for Minimization", ANL-5990 REV, AEC Research and Development Report, November 1959.
- [12] H. Pierre Noyes, "Determination of the Proton-Proton 1S_0 Shape Parameter", Phys. Rev. Letters, 12, 171 (1964).
- [12a] H. Pierre Noyes, "The Interaction Effect in n-p Capture", Nuclear Physics, 74, 508 (1965).
- [13] Martin L. Perl and Mary C. Corey, "Empirical Partial-Wave Analysis of $\pi + p$ Elastic Scattering Above 1 GeV/c"; Phys. Rev., 136B, 787 (1964).
- [14] A.C. Hearn, "Symbolic Computation of Feynman Graphs, Using a Digital Computer", Bull. A.P.S. II, 9, 436, (1964).
- [15] A.H. Rosenfeld and W.E. Humphrey, "Analysis of Bubble Chamber Data", Ann. Rev. Nucl. Sci. 13, 103, (1963).

- [16] Frank T. Solmitz, "Analysis of Experiments in Particle Physics", *Ann. Rev. Nucl. Sci.* 14, 375, (1964).
- [17] F.T. Solmitz, A.D. Johnson, and T.B. Day, "Three View Geometry Program", P-117 Alvarez Group Programmer's Note, University of California Lawrence Radiation Laboratory, March 25, 1965.
- [18] Several of the changes planned in bubble chamber analysis codes were suggested by Mr. J.J. Merkin of IBM.
- [19] W.F. Miller, J. Meadows, A. Smith, and J. Whalen, "On-Line Data Acquisition with Feedback to Accelerator Control", *Proc. Conf. on Automatic Acquisition and Reduction of Nuclear Data*, Kernforschungszentrum, Karlsruhe, Germany, July 13-16, 1964, p. 222.
- [20] "The N.B.S. On-Line System - Design for Flexibility", *Nucleonics*, 22, 57 (1964).
- [21] S.J. Lindenbaum, "The On-Line Computer Counter and Digitalized Spark Chamber Technique", *Physics Today*, 18, 19, (1965).
- [22] "Future SLAC Computational Facilities", SLAC Computation Group Staff Paper, June 1, 1964.
- [23] D. Hartin, et al, " π^- -p and p-p Elastic Scattering at 8.5, 12.4, and 18.4 GeV/c", in *Press Nuovo Cimento*, pp. 15-18, of preprint.
- [24] R. Clark and W.F. Miller, "Computer Based Data Analysis Systems", *Methods in Computational Physics*, Vol. V., Academic Press, 1966.
- [25] Richard Royston, Argonne National Laboratory, Private Communication.
- [26] Arthur H. Rosenfeld and William E. Humphrey, "Analysis of Bubble Chamber Data", *Annual Review of Nuclear Science*, 13, 103, (1963).
- [27] Fred Martin, Stanford Linear Accelerator Center, Office Memorandum, July 31, 1964
- [28] Fowler, University of Maryland, Private Communication.
- [29] H. Nelson, "Program Optimizing Techniques for the CDC 6600 Central Processor", Lawrence Radiation Laboratory, UCRL-12489 (April 6, 1965).
- [30] An experimental Algol-like assembly language for S/360 computers is described in:
N. Wirth, "A Programming Language for the 360 Computers", Technical Report CS33, Stanford Computer Science Department, Dec. 1965.
- [31] R.M. Brown, M.A. Fisherkeller, A.E. Gromme, and J.V. Levy, "The SLAC High-Energy Spectrometer Data Acquisition and Analysis System", *Proceedings of the IEEE*, 54, 12 (Dec. 1966).
- [32] R. Russell, "Use of the 1800 Computer in On-Line Spark Chamber Experiments", GSG Memo #44, Computation Group, Stanford Linear Accelerator Center.
- [33] S. Howry, "A Concise On-Line Control System", SLAC-PUB-248, Stanford Linear Accelerator Center, October 1966.