RESTRICTED

Copy Nº

AMP Report 171.2R

Moore School of Electrical Engineering
University of Pennsylvania.

DESCRIPTION OF THE ENIAC AND COMMENTS ON ELECTRONIC DIGITAL COMPUTING MACHINES

(AMP

Distributed by the
APPLIED MATHEMATICS PANEL
NATIONAL DEFENSE RESEARCH COMMITTEE

This document contains information affecting the national defense of the United States within the meaning of the Espionage Act, 50 U. S. C. 31 and 32. Its transmission or the revelation of its contents in any manner to an unauthorized person is prohibited by law.

November 30, 1945

RESTRICTED

November 30, 1945

MOORE SCHOOL OF ELECTRICAL ENGINEERING
UNIVERSITY OF PENNSYLVANIA

DESCRIPTION OF THE ENIAC

and

COMMENTS ON ELECTRONIC DIGITAL COMPUTING MACHINES

This Report Was Prepared By

J. P. Eckert, Jr.
J. W. Mauchly
H. H. Goldstine
J. G. Brainerd

Contract W 670 ORD 4926

The increasingly important role played by computing in analysis led the Applied Mathematics Panel, NDRC, to set up a study to provide for a survey of the most important computers, both continuous and digital. The electronic computers, EDVAC and ENIAC, have been constructed at the Moore School of Electrical Engineering, University of Pennsylvania, under Contract W 670 ORD 4926 with the Army Ordnance Department. This report on the ENIAC and other electronic digital computers, prepared by members of the staff of the Moore School, is being distributed by the Panel with the approval of the Office of the Chief of Ordnance.

TABLE OF CONTENTS

PREFACE

The ENIAC (Electronic Numerical Integrator and Computer) is a device now in the test stage. It is not in a state in which a report for general distribution is warranted. The present report is supplied to the Applied Mathematics Panel for limited circulation among a particular group, the members of which have a special interest in computing machines. The contents should be considered in the light of the preceding statement, with the assurance that technical descriptions will be published in technical journals as soon as it is felt that this is justified.

The EDVAC (Electronic Discrete Variable Computer) is in an early design stage, and remarks concerning it are subject to future change as work progresses.

Of the authors of this report, Mr. Eckert has been chief engineer of the project and deserves credit for most of the technical work. Dr. Mauchly has assisted him. Captain Goldstine as liason officer of the Ballistic Research Laboratory, has been actively engaged in the project from its inception. The project originated in a memorandum prepared by Dr. Mauchly at a time when the BRL work at the Moore School was so extensive that everyone was impressed by the great desirability of a high-speed, high-capacity computing device, faster than any then available.

It is desired at this point to make certain acknowledgments:

To Colonel Leslie E. Simon, Director of the Ballistic Research Laboratory, Aberdeen; Colonel Paul N. Gillon, Chief of the Research and Development Section of the Army Ordnance Department; Mr Sam Feltman Chief Engineer of the Research and Development Section of the Army Ordnance Department; Dean Harold Pender of the Moore School, and Mr. Miles E. Nelson, supervisor of the Moore School shop, thanks are due for hearty and extensive cooperation.

To Mr. S. B. Williams, of the Bell Laboratories, the project is indebted for many helpful suggestions on the relay circuits and considerable aid on many occasions in supplying detailed information on telephone parts which proved useful in the ENIAC.

The project also wishes to thank Mr. A. H. Dickenson and Mr. John Wheeler, of the IBM Company, for designing the special card reading machine and for suggesting circuits for connecting it to the relays in the ENIAC. Thanks are also due to the IBM Company for supplying machines and auxiliary parts.

Livingston and Company, Incorporated, of Philadelphia, went out of its way to cooperate in expeditiously fabricating certain parts for the ENIAC; for this our thanks are extended.

To Eggly Engineers the project is indebted for design of the ventilating system which is an integral part of the ENIAC.

Engineers, mathematicians, report writers and others who have worked at any time in a design or supervisory capacity on the project or otherwise made a substantial contribution to it are the following:

Arthur Burks
C. C. Chambers
Joseph Chedaker
Chuan Chu
James Cummings
Leland Cunningham
John Davis
Harry Gail
Adele Goldstine
Irwin Goldstein

Harry Huskey
Hyman James
Edward A. Knobelauch
Herman Lukoff
Robert Michael
Frank Mural
T. Kite Sharpless
Robert Shaw
C. Bradford Sheppard
Homer Spence

S. R. Warren, Jr.

The project is particularly indebted to the above persons for their contributions.

Thanks are due to Professor Hans Rademacher of the Mathematics Department of the University of Pennsylvania who carried out for the project some as yet unpublished studies concerning the optimum interval to be used in the solution of certain sets of equations in order that the expected error from round-off and truncation shall be a minimum.

Professor John Von Neumann, of the Institute for Advanced Study, is an active participant in the planning of the EDVAC, especially with regard to its overall logical organization. Thanks are certainly due to him for his continued work and interest in the project.

J. G. Brainerd.

DESCRIPTION OF THE ENIAC
and
COMMENTS ON ELECTRONIC DIGITAL COMPUTING MACHINES

INTRODUCTION

In recent years various large machines have been devised for carrying out in more or less automatic fashion the numerical solution of mathematical problems which could hardly be undertaken without such machines. The Moore School of Electrical Engineering of the University of Pennsylvania has, for the past two and one-half years, been engaged in the development of such machines for the Ballistic Research Laboratory of the Aberdeen Proving Ground.

The ENIAC, described in this report, is the first general purpose automatic electronic digital computing machine. Its speed considerably exceeds that of any non-electronic machine, and its accuracy is in general superior to that of any non-digital machine (such as a differential analyzer).

The ENIAC is extremely flexible, and is not fundamentally restricted to any given class of problems. However, there are problems for which its speed is limited by the input and output devices, so that it is impossible to derive the full benefit of its high computing speed in such cases. The ENIAC carries out its entire computing schedule automatically, but the sequence which it is to follow must be set up manually beforehand. The intended use of the ENIAC is to compute large families of solutions all based on the same program of operations, in which case the time spent in manual set up can be disregarded.

A second electronic digital machine, the EDVAC, is now being planned. It will be of larger capacity than the ENIAC, have a somewhat higher computing speed, and will be completely automatic, including set up. Despite these features, it will require considerably <u>less</u> equipment than the ENIAC, since the electronic components will be used in a quite different and much more efficient way.

The nature of the problems which these machines bring within the range of computation, and the real necessity for speed in carrying out such computations, are covered in the first chapter of this report. Following this, chapter 2 points out that to attain the speed, accuracy and flexibility required, electronic digital machines must be used.

A description of the ENIAC in Chapter 3 is intended to give the reader a rather complete account of the general features of the ENIAC and also provide him with some information on the way in which the various units can be used. The important function of control is considered first, then various kinds of memory or storage facilities, then the arithmetic units, and finally the input and output devices. (Specific explanations of some arithmetic and programming techniques will be found in the appendices; there is also an appendix giving constructional data.)

The later chapters of this report are concerned with a discussion of some general principles which seem pertinent to computing machine design and which have been used in formulating the plans for the EDVAC.

It should be noted in this introductory section that it is recognized that the object of computing machine design is not merely to speed up arithmetic processes, but to attain a high overall speed, including the problem set-up and the preparation of results in useful form. It is desirable to have as much as possible done automatically.

It is also to be observed that a great deal of the equipment in non-electronic machines is "in multiple", that is, concurrent operation of many parts is used to increase computing speed. Electronic devices are inherently so fast that it is unnecessary to achieve speed in this way. By resorting to "serial operation", a considerable saving in equipment may be secured.

the consequent loss in speed is tolerable only when electronic components having high inherent speed are employed. Reliability and maintenance are aided by this equipment reduction, and serial operation also has important advantages both from the point of view of checking and because it simplifies the work of planning the computational program.

CHAPTER 1

THE NEED FOR HIGH SPEED GENERAL PURPOSE COMPUTING MACHINES

1.1 Need for General Purpose Computing Machines

Before proceeding to a description of the ENIAC there will first be given a brief indication of the need for truly high-speed computing machines in the applied sciences and specifically the needs of the Ballistic Research Laboratory, Aberdeen Proving Ground, for such machines. The discussion of these points will be brief since it is believed that von Neumann is planning to devote considerable space to them in an AMP report soon to appear.

An examination of the literature of the physical sciences shows that the principal emphasis in these fields has been in the solution of linear problems (those which can be formulated in terms of linear equations) which can be handled by analytic techniques. Physical problems different from these are not necessarily more difficult from a physical point of view, but they have been by-passed in favor of the problems whose analytic solutions are possible of attainment.

Those problems which cannot be solved analytically have been handled by computational methods or through the use of specific analogy machines. As an illustration of the computational approach we might mention the truly remarkable work of Hartree on the structure of the atom, a series of calculations extending over a period of about 15 years. An exemplification of the latter technique is found in the use of wind tunnels. At present, the supersonic wind tunnel at the Ballistic Research Laboratory is used about 30 % of the time as an analogy machine to solve two-dimensional steady state aerodynamical problems. Industrial companies frequently resort to highly specific analogy machines to solve, for example, linear equations of electric circuit theory or the partial differential equations which enter into electron-optics problems. It may be noted that much of present experimental work consists essentially of the solution of mathematical problems by analogy methods. If one had a computing machine of sufficient flexibility the necessity for these experiments would be obviated provided that it was sufficiently rapid to solve the problems in a length of time competitive with that of experimentation. Furthermore, with computing machines whose speeds were orders of magnitude faster than experimental methods, not only would progress in mathematical physics and engineering be accelerated, but, undoubtedly, research would also be extended into fields as yet unexplored.

To handle efficiently large classes of problems which can be mathematically formulated, automatically sequenced general purpose computing machines are needed. The machines thus far developed or, at present, contemplated are sufficiently general to solve problems whose complexity is comparable to those of two-dimensional transient or three-dimensional steady state aerodynamical motion. Such machines will stimulate the investigation of many problems, otherwise computationally unapproachable, in many other fields. We mention only a few of the more obvious: quantum mechanical and electrodynamical studies, molecular and statistical studies in chemistry, astrophysical applications, exploratory investigations in theoretical meteorology as well as researches in mathematical statistics and mathematics itself.

1.2 Need for Speedy Computing Machines

In discussing the speed of computing machines it is perhaps desirable to distinguish between so-called continuous variable and digital machines. Although existing continuous variable machines such as the differential analyzer and the a-c network analyzer are exceedingly rapid, the class of problems which they can solve is limited. Since both of these machines perform all operations of a computation in parallel, the size and complexity of the problems they can solve is limited by the number of arithmetic organs they contain. In addition, these machines are necessarily restricted in their accuracy due to their inherent nature. We will, therefore, confine our attention to digital computing machines.

The time required to carry out a multiplication provides a rather good index of the speed of a digital computing machine inasmuch as the time spent in multiplication usually represents the major part of the computing time when such machines are used. In general, the computing time for a given problem can be estimated roughly as 2 or 3 times the amount spent in multiplications. The multiplication time for the ENIAC is about 3 msec, for the EDVAC, about one msec. For electromechanical digital machines, the multiplication time ranges from about 1/2 second to 5 seconds.

To give some idea of the time consumed in carrying out an extensive computation, it might be well to consider a typical aerodynamical problem of interest to the Ballistic Research Laboratory, which involves the solution of quasi-linear hyperbolic differential equations.

I. E. Segal has formulated the partial differential equations for describing the pressure on a high-speed, sharply pointed and non-yawing projectile. The knowledge of this pressure enables one to compute the head drag, the air flow between the projectile and the head wave, and of course, the head shape of lowest drag in a given group of head shapes. If the mathematical solution of this problem is successful, it will evidently point the way to the solution of many problems now handled in wind tunnels or precision firing ranges as experimental problems. We shall then describe Segal's problem and his method of solution as a typical example in mathematical physics illustrating the need for high-speed computing equipment.

We assume as given the shape of the head wave and compute from this the shape of the projectile as well as the flow. The partial differential equations describing the motion of the fluid are quasi-linear hyperbolic partial differential equations wherever the flow is supersonic, and they are solved by the introduction of characteristic parameters. It is assumed by Segal in the derivation of his equations that the air is a compressible and non-viscous, as well as an isentropic, fluid. The considerable simplification brought about by taking the head wave as given, however, results in the inconvenience of having to interpolate in a family of previously computed head waves to determine the flow around the given projectile head at a given Mach number. Segal estimates the number of head waves needed for this interpolation to be in the order of 25.

Before the introduction of characteristic parameters the form of equations to be solved is

$$(a^2 - u^2)\, u_y - uv\, (u_y + v_x) + (a^2 - v^2)\, v_y + \frac{a^2 v}{y} = o,$$

$$v_x - u_y = o,$$

where a is the local velocity of sound; u, v are the components of the fluid velocity at the point (x,y), and where the subscripts denote the partial differentiation.

Without going into the details of the step-by-step integration method used by Segal, we can say that there are approximately 90 multiplications to be performed at each point in the field of integration, and that there will be about a thousand such points. Consequently, on the ENIAC the pure computing time, apart from reading or printing, will be about 5 minutes. As will be seen in the subsequent discussion, the ENIAC has at least two types of memory; a fairly high-speed memory, used locally in connection with the arithmetic operations, and an indefinitely large memory capacity on IBM cards. The machine is equipped both to read and punch such cards. In this problem it appears necessary to use this slow-speed memory to store the results of previous computations and then to reintroduce these data at a later time for subsequent calculations. In fact, it may be necessary to read 2 cards and to print 2 others for each point. The reading and printing however, can probably be done concurrently. It is therefore sufficient to consider the printing time alone. Inasmuch as there will be 2000 cards printed at a rate of 100 cards per minute, the printing time is of the order of 20 minutes. It is thus seen that the total solution for Segal's problem for a given Mach number and a given head wave will be, with ideal ENIAC operation, about a half hour.

As was remarked earlier, it will probably be desirable to do about 25 solutions of this problem for different head waves and possibly about 10 solutions for different Mach numbers so that the total problem will be of the order of 250 solutions of the equations. This would represent, under ideal operating conditions, about 125 hours of total computing to provide a reasonably complete table for the determination of the drag on an ordinary shaped projectile head moving at supersonic speeds. The memory of the EDVAC is planned to be much larger than that of the ENIAC, and the calculated solution time of Segal's problem on the EDVAC is about two minutes for each head wave and Mach number. It is thus seen that on the EDVAC, the total time of solution for the complete problem will be of the order of 8 or 9 hours. This great speed is possible primarily because of the very large high-speed memory of the machine.

Let us contrast this computing time with that on a hypothetical electromechanical machine. Assuming a multiplication time on such a machine of about one second and a printing time of 5 characters per second, there would be spent 25 hours for the pure computing time and about 8 hours for printing making a total computing time of about 33 hours. Consequently, the total time of solution, about 8,000 hours, is sufficiently long so that one would not handle the problem by computational means, but rather would be forced to handle the problem by experimental techniques either in supersonic wind tunnels or precision firing ranges of the sort in use at the Ballistic Research Laboratory.

We should perhaps next inquire into the reasons why one is not content to take extremely long times to carry out a numerical computation. The three arguments on this point are cost per problem, the necessity for getting a solution quickly, and the scope and range of problems which one could and would be willing to undertake. The second point is important not only because of deadline considerations but also because the high-speed machine can be used as a tool of the physicist or engineer, much as his other laboratory equipment is used. The importance of this consideration cannot be over-emphasized since it will allow the scientist to undertake investigations which he would otherwise not even attempt.

1.3 The Needs of the Ballistic Research Laboratory

The Ballistic Research Laboratory is charged by the Ordnance Department with the two responsibilities of carrying out research leading to the development of better gun-projectile-propellant combinations and of producing firing and bombing tables for proposed or existing combinations. In general, the former activity results in computational problems of an aero- or hydro-dynamical nature, whereas the latter gives rise to simple but exceedingly laborious and extensive calculations. This dichotomy in the computing work of the Laboratory has been reflected in the designs of the ENIAC and the EDVAC, as we shall indicate later.

Although the Laboratory utilized both its own and the Moore School of Electrical Engineering's differential analyzers, assembled a staff of about 200 computers and organized an extensive IBM group, it was felt that even this computing facility would be inadequate to provide the combat services with the firing and bombing tables that were so urgently needed. Accordingly every effort was made to seek entirely new tools for this purpose.

In August, 1942, J. W. Mauchly had summarized briefly, in memorandum form, the advantages to be expected from an electronic high-speed computer of the type that could reasonably be developed at that time. Early in the spring of 1943, Captain Herman H. Goldstine of the Ballistic Research Laboratory, and Colonel Paul N. Gillon of the Office of the Chief of Ordnance, became interested in the possibilities of such a device for carrying out the preparation of firing and bombing tables. At Goldstine's request, Mauchly and J. P. Eckert, Jr. wrote a tentative technical outline of a machine which would be capable of numerically integrating trajectories for firing tables and which would handle other computing jobs of similar complexity. This material was then, April, 1943, included by J. G. Brainerd in a report which formed the basis for a contract between the University of Pennsylvania and the Government to develop an electronic device along these lines. The project was set up under the supervision of Brainerd, with Eckert as chief engineer and Mauchly as principal consultant; Goldstine was appointed resident representative to take technical cognizance of the project for the Ordnance Department.

Under this contract a staff of the Moore School has, since 1 July 1943, worked continuously on the design and construction of the ENIAC. The construction of this machine is at the present time just completed and an acceptance test is in process. It is worth remarking that the final form of the ENIAC represents a change from the initial plans which contemplated primarily a device for solving non-linear total differential equations. These changes were intended to increase the generality of the device so that the ENIAC could be used as a (reasonably) all-purpose digital device. It is still, however, most efficiently usable as a machine for handling fairly simple computations which must be repeated many times, as is the case in making firing tables.

This property stems, to some extent, from the pioneering nature of the work. To make possible simple calculations with a unit, as it was completed, each was designed so that it contained not only its arithmetic facilities but also an extensive group of logical controls. Possibly another factor which limits the generality of the ENIAC is the type of high-speed memory used. It was not until early in 1944 that Eckert and Mauchly invented a device which permitted in a practical manner an increase in memory of two orders of magnitude.

When this new advance in the art was made, a new contract was entered into between the Government and the University for the development of the EDVAC (Electronic Discrete Variable Calculator), an automatically sequenced, all-purpose electronic computing machine. It is expected that this new machine will be useful as a basic research tool for the Laboratory's investigations, e.g., in aerodynamics and interior ballistics.

The development of this machine is still in its early stages since the staff, until recently occupied with the construction of the ENIAC, is also designing the EDVAC. They are being assisted by von Neumann in planning the logical aspects of the EDVAC. The administrative supervision is being handled by Dr. S. R. Warren, Jr.

1.4 Scope of Problems That Can Be Handled on the EDVAC.

It might be well here to clarify the "all-purpose" character of the EDVAC. It is expected that this machine will make possible an exploration of the purely mathematical theory of non-linear differential equations. It will also make possible a computational approach to the various physical theories which were mentioned above in Section 1.1.

Of particular importance to the Laboratory is the possibility of considering non-viscous, isentropic and compressible hydro- and aero-dynamical problems, as well as the more difficult problems in the theory of shock waves. Also of great importance to the Laboratory is the studying of elastic and plastic properties of materials.

It is probable that these machines will necessitate a complete reorientation of thinking about computational techniques. The current techniques of numerical approximation are based on an "economy" of numerical operations which may not be pertinent in selecting the procedures to be followed on the ENIAC or the EDVAC.

CHAPTER 2

ADVANTAGES OF ELECTRONIC DIGITAL MACHINES

2.1 Purpose of This Chapter

The ENIAC and the EDVAC are electronic digital machines. The purpose of this chapter is to indicate reasons for believing that machines of this type have important advantages. A digital machine has more flexibility and accuracy than is possible or practical in a continuous variable machine. A digital machine which is electronic can have a very decided speed advantage over one which is non-electronic. The question of reliability is deferred to Chapter 5; It will be sufficient now to say that it is believed that electronic digital machines can be made as reliable as others.

2.2 The Value of Speed, Flexibility and Accuracy

For the type of problem outlined in Chapter 1, it has already been emphasized that speed is an essential, not a secondary, consideration. The value of a result is strongly affected by its timeliness. It may not be easy to assess the value of timeliness in any specific case, but undoubtedly timeliness can be extremely important. Higher speed is therefore worth attaining even if its attainment increases costs. As Sec. 2.6 will indicate, the use of electronic elements to increase speed may actually reduce costs.

Flexibility and accuracy also are valuable from this point of view. With specialized machines, specific problems may be solved rapidly, but time may be lost when new problems require either a new machine or the modification of an old one. Further, even if time is no object, the cost of a group of specialized devices can easily total more than the cost of one flexible all-purpose machine.

2.3 Ultimate Design Aim

The ultimate aim of computing machine design may be taken to be the reduction of the overall cost of carrying out numerical solutions to mathematical problems. This point of view will be amplified in Chapter 4. No attempt is made to discuss the actual cost of equipment or operation, but some general comparisons which are relevant to such costs can be made. Actual machine operation is only one of several items contributing to the total cost. It is, however, the major item when a large problem is being done on a slow machine. It is logical, therefore, to give this item first consideration. Since analogy machines (continuous variable or non-digital machines) are capable of producing solutions rather rapidly in some cases, it is necessary first to point out why digital machines are advocated. If digital machines are accepted for reasons of flexibility and accuracy, then the most significant reduction in overall cost can be secured by the use of electronic components to increase speed without increasing equipment.

2.4 Accuracy

The accuracy obtainable with continuous variable machines depends upon the accuracy with which the component parts are constructed. However, exactly the same statement may be made about a digital machine. The essential difference is in the kind of dependence which holds in each case. In a digital machine, minor inaccuracies of components can be made to have no effect at all upon the accuracy of the computations. In continuous variable machines, there is a continuous (though not necessarily simple) relationship between the accuracy of the parts and the accuracy of the results (before translation into digital form). Increased computational accuracy can be attained only by increased precision of the parts. Not only are there practical limits to the precision with which the parts can be manufactured, but the maintenence of such precision during subsequent operation is extremely difficult. For instance, mechanical parts will lose precision through wear. On the other hand, digital machines require only that the accuracy of the parts be kept within certain tolerances, and the tolerance band can be made very wide. The accuracy of a digital machine is then limited only by the amount of equipment which can be used in it. This limitation is more often an economic one than a physical one.

2.5 Flexibility

Analogy computing devices vary greatly in flexibility. Scale models and fixed electrical networks which are completely specialized represent an extreme which is of no interest here. Large network analyzers and differential analyzers are usually considered typical examples of highly flexible analogy devices. Nevertheless, these machines are somewhat specialized and restricted in application.

A digital machine which can be directed to carry out any of the common arithmetic operations in any desired sequence on any given set of numbers has all the generality and flexibility required for any practical purpose. (It cannot compute the "exact" value of pi, but it can compute in a finite number of steps any desired approximation to pi.) Therefore, it can, for example, compute to any desired approximation the solutions of non-linear partial differential equations which are not obtainable from any existing analogy computer. This greater generality might be taken as sufficient justification for a digital machine.

Whether a digital machine is to be preferred for solving problems which can be solved on existing specialized machines depends on the speed (or cost) considerations which will now be discussed.

2.6 Digital Computing Speed

The preceding sections have indicated that advantages of accuracy and flexibility recommend digital rather than continuous variable computing devices. Even moderately large problems which are within the range of existing analogy machines require an enormous number of arithmetic operations when handled by digital machines. The cost of digital computing must be low if such calculations are to be undertaken.

Only very rough comparisons need to be made to demonstrate that digital computation by electronic devices should be a great deal faster than by the electro-mechanical devices used in existing large computers. Typical electro-mechanical devices are relays and electrically controlled counter wheels. Five or ten milliseconds is about the operating time for reliable high speed relays. Analogous operation of a vacuum tube can take place in 0.5 to 1 microsecond. The electronic device is therefore capable of operating about 10,000 times faster. (It may be observed that faster tube circuits can be developed at least as readily as faster relays can be designed.)

A vacuum tube and its associated circuit components costs approximately the same amount as a relay. This indicates that an electronic digital computer costing about the same as a non-electronic digital computer should certainly be a great deal faster, and might possibly be faster by a factor of 10,000. For various reasons, the actual factor for the ENIAC is something like 100, and for the EDVAC will, it is hoped, probably be more nearly 1000.

Moderately large computing problems are already being done by digital electromechanical devices. If digital computing costs can be divided by several powers of ten by appropriate use of electronic elements, it seems clear that electronic digital machines should find extensive application.

2.7 Comments on the ENIAC Design

Essentially, the preceding sections of this chapter have presented the point of view which led to the development of the ENIAC. It will be appreciated that the ENIAC design was also influenced in other ways which may be mentioned briefly. During the war, speed was the main object - for reasons of timeliness rather than reduction of cost. Not only was emphasis put on computing speed, but also on constructing the machine as soon as possible. For this reason, extended research and development were curtailed in favor of employing ready methods which could be put into production quickly. The emphasis on computing speed came about because the principle use contemplated for the ENIAC was the calculation of firing tables. In such work, only a small quantity of numerical information has to be introduced and withdrawn during a long computational process. Hence, no effort was made to achieve high input or output speeds. Also, for such work, the same schedule of operations might be followed for several days. Hence it was reasonable to have this operating sequence set up manually before the calculations were started, even though this might require several hours. It is therefore evident that the ENIAC is well suited for highly repetitive calculations, but that its high computing speed may be limited by the input and output devices when large quantities of numerical inofrmation are put through rather simple arithmetic processes.

The description of the ENIAC in Chapter 3 must be consulted to obtain a clear picture of the capabilities and limitations of the ENIAC. However, an idea of the computing speed can be given very briefly: additions may be done at the rate of 5000 per second, and 10-digit numbers may be multiplied together at the rate of 360 multiplications per second. Punch card devices introduce numbers at the approximate rate of 200 decimal digits per second. The rate at which numbers may be withdrawn (coming out on punched cards) is somewhat less. Since additions, multiplications, divisions and other processes can be carried on simultaneously, the contrast between computing speed and input and output speed is obvious.

2.8 Comments on the EDVAC Design

As the preceding section has briefly indicated, the ENIAC design has emphasized high computing speed to avoid what has heretofore been the most severe limitation on large computing problems, but has neglected other aspects of the overall computing job. If the time required for purely arithmetic operations is sufficiently reduced by electronic methods, other contributions to the overall time and cost of numerical mathematics may need reduction to preserve an efficient balance. In designing the EDVAC, all parts of the overall computing job are being considered. A discussion along these lines is given in Chapter 4.

CHAPTER 3

DESCRIPTION OF THE ENIAC

3.1 Introduction

In order to describe such digital computing machines as the ENIAC, it is desirable to set up a classification into which the different elements of the machine may be placed. The classification which we will employ is one which is particularly suited to large scale digital computing machines. It is probably not the best system for classifying the elements of continuous variable machines. However, even here, some interesting comparisons can be made using the following system of classification.

We divide the various units of the ENIAC into three classes - arithmetic elements, memory elements, and control elements. The arithmetic elements of the computing machine are those which form the basic operations of arithmetic - addition, subtraction, multiplication, division, and perhaps, square rooting. There should be added to this list another operation which we classify as being arithmetic. This is "magnitude discrimination" or simply "comparison." In this operation we expect to obtain a result which is dependent upon which of two numbers is the larger. This operation is essential in any automatic computing machine and is used to make decisions between alternative possibilities which exist in many problems.

The memory elements of the machine may be divided into two groups - the "internal memory" and the "external memory." The internal memory includes all memory devices within the machine and is thus finite, while the external memory exists outside of the machine in such a form as punched cards or perforated tape. This memory may be increased indefinitely but has the limitation that it must be associated with the machine by an input and output mechanism which is usually comparatively slow.

It is convenient to divide the internal memory into three classes - first, memory for numerical data which can be altered by the operation of the machine (usually rapidly); second, memory for numerical data such as empirical data which are known before the machine is started and may, therefore, be introduced into the memory device slowly, but must be withdrawn rapidly during the computing of the problem; and third, there must be a memory for instructions - those manifestations which cause transfer between the various memory and arithmetic units and cause the arithmetic units to do the various operations on the numbers. This form of memory, like the second class, may be set up slowly since the necessary information is available before the computation is started and likewise, it must produce its effect rapidly since it must control the operation at all points of the computation.

The discussion of the ENIAC will thus be divided into four parts - first, timing relations and program control; second, memory; third, arithmetic units; and fourth, input and output equipment. The general constructional data are included in the appendix.

The important arithmetic units of the ENIAC are: twenty accumulators, one multiplier, and one combination divider and square rooter.

The accumulators provide facilities for storing numbers computed in the course of a problem and further allowing the addition or subtraction of a second number to or from the stored number. They are capable of performing these operations with up to ten decimal digits and the associated plus or minus sign.

The multiplier computes the product of two decimal numbers of up to ten digits each.

The combination divider and square rooter computes the quotient of two nine digit decimal numbers or finds the square root of one nine digit decimal number.

Electrical connections are established between the units of the ENIAC by connecting them to trunk or transfer lines with plug or cable assemblies in order to provide the intercommunication of numbers. A number of input and output circuits from different units of the ENIAC may be connected to a single trunk provided the number of units is not greater than forty or fifty depending on the length of the trunk. The decimal digits are transmitted into, and received from, these trunks in the form of groups of pulses having a number of evenly spaced pulses, equal in number to the value of the digit represented. Thus, such a group may have from zero to nine pulses. In order to obtain high speed, the trunks have eleven wires in them to enable simultaneous transmission to the ten digits and the sign. It is not possible to transmit signals from more than one unit into a trunk at the same time. Many units may, however, receive signals simultaneously from the same trunk.

3.2 Timing Relations and Program Control

Since the ENIAC contains a number of trunk circuits, operations between various pairs of ENIAC units can be carried out simultaneously. This is possible not only because of this multiple trunk system, but because all units are synchronized by permanent electrical connections with the "cycling unit". Therefore if several operations are started simultaneously between various units of the ENIAC, and since all of these are timed from one and the same circuit, the various operations will end at known times relative to one another. Thus it is possible to plan the next group of simultaneous operations with the assurance that all of the prerequisite steps of the first group have been completed. If the timing of the various operations were not known, interlock circuits would be required to insure the completion of those various operations which are prerequisite to the next group.

The cycling unit supplies a number of specially shaped signals to the other units. It contains an oscillator or "clock" which generates impulses at the rate of one hundred thousand per second, each pulse having a duration of two microseconds. These pulses are fed into a twenty position electronic stepping switch or "counter", which enables the cycling unit to put out a special impulse or "program pulse" at every twentieth pulse of the "clock". These program pulses form the basis of the programming system and mark the beginning and end of the addition cycles which are the basic arithmetical intervals of the machine. The addition cycles are thus repeated at one twentieth of the clock rate, or at five thousand per second. An addition, therefore, takes 1/5000 of a second, or two hundred microseconds.

Between these uniformly recurrent program pulses, groups of pulses are supplied to the other units of the ENIAC so that when called upon to transmit or receive numbers, the required signals will be available to enable the carrying out of the operation.

When one unit is required to transmit numbers to another, the pulses are transmitted through a trunk to the other unit. These pulses originate in the cycling unit and are simply relayed as a whole or in part by that unit.

A switch on the cycling unit enables the operator to suspend continuous operation. The twenty steps of an addition cycle are then carried out every time a push button is pressed. This allows the machine to operate in a normal way during the addition, but permits the operator to advance the problem step by step allowing whatever time is required to check the results of the previous operation. Another position of the switch suspends this type of operation and allows a similar examination of the twenty steps that occur in an addition cycle. This ability to stop the machine at any or all stages of its operation is possible since all of the memory elements are able to retain their state as long as desired, assuming the electric power supply is not interrupted. Each memory element is connected to a small neon lamp which lights up when the element is in the "on" position. Since all the circuits except some of those carrying pulses are directly coupled, any lower clock frequency can be used.

The program pulses previously mentioned as the means of sequencing the machine serve to initiate circuits called "program controls" on the various units. These units, when initiated by a program pulse, cause the unit on which they are located to operate in accordance with the settings of several switches which are part of the program control. When the operation is completed, the program control may transmit another program pulse through special program trunk circuits of one wire each. The program controls are connected to the trunks by a plug and cable assembly, in some respects similar to those previously mentioned for the digits. A program control capable of transmitting a program pulse is known as a "transceiver". A transceiver program control may repeat its operation as many as nine times in accordance with the setting of an associated switch before terminating its operation and transmitting its output program pulse. Some of the units are provided with program controls called "receivers" which operate only once upon reception of a program pulse and do not transmit a pulse when the operation is terminated. Transceivers, with their switches set to positions which do not operate the associated units, can be used to provide delays and isolating of "buffing" action.

It is now seen that one can set up sequences or chains of operations simply by connecting the output of one program control to the input of another program control, making connections through the program trunk lines using plug and cable assemblies. A special control, on a panel known as the "initiating unit", gives out a pulse upon operation of a manual push button, and this pulse can be introduced into such chains to initiate their operation.

In brief, the program control units known as transceivers and receivers remember, with their switches, what processes are to be done. They also time the operations, allow operation from a single terminal input, and permit signals from the cycling unit to cause the required operation to take place in the unit to be controlled.

If it were sufficient in most problems simply to go through a sequence of operations, the above chain system would suffice. However, two difficulties arise in practice. The number of steps in most interesting problems is so large that the group of about three hundred program controls in the ENIAC would be entirely inadequate. Secondly, it is sometimes desirable to make a choice between two or more sets or chains of operations, this choice depending upon some numbers which are the result of an earlier computation. If this choice were to occur only a few times in the course of a computation, one could allow the sequences to run out, thus stopping the computation. At this point the operator could make the choice manually. If this situation occurs a large number of times in a commutation manual choice would be impractical. Both of these difficulties may be overcome by the use of a unit in the ENIAC called the "master programmer".

The master programmer consists of ten units, each of which may be employed in various ways to count program pulses and to switch program connections. Each unit contains a six position electronic stepping switch and an associated counter. For each unit there is one input channel and any program pulse entering this unit is registered in the counter and also causes the transmission of a program pulse from one of six possible output terminals. The position of the stepper determines which output terminal is so activated. A group of manual switches is associated with each counter-stepper unit, and when the number of pulses received by the unit has advanced the counter to a number corresponding to the switch setting, the stepper is moved to its next position and the counter is cleared to zero. Separate switches are provided for each stepper position. The intervals between transfer from one output channel to the next can therefore be determined in terms of the number of pulses received, and a different number may be used for each stepper position. If less than six outputs are desired, a special switch can be set to cause the stepper to return to its first position after reaching any chosen position.

An input terminal allows the counters to be set directly from a pulse group. Another input terminal allows the stepper to be stepped directly from a pulse group. Finally, an input is provided to allow direct re-setting of the stepper. Thus, there are altogether nine terminals - one program input, six program outputs, one direct stepper input, and one stepper re-setting input terminal.

To return to the two difficulties previously mentioned which the master programmer is to overcome, it is clear that since most problems may be separated into a large number of repeated routines, each with a comparatively few steps, the master programmer will allow the ENIAC to cope with these problems in spite of having only three hundred program controls.

The master programmer temporarily forms program chains into "rings". This would be impossible without the master programmer, since the number of program cycles around these rings would be uncontrollable if such a ring were established manually.

It is very important to understand that this far from exhausts the possible improvements which the master programmer makes in the ENIAC. It is possible to have the main routine divided into sub-routines, in which case one stepper is used to feed another stepper, thus allowing the proper sub-routine to be chosen in the course of a regular routine. This process can be carried even further, and thus an elaborate hierarchy of program sequences can be established. The saving in program controls accomplished by such arrangements is enormous.

The second difficulty, that of having to make a number of numerically determined choices as to what routine to do next, can also be overcome by using the master programmer. It is simply necessary to use an accumulator as a magnitude comparing device and to have it signal a stepper through its direct input to change to a new routine.

In the ENIAC, the master programmer does not serve as the sole governing unit, but coordinates many small decentralized control units.

3.3 Memory

The ENIAC has its memory divided into four fundamental classes. It is our present purpose to discuss three of these classes and to postpone the discussion of the fourth until later.

The first class is the one in which the information can be introduced and withdrawn at extremely high speeds, preferably in one addition time

(1/5000 second). As was previously noted, this is the type of memory provided by the accumulators. This type of memory is necessary to hold the information computed in the course of a problem while some further calculation is made upon it, or until the time when some operation is to be done which requires it.

The second class of memory consists of three function tables or constant storing units. This class of memory differs from the type of storage employed in the accumulator, mainly in that the tabular values are introduced manually which requires considerable time. These values, however, may be withdrawn at the relatively high speed of five addition times (1/1000 second). The function table provides a hundred tabular values of twelve digits each - each tabular value having, however, two sign indications. The digits, as well as the sign indications, are all set by separate manually operated rotary switches. Thus, there are fourteen switches for each tabular value. The two sign indications are provided so that the twelve digits of a tabular value may be assigned to two separate numbers. By dividing the twelve digits into two equal groups, two functions of one hundred tabular values, or one function with two hundred tabular values, can be stored in a single table. The hundred tabular values of the function table are designated by the successive decimal integers from zero to ninety-nine. Thus, the first two digits of an argument may be used to choose the correct tabular value, while the remaining digits are used in an interpolating routine which makes use of the regular arithmetic functions of the machine to obtain the value of the function corresponding to the argument. Thus, the function table supplies only the nearby tabular values that are to be used in an interpolation and does not itself do any interpolation. Any desired interpolation formula is set up as a sequence of operations, using the accumulators and multipliers in the same way as is done with any other arithmetic problem. In order to obtain the nearby values which are required in the interpolation, the program controls of the function table also produce arguments which are one or two integers above or below the argument introduced.

Four tabular values, in addition to the one hundred mentioned above, have been added - two at each end of the table so that those extra arguments required above will be present in the table when the argument value is either zero or ninety-nine. These tabular values are designated by the decimal integers -2, -1, 100, and 101 respectively. Thus in reality, the table contains 104 entries. Since each entry has fourteen switches, there are 1456 switches to be set. If the switch settings are read to the person setting the switches, they can all be set in approximately half an hour. This comparatively long set-up time is partially mitigated by having three such function tables in the machine and by providing the switch assemblies with wheels and easily disconnected cable connections so that several banks of switches can be left set up and plugged in when required. In fact, where a function is required frequently, it is practical to replace the switch assemblies by permanently wired assemblies which can be plugged in when needed.

Several additional uses of the function tables are possible. They may be used for the storage of any group of constants which must be introduced into the machine at a high speed. In problems where the ordinary program controls of the ENIAC are insufficient, it is possible to employ the table as a 104 position program selecting device using its outputs to initiate program circuits rather than feeding the outputs into digit channels. A function table may be used, therefore, to sequence operations, or chains of operations, which may be initiated in sequence or chosen at random depending on the value of the two pulse groups introduced as argument into the table. These pulse groups may be obtained either from the program circuits or from the digit circuits of the machine.

To sum up, three function tables are provided, of a hundred and four entries each, with facilities for dividing the entries into two numbers where twelve digit accuracy is not required for one function. The most notable feature of this type of memory is its large capacity and high speed. Unfortunately, this is coupled with the necessity for manual, and therefore slow, introduction of the data.

The third class of memory is the program equipment which was previously described. The program circuits remember what processes are in progress, what these processes are how long these processes take, and what processes are to be done next. The actual physical memory is made up of the many tube circuits switches trunk circuits, and plug and cord assemblies which make up the programming system in the ENIAC.

The fourth class of memory will be covered later when we discuss the introduction and withdrawal of data from the ENIAC.

3.4 Arithmetic Units

The accumulators are the basic arithmetic units of the ENIAC They are able to add a number to a number already stored in them. This ability to add is inherent in the ten stage ring counters which serve as the memory elements. In order that numbers may be simultaneously added into all ten counters which provide the memory of an accumulator a special circuit inserts pulses corresponding to any carry-overs which may take place after the ordinary transfer process Since subtraction is carried out by a system of complements with respect to 10^{10}, an extra pulse is required to simplify the mechanism for transmitting numbers. Seven pulse times are required for the carry-over process. This additional time must be added to the ten pulse times required to transfer the number, and the three additional pulse times required to allow the program equipment to operate prior to the transfer. A total of twenty pulses is obtained which corresponds to the twenty stages of the cycling unit. Thus, an accumulator requires twenty pulses spaced at ten microseconds, or 1/5000 of a second to do an addition. The complement system of subtraction was used to avoid the necessity for having counter rings which could advance in either direction.

The accumulators are provided with two output circuits - one for positive numbers and one for their complements. It is thus possible to transmit simultaneously a number from an accumulator into one channel and its complement into another. If both output circuits are connected to the same channel, it is of course impossible to do more than one of these operations at a time. Five input circuits are provided which allow the accumulator to receive from five different channels. This is a necessity if several simultaneous operations are to be carried out between different pairs of units. These input channels are also useful in that they provide a simple method of multiplying numbers by powers of ten.

The accumulator contains eight transceiver program controls and four receiver program controls. Since the transceiver program controls allow an operation to be repeated up to nine times a simple method of multiplication by small constants is possible. A program control can, if desired, by the setting of one of its switches cause an accumulator to clear its counters to zero following a transmission. A round-off switch is provided on each accumulator which enables it to clear and leave a five, instead of a zero, in any one of the decades. This five, in conjunction with a special plug device to delete the undesired digits, allows retention of as few significant figures as may be required. If more than ten significant figures are required, it is possible to connect two accumulators in tandem by means of special plug and cable assemblies and to obtain as many as twenty significant figures. The accumulators indicate the numbers which they contain on a bank of neon lamps, which provide a very convenient facility in checking the operation of the ENIAC.

As previously stated, signals may be taken from the sign indicating circuits of the accumulator to provide magnitude discrimination which permits computed numbers to control the program sequence of the machine.

In addition to the two outputs previously mentioned, the accumulators have a set of "static" outputs which provide a ten wire circuit from each counter ring, one from each stage. Thus, since there are ten counter rings, a hundred output circuits are available. These circuits provide all the information about a number simultaneously and are essential in the multiplier.

The multiplier is fed by static outputs from the two accumulators which receive and hold the multiplier and multiplicand respectively. The multiplier contains an internal multiplication table which allows the multiplication of one digit of the multiplier by ten digits of the multiplicand simultaneously. Since pairs of single decimal digits have, in general. two decimal digits in their product, it is necessary to provide two accumulators to receive the twenty digits which result from the multiplication of one digit simultaneously by ten digits. This is true because it is desirable in the interest of speed, to add all of these "partial products" simultaneously to the accumulated partial products of the preceding steps. These two accumulators can conveniently accumulate the "units set" and the "tens set" of these partial products without any interference in the normal carry-over. Since each set of the successive partial products requires ten channels to receive them, and since every set corresponding to the different digits of the multiplier times all of the multiplicand digits must be shifted by one channel with respect to the preceding one, the product accumulators require twenty input channels. These twenty digit accumulators are made by coupling two ordinary accumulators in the manner previously described. Therefore, two sets of double accumulators. or four ordinary accumulators, are required to collect the partial products from the multiplier. After this collection has been accomplished, the final product is obtained by adding the units set and the tens set with a single shift between the channels to obtain the proper relation between the two sets.

If one of the numbers is a complement, the product obtained will be in error by a number whose magnitude is 10^{10} times the other number. If both numbers are complements, the product obtained will be in error by a number whose magnitude is 10^{10} times the sum of the two numbers. Following the collection of partial products, but before the final product is obtained, a special sign-indicating circuit causes appropriate corrections to be made, transmitting them from the multiplier and multiplicand accumulators into the product accumulators.

The multiplier is equipped with twenty-four transceiver program control units, any one of which can control the above process. These program controls each contain a switch which allows the multiplication to be carried out with any number of digits in the multiplier between the maximum of ten and a minimum of two. In addition, other switches in each program control are available which allow the multiplier to cause the reception of numbers into the multiplier and multiplicand accumulators and which allow transmission and rounding-off by the product accumulators.

The time required for multiplication is computed as follows: (a) one addition time to operate the multiplication table, (b) one addition time for each digit of the multiplier, (c) one addition time for complement correction terms, (d) one addition time to add the units and tens parts of the partial products, and finally, (e) one addition time to transmit the product and receive the next multiplier and multiplicand. Therefore, the maximum time of a multiplication will be obtained with a ten digit multiplier and is fourteen addition times or 1/360 of a second.

The third, and last, arithmetic unit of the ENIAC is a combination divider and square-rooter. This unit can either divide or take a square root, but cannot do both at the same time. This compromise seems desirable since square rooting is fairly infrequent, and is so similar to division, that very little additional equipment was required to make a combination unit. The method employed to do these processes was chosen so as to require a minimum of equipment since neither of these processes are as frequent as the other arithmetic processes, and, therefore, would not justify a large expenditure of equipment.

Division is carried out by the conventional method of successive subtractions.

The only uncommon feature is that the shifting operation is accomplished by transmitting the dividend back and forth between the dividend-accumulator and a special shifting accumulator. This, unfortunately, limits single divisions to nine decimal places. However, since the remainder can be retained, the process can be carried further by a special program arrangement if greater accuracy is desired. Another feature of this divider is that, after successive subtraction of the divider from the dividend has produced an overdraft, the remainder is shifted one place to the left and the divisor is added instead of subtracted in obtaining the next digit of the quotient. This turns out to be somewhat simpler to mechanize than the ordinary method of restoring the overdraft.

Square rooting is accomplished essentially by the method of subtracting successive odd numbers and is carried out by a method otherwise analogous to division. As in division, the overdraft is not restored before shifting. The combination divider-square rooter makes use of the ability of the accumulators to provide magnitude control and is in itself, therefore, merely a permanent assemblage of program equipment and circuits for supplying the various constants required to build up the answers. This unit requires several addition times to set up and round off its results. The main computational time is that required for the successive additions and subtractions. An addition time is required between each trial addition or subtraction to sense the sign of the remainder. Since five and a half trials are required on the average, one can estimate the total number of addition times required by lumping the computational time with the set-up and round-off time in a simple formula which gives the total addition times as approximately 13 multiplied by one more than the number of digits required in the answer. Thus, for nine digit answers, the average time required for division or square root is 130 addition times (approximately 1/38 second). Since this time is variable, depending upon the numbers employed, and may be as high as 210 addition times (1/23 second), considerable saving in time may result in carrying out parallel operations.

In order to make possible such parallel operations, an interlock circuit is arranged so that the output pulse, which signifies the end of dividing or square rooting, is not transmitted until both the parallel operation and the division or square root have been completed. This unit is equipped with eight transceiver program controls, each having its own interlock input as well as its own regular input and output.

3.5 Input and Output Equipment

Since the actual computing process is a highly automatic and rapid one, there must be some system for introducing or withdrawing numerical data which is considerably faster than setting switches or copying down numbers from a bank of neon bulbs, which are the only means so far described for doing this.

In order to save the time required to develop any special apparatus, automatic business machines which operate with Hollerith punched cards are employed. Comparatively simple modifications of standard machines of this type give higher speed for introducing or withdrawing data, than appear possible with any other type of commercial equipment. This speed results from the simultaneous use of eighty channels.

Paper tapes are also used in this field by others, and have the advantage of being more easily handled and thus less likely to lose their proper sequence. They have an additional advantage in that they can easily be run forward and backward, and can introduce and withdraw data onto different parts of the same tape simultaneously. Tapes have an inherent disadvantage when used for sorting, since a considerable waste of tape is involved.

Paper cards were chosen because of the economy with which they could be sorted, because no fast tape machines were available, and further, because few of the desirable auxiliary machines that are so necessary to scientific computing were available for handling paper tapes.

On the other hand, card machines were available which would punch, verify, and reproduce the cards, which would print the information upon the cards themselves or upon other sheets of paper, and which would sort and collate the cards. Standard card machines were supplied by the International Business Machine Company, who also kindly coöperated by supplying a special machine for taking information from their standard cards and putting it in a form which could be used in the ENIAC. Also, IBM supplied a standard machine suitably modified to permit punching standard cards from information taken from the ENIAC. Various plugs and cables used for attaching the portable function table switch panels to the function table control panels were also supplied by IBM.

It is, of course, evident that some intermediate memory, such as the cards provide, is necessary if the transition from slow manual operations tc fast mechanical operations is to be made. In addition, however, these cards play a very important role in providing the machine with an auxiliary memory of infinite capacity. It is this memory which was referred to earlier as the fourth class. While the absolute speed of introducing and withdrawing data from this type of memory is quite fast (approximately 1/12 second for a ten digit number), it must be considered as a slow memory compared to the electronic equipment in respect to both the putting in and the taking out of data. Since the card machines, both in reading data from the cards and in punching data onto the cards, scan across one dimension of the card, some auxiliary memory is required to hold these numbers.

Rather than use some of the expensive and limited electronic memory which the accumulators provide for this rather slow operation, a number of telephone relays, of a very reliable type, were employed for this purpose. These relays and the equipment for testing them were supplied by the Western Electric Company. The relays were designed by the Bell Laboratories.

Relays are employed not only because they provide an inexpensive memory of adequate speed, but also because they constitute the simplest method of transition from the electro-mechanical card machines to the electronic circuits. This is especially true in the printer where, because a number of contacts may be put on a single relay, it is possible to have a simple mechanism for converting negative numbers expressed as compliments (in the electronic circuits) to their true negative form (on punched cards). This latter form is preferred on the cards ooth to facilitate examination and printing of the numbers and because it fits in with the system which is used by the standard auxiliary card machines.

The machine which reads the cards is able to store eighty decimal digits in a group of relays. By means of special holes punched in the cards, it is possible to designate them as "master" and "detail" cards. Some of the eighty digits may be taken from one of these types, and the rest of the eighty digits from the other types. The master-detail card arrangement is very useful when it is desired to do problems in which several constants are required for a considerably larger number of cycles than those which are introduced on the regular or detail cards. These constants are put on special master cards on which a special punched hole causes information from these cards to be retained until another master card appears. The master cards are then used for the less frequently changed numbers or constants while the detail cards are used for those which change frequently. The numbers are stored in the relays, which are associated with the input machine. These in turn control tube circuits associated with thirty transceiver program controls and allow any group of five or ten of the eighty digits to be transmitted at any one time into the rest of the machine. Such a transmission requires only one addition time (1/5000 second). Twenty additional digits and four sign indications, which can be set by hand switches, are provided for constants which do not require alteration in the course of a problem.

This input unit, called the "constant transmitter", can receive cards at the rate of 120 per minute. The output or card punching unit, which we call the "printer", operates from a group of relays which are controlled by the static outputs of the counters in the accumulators and master programmer. Semi-permanent cable connections allow the printer to have its eighty columns controlled from any eighty of the two-hundred and twenty counters which the machine has, with the exception of those which already have their outputs employed in the multiplier and multiplicand accumulators. The printer is capable of punching cards at the rate of one hundred per minute.

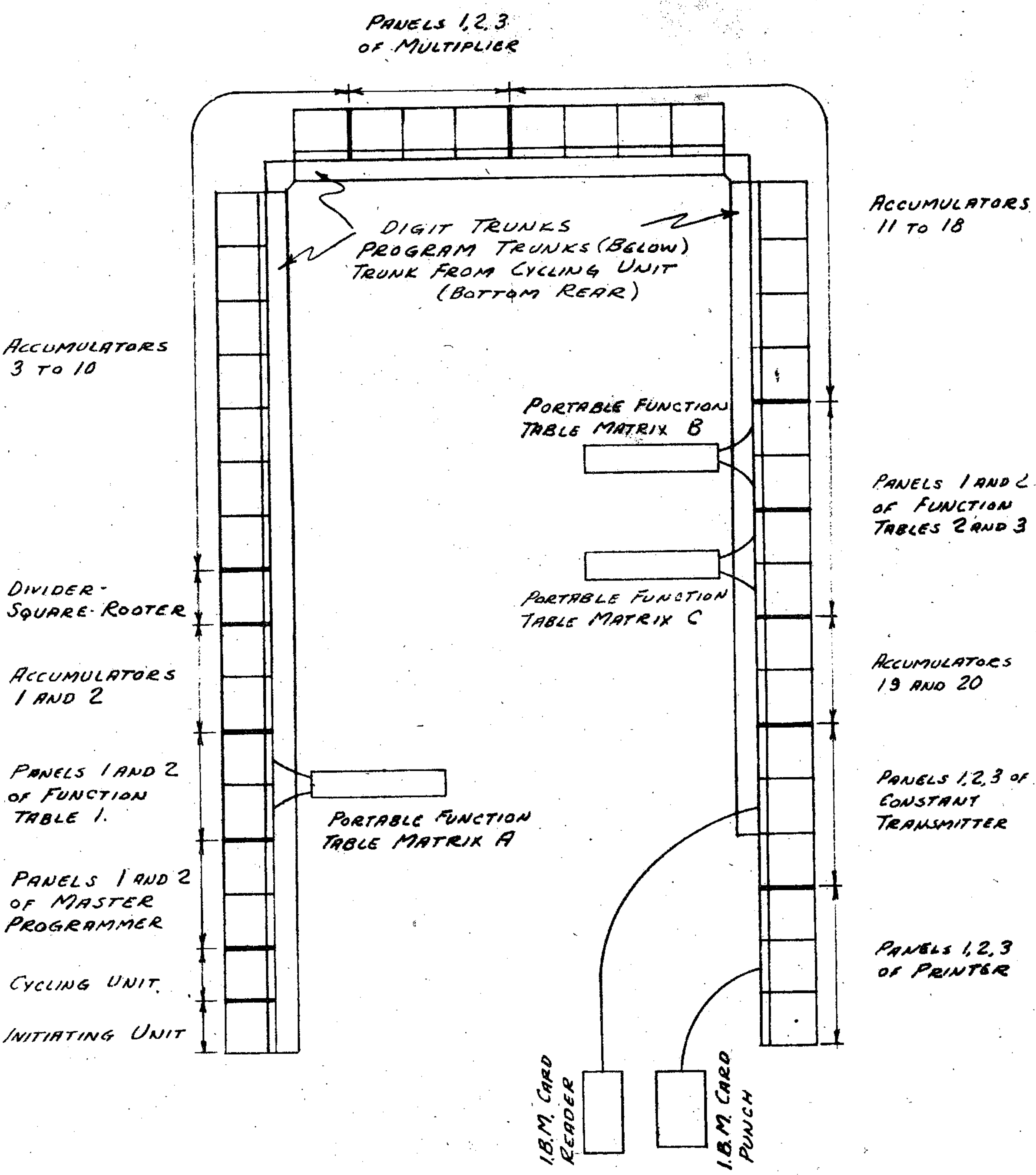


Fig. 1 - LAYOUT OF BASIC UNITS OF ENIAC

ENIAC TIMING CHART

OPERATION	TIMINGS		
	Seconds	Micro-seconds	Addition Times
ADDITION, SUBTRACTION, OR TRANSFER Including transfer from constant transmitter	$\frac{1}{5000}$	200	1
Transfer repeated "n" times in the accumulator (n = 1 to 9)	$\left(\frac{1}{5000}\right)$ n	200 n	n
MULTIPLICATION By "n" digit multiplier (n = 2 to 10)	$\left(\frac{1}{1250}\right)+\left(\frac{1}{5000}\right)n$	800 + 200 n	4 + n
By ten digit multiplier	$\frac{1}{360}$	2,800	14
DIVISION OR SQUARE ROOTING Average time for nine digit result	$\frac{1}{38}$	26,000	130
Average time for "n" digit result (n = 3, 6, 7, 8, or 9)	$13(n+1)\left(\frac{1}{5000}\right)$	13(n+1)(200)	13(n+1)
Maximum time for nine digit result	$\frac{1}{23}$	42,000	210
Maximum time for "n" digit result (n = 3, 6, 7, 8, or 9)	$21(n+1)\left(\frac{1}{5000}\right)$	21(n+1) (200)	21(n+1)
OBTAINING A FUNCTIONAL VALUE ONCE	$\frac{1}{1000}$	1000	5
Repeated " n " times (n = 1 to 9)	$\frac{1}{1250}+\frac{1}{5000}$ m	800 + 200 m	4 + m

CHAPTER 4

DESIGN PRINCIPLES FOR HIGH SPEED COMPUTING MACHINES

4.1 The ENIAC and the EDVAC

As was said in Sec. 2.7, the principal aim in designing the ENIAC was to achieve high speed in the computation of firing tables. There is no essential or fundamental restriction imposed by the ENIAC design on the character or complication of the problems which it can do. Practical limitations on internal memory capacity, both for numerical data and for programming sequences, serve to limit the efficiency and speed of the ENIAC when used for problems beyond its intended scope. Nevertheless, the ENIAC should be able to solve many of these larger problems faster than they can be done on any other existing machine. Undoubtedly, the ENIAC will be used in this way until the EDVAC or some other machine with better facilities for handling non-linear partial differential equations can be built.

In the design of the EDVAC it is hoped to achieve a great deal of flexibility in the handling of both large and small problems, and to do this at a reasonable cost -- that is, in a reasonable time with reasonable amounts of equipment. Reasons for believing the reliability will be good are given in Chapter 5. Since the EDVAC will be a digital machine, any desired accuracy can be attained.

Some principles which have guided the design of the EDVAC will now be discussed.

4.2 The Overall Computing Problem

A general aim of computing machine design should be to increase the speed and lessen the cost of scientific computations. This implies much more than high-speed arithmetic. For purposes of discussion, it will be useful to sub-divide the overall computing job into these four steps:

(1) Preparation of the problem in a symbolism and form appropriate to the machine.

(2) Actual physical preparation of the machine to carry out the work of obtaining the solution.

(3) Machine operations such as computing or sorting to achieve the solution.

(4) Preparation of the solution in a form suitable for examination and use.

There is no particular advantage in reducing the time and cost of any one of these steps if the overall time and cost are largely controlled by the other steps. It is also important to recognize that all four steps are affected by the design of the computing machine and therefore the designer of a machine should carefully consider the way in which each step is influenced by his design.

4.3 Automatic Operation Desired

It is reasonable to try to make all operations as automatic as possible. Manual operations cannot compete with machine operations in speed, and even unskilled or routine clerical labor is expensive when compared to the cost of automatic machine processes.

Since step 3, the machine computation, can be done automatically and at high speed by electronic digital machines, it becomes important to reduce the time and cost of steps 2 and 4 by rendering them automatic. This is possible, although it is not immediately evident what this will cost in terms of equipment. The point which it is desired to emphasize in this section is that one can afford to spend a great deal of equipment in this way.

In making steps 2, 3, and 4 all automatic, however, a greater burden is thrown on step 1, because the operator must plan all subsequent steps beforehand and be able to specify, in some sort of symbols, all the instructions needed to carry them out. For almost all problems of any complexity at all, step 1 will be the most time-consuming and costly. When this fact is recognized, it is seen that an important object of computing machine design is to simplify step 1. Further discussion of this objective will be given in Section 4.14.

4.4 Speed Versus Equipment

Computing speed and amount of computing equipment are to some extent interdependent. The designer is faced with many choices which involve this relationship. If a designer is restricted to the use of but one type of element or component, the computing speed which he may achieve is more or less fixed by the amount of equipment he can use. By using more equipment, and carrying out many operations at once, he can effectively increase the computing speed. Conversely, he may prefer to sacrifice speed in order to reduce equipment. This exchange has limitations, of course, since all operations of a long calculation cannot be done concurrently, nor can equipment be reduced to the vanishing point by going to the other extreme.

It is obvious that excessive equipment is undesirable from the point of view of both first cost and maintenance. It would therefore be unwise to multiply equipment unless the speed so gained in the automatic processes serves to materially reduce the overall computing time.

4.5 Levels of Serial or Multiple Operation

At this point it is convenient to distinguish various levels at which the designer may choose to gain speed by use of more equipment or may choose to sacrifice speed to save equipment. The terms "multiple" and "serial" will be applied to these choices.

	LEVEL	SERIAL	MULTIPLE
(a)	Representation of numbers	Digits of the same number follow each other through same channels and devices.	Separate channels and devices operate on all digits of a number at once.
(b)	Standard arithmetic processes, such as multiplication	Digits are passed in sequence through the same adder. Multiplication is done by repeated addition.	A large number of adders act on all digits at once. (Partial products may be used - if so, they are all summed by multiple adders.)
(c)	Combination of standard processes by programming.	Never more than one standard operation is done at one time. All steps are arranged in a single sequence.	Several standard arithmetic processes may be carried on at the same time. This is subject to the operator's control.

It is assumed, in level (c), that the various standard operations which may be put in either serial or multiple order are all part of the same computing problem. The process of doing many problems on as many different machines operating simultaneously might be thought of as a fourth category of multiple operation, but this is of no interest to the designer.

4.6 Multiple Operation in Non-electronic Machines

For non-electronic computing machines, the computing time for complicated problems may be quite large compared to the time spent in step 1 or other non-automatic steps. Multiple operation to gain speed therefore seems desirable. In fact, multiple operation at level (a) is quite customary, and various degrees of multiple operation at level (b) are often used. Any attempt to gain speed by multiple operation at level (c) has a disadvantage not shared by the other two levels. Here the operator must plan his problems carefully so as to make use of the speed which is afforded him. To some extent, then, the gain in computing speed is offset by the additional burden which is transferred to the non-automatic step 1.

4.7 Serial Operation in Electronic Machines

The situation is quite different when electronic machines are considered. In this case computing speeds are so high that often many numerical solutions can be obtained in less time than may be required for carrying out step 1. It is therefore reasonable to examine the possibility of sacrificing some of this speed in order to reduce equipment. The sacrifice in speed need not be as drastic as one might suppose, however, for the following reason. There is no sharply defined limiting speed for electronic equipment, and the design of electronic computing circuits has not yet reached "speed saturation". As the total amount of electronic equipment is reduced, intensive circuit design becomes easier and more practicable, so that each element may be able to operate at a higher rate. (Thus, the fundamental pulse rate for the EDVAC is expected to be ten times that for the ENIAC.)

It does not appear unreasonable to insist on serial operation at all levels. Specifically, this means that

(a) Numbers are to be represented by timed pulse trains, all of the digits for any one number following each other in time sequence through the same channels and circuits.

(b) All of the simple elements of a standard arithmetic process are performed in serial order. Thus, successive digits are added by the same adding circuit, and multiplications are carried out by performing a number of additions in sequence.

(c) There is never more than one standard arithmetic operation being carried out at any one time. All arithmetic steps follow each other serially.

4.8 Serial Operation in Analogy Machines

It is of interest to note that analogy machines also employ serial operation for certain processes. Network analyzers, once they are set up, produce the desired solutions serially with only a few measuring instruments. The alternative, that of using many instruments distributed over the network and reading all of them at once, would be a multiple process which can hardly be justified.

Also, differential analyzers scan continuously by variation of the independent variable (usually referred to as "time"). In this way, a relatively small number of integrators are able to produce a continuum of results.

Nevertheless, these machines do not employ scanning (or serial operation) to the fullest possible extent. In a differential analyzer, each integrator must, during a given problem, be continuously associated with just one integral process. As more complicated problems are devised, more integrators must be used. A similar remark holds for the network analyzer. Each branch of the network has certain impedances associated with it, and as problems become more complicated, more impedance elements must be used. To solve a non-linear problem on a network analyzer numerous non-linear impedances are required. If, by some scanning process, a single non-linear element could be "switched" from place to place in the network, scanning all the positions where such an element is required, the network analyzer would in effect be able to reduce equipment by serial operation.

4.9 Memory Scanning by a Computer

Although the notion of a serially operated network analyzer may not appear practical, such an idea is closely connected with a practical and quite flexible scheme for a computing machine. Let this machine have a single arithmetic device and a great many number registers, or memory elements, so that it may receive numbers from the memory and also store numbers which it computes in the memory. The arithmetic unit is the single "non-linear element" which scans the various points in the "memory network." Because it is completely flexible, however, this unit must be told how to scan and what sort of computations to do as it proceeds.

4.10 Instruction Memory

For high-speed computation, not only must the arithmetic organ compute at high speed, but the memory elements must be able to supply and receive numbers at high speed. Obviously, a further requirement is that the instructions to the arithmetic unit must keep pace with it. The faster the instructions can be carried out, the faster must new instructions be supplied.

This requirement as to instruction speed can be met in a direct and simple fashion by storing instructions (in numerical code, so to speak) in memory elements or registers of the same type as those used to store the numerical data. The function of receiving such instructions from the memory, interpreting them, and causing the instructions to be carried out, can then be vested in a "control unit" which is also a part of the computing machine.

4.11 Function Table Memory

If high-speed electronic memory elements are used for numerical data produced in the course of operation, and are also used for storing operating instructions, it appears that empirical or mathematical functions which are necessary to a computation should be accommodated in the same way. By so doing, all memory elements are made identical, and a very desirable unification is achieved.

4.12 Flexibility of Memory

By unifying all memory functions through the use of similar memory elements for both operating instructions and numerical data (including function tables), it is possible to obtain new flexibility. All memory elements can be interchangeable. Consequently there is no need to specialize any memory element. Any portion of the total memory may be allotted to operating instructions, and any portion to function tables. This allotment is in no way fixed, so that any part of the memory may be used for any purpose. The same memory element may be used in various ways during the same problem, in fact.

4.13 Memory Capacity Needed

No matter how much high-speed memory capacity is provided, interesting problems are bound to arise which could profitably use even more. Such problems must then be handled by making use of an infinite exterior memory capacity of which IBM punched cards are one example. There would be no need at all for an internal memory if such an infinite external memory could operate at electronic speeds. Since this does not seem possible, some compromise must be made. Some form of permanent and inexpensive infinite memory must be provided. Any improvement in the rate at which numbers and instructions can be transferred from this memory into the internal memory, or vice versa, is greatly to be desired. Obviously, high rates can be achieved by use of multiple equipment, but this does not lower the cost. Any simple and practical form of external memory so far considered cannot compare in speed to the electronic internal memory speed. The internal memory capacity must therefore be chosen with reference to the kind of problems which are to be done at high speed (that is, at speeds not limited by the input and output rates). For instance, an internal memory capacity equivalent to about 2000 ten-digit decimal numbers may be considered a minimum for handling numerical solutions of partial differential equations having two independent variables.

4.14 Simplification of Step 1

It is believed that serial operation at level (c) is of extreme value in simplifying the planning and layout of a problem which comes under step The operator has no parallel timing problems to worry about. Since the machine can do but one operation at a time, his attention can likewise be put on only one thing at a time.

The machine must be capable of carrying out any numerical operation required by the mathematician. It is evident that only a few basic or standard operations could be built up from these Certain operations, although expressible in terms of more elementary ones are used so frequently that it is advisable to have the machine "know how" to do these processes. An example of this is multiplication, which should be handled as a basic operation for which no elaborate instructions need be given. On the other hand, the temptation to provide a great variety of standard operations must be resisted. More of these requires more control equipment, but the main reason for limiting the number of basic operations is that step 1 is not simplified but rendered more difficult by having too many operations to remember and to choose among. An instruction code must be simple enough and compact enough to be easily memorized and used without need for constant reference to a dictionary. It is believed that adequate flexibility and easy coding is provided by restricting the arithmetic operations to addition, subtraction, multiplication, division and possibly square root

A general machine needs also the ability to alter its program of instructions in accordance with the outcome of some numerical calculation. In punched card computing machines this function can be performed by "digit control." One form which this can take is that of an instuction which causes the machine to choose between two courses of action when two numbers, a and b are specified. One program is followed if a-b is positive or zero, and the other if a-b is negative Such an order can obviously be applied to programming identity checking (see Section 5.3), but it has many other applications.

Digit control is easily carried out in the ENIAC, and the instruction code for the EDVAC is to contain this essential operation.

Another most important requirement on the programming or instruction possibilities of any large and general machine is that it must be possible to set up hierarchies of control. This may be expressed by saying that it must be possible to set up a given program routine just once, and call upon it as often as required in the course of some other program routine. If this facility is not provided, the same instructions mught have to be set up many times. This would be wasteful of memory equipment and wasteful of the operator's time. It is impossible to overemphasize the extreme limitation which lack of this facility would impose. In the ENIAC, the master programmer serves the desired purpose. Instruction orders which operate upon other instruction orders will make this procedure logically possible, in the EDVAC.

One further desirable characteristic for programming systems should be pointed out. Any sequence of instructions which define a problem should be capable of rapid and easy modification by the operator when he desires to make slight variations in his problem. This is obviously possible in the ENIAC, since the operator has immediate access to all of the manual program controls. In a machine which is automatically set up from tape, for instance, it is not obvious that this condition is fulfilled. In the design of the EDVAC, it is hoped to meet this condition.

In conclusion, it may be remarked that for any large machine it is desirable to establish a "library" of standard program routines. In the case of the ENIAC, such a compilation can sometimes avoid duplication of effort in the planning stage of a new problem which has parts resembling old problems, but it can hardly do much to reduce manual set-up time. For machines which are automatically set up, a great deal of time and effort can be saved by such a library. Electro-mechanical machines now make extensive use of such libraries, which include function tables as well as program routines. Exactly the same techniques can obviously be applied to the EDVAC.

CHAPTER 5

RELIABILITY AND CHECKING

5.1 Relationship of Reliability and Checking

The need for checking depends greatly on the inherent reliability of the device being checked. A device which fails extremely often is not worth checking, and one which never fails needs no checking. In practice, failures can never be eliminated completely, and some method of controlling and recognizing failures is highly desirable.

In a digital computing machine, the failure of almost any component can vitiate the entire computation. Even a momentary failure of any part may therefore be considered as a failure of the whole machine. Since a large computing machine contains many parts, its failure frequency may be high even though the individual components have a low failure rate.

5.2 Detection and Localization of Failures

Checking devices or methods detect rather than prevent failures. Since the checking device or process may also fail, it might seem that there is no end to checking. It must be recognized, however, that no system can really detect all errors; the true aim of checking is to reduce the probability of undetected failures to an extremely low value which is deemed tolerable.

A secondary, but by no means trivial, function which may be served by checking is to localize the failure. The importance of this feature increases with the size of the machine and the complexity of the calculations. Suppose a long and involved problem is done by two different methods on two different machines. If the results agree, there is very little chance that they are wrong; but if they fail to agree, there may be no clue as to which is right or where the error was made. To avoid having this situation recur frequently, either checking must be done more frequently on smaller parts of the problem, or the machine components must have exceedingly high reliability.

5.3 Types and Levels of Checking

It is convenient in discussing checking methods and devices to use the following terminology:

Repeat checking	The same operation is done again in the same way with the same equipment.
Multiple checking	The same operation is done in the same way, but using entirely separate pieces of equipment of the same construction.
Identity checking	Two equivalent but different methods of doing the same operation are employed, and the results compared. The two methods may make use of different equipment, or the same equipment in different ways.
Test checking	A test problem, for which the correct answer is known, is run through the machine before and after new problems are done. Presumably this problem can be designed to reveal most of the possible failures, and perhaps can help localize these failures.
Smoothness checking	The smoothness of a series of calculated values is examined, usually by higher order differences.
Programmed checking	Checking which is carried out in accordance with specific instruction orders (at the control of the operator).
Built-in checking	Checking which is not at the discretion of the operator, but is always done by the machine without instruction.

These categories are not all exclusive , nor exhaustive. The first five are types which are usually programmed, not built-in. There are numerous types of built-in checking which are hard to classify. One of the best is that used by certain relay computers which detect whether the proper number of relays have operated in each step of the problem, allowing no operation to proceed until the immediately prior operation has been checked in this way. This method is notable for the nicety with which it localizes errors, and for the high reliability which it achieves.

It is convenient to speak of various "levels" of checking. At the lowest level, every unit operation is checked, while at the highest level an entire problem might be multiple checked or identity checked on two distinct machines. Except for the rel'y check just mentioned, both of these extremes are unduly expensive, since intermediate methods can usually be found to insure adequate reliability combined with reasonable localization of failures.

It is obvious that a digital computing machine which has any claim to flexibility will certainly permit programmed checking of any sort desired. Smoothness checking, for instance, can be aided by automatic calculation of the required differences. The designer is concerned, however, with the question of what checking devices should be built-in. In answering this question, careful consideration should be given to the kind of failures which are likely to occur.

5.4 Character of Failures

Equipment failures can be broadly classified into these two groups:

Type 1 - lasting failures, which persist until corrected by the operator or maintenance personnel.

Type 2 - transient failures, for which the apparatus passes back and forth between correct and incorrect operation, usually but not necessarily at irregular intervals. Type 2 failures are often called "intermittents."

Type 2 faults are more difficult to detect and locate. If not detected immediately, there is no way of knowing how many calculations have been spoiled.

Probably one reason why relays have been preferred to vacuum tubes for use in large computing machines is that more is known about their failure characteristics. Type 1 faults are extremely improbable in good relays. Most failures are type 2, caused by dust particles fouling contacts for a few operations only, being shaken out by subsequent operations. These failures are reduced by using two contacts in parallel, but they still predominate over type 1. The step-by-step built-in checking circuit mentioned in Section 5.3 is therefore quite essential for computing reliability.

For vacuum tubes, quantitative information on failures is very rough, and comparisons are difficult because conditions of usage are not standardized. Some general facts about their failure characteristics are of considerable importance to any discussion of checking. Suitably tested tubes operated under proper conditions can have a useful life of about 10,000 hours. Type 2 failures are rare, and type 1 failures are high during the first few hours of operation, becoming low thereafter. Type 2 failures may arise from mechanical vibrations, loose connections, and such things, but the period or interval associated with these is usually long enough to cover many operations at the high electronic computing speeds. Extremely low-level checking is not needed to detect these.

As tubes age, their signal output gradually decreases, until at some time an occasional pulse may not affect the device it is supposed to operate. For this reason it is important to design computing circuits so they are insensitive to rather large changes in signal strength. The liklihood of type 2 failures from aging effects can then be greatly reduced by periodically testing all tubes. By other methods, potential type 2 failures can be forced into becoming type 1 failures which are readily located. There is no comparable procedure for relay machines.

5.5 ENIAC Checking and Reliability

The ENIAC has not been provided with any built-in checking devices in any strict sense of the term. However, there are circuits which see to it that all of the numerous power supplies are operating within certain tolerance limits, and other circuits which shut down the whole machine if certain fuses are blown. Another feature, described in Chapter 3, is also related to checking and localization of failures. The cycling unit is capable of operating the machine by single addition time steps, or by single pulse steps, each step controlled by a push button. All counters and other flip-flops are equipped with neon indicating lamps, so that many details of circuit operations can be followed visually during a manually-controlled step-by-step check. It is also possible to operate the machine, either "continuously" or step-by-step, at any pulse repetition rate lower than the standard 100,000 per sec., or even somewhat above this rate. Thus certain timing tolerances can be investigated.

Since the ENIAC has no built-in checking, careful attention has been given to improving reliability by circuit design. Tubes and other components are operated at very conservative ratings. Tubes are used as binary elements, being either on or off. Large safety factors are provided, both in signal strength and in timing. In general, all tolerances have been made consistently large.

Any kind of programmed checking may, of course, be used, except that there is no way of multiple checking operations such as multiplication, division and square root, for which duplicate equipment does not exist. Identity checking and smoothness checking appear to be reasonable methods. Considering the character of vacuum tube failures, as discussed in the preceding section, repeat checking is probably of no value at all since the same errors will probably be made during the second run. For exactly the same reason, test checking is very likely to be quite successful, and if this turns out to be the case, this will be the most efficient sort of general purpose checking. Only after considerable operating experience will it be known how frequently (or infrequently) test checking will have to be done. Whenever the nature of the problem allows it, smoothness checking should also be done, but this method does not reveal systematic errors.

5.6 Advantages of Serial Operation

From the standpoint of checking and reliability, completely serial machines have important advantages. In the first place, since less equipment is involved, there is less liklihood of a machine failure. The time lost when a failure occurs is also reduced, since both diagnosis and repair are easier with less equipment.

Quite apart from this, however, serial machines have a considerable advantage with respect to checking. Since there is less equipment to be checked, less equipment is needed to do the checking. In fact, a number of components can sometimes be checked by a single checking device. Test checking is easier to work out for a machine having fewer components and proportionately more of them used in each operation. Since many parts of a serial machine are in almost continuous operation, there is also more opportunity for detection and location of type 2 failures.

A very significant fact is that all of the digits for any number must pass through the same channel. This means that a type 1 failure in such a channel will affect all the digits of every number that uses that channel. It is quite probable that the calculations will be so completely upset that the failure will be obvious when the results are inspected, presuming that no built-in checking catches this failure. This is to be contrasted with the possibility, in a multi-channel machine, of a single digit failure causing only a very small error in the results. Such an error might be sufficiently systematic as to go undetected by smoothness checks.

When type 2 failures are considered, the fact that all digits must pass through the same channel is even more important. In a serial machine any type 2 failure will be almost certainly cause a gross error, and again might be noticed without the aid of built-in checking devices. In a multiple machine, if a type 2 failures occurs in a place where it dos not cause a gross error, the faulty component may cause errors only infrequently and thus go undetected during the course of many problems, unless identity checking or ingenious test checking is employed.

Although all failures in a serial machine are likely to cause gross errors, and would probably be noticed without built-in checking devices, it seems reasonable to build in any checking device which does not require a great deal of equipment.

5.7 Checking in the EDVAC

For the EDVAC, it is expected that most of the equipment will be associated with the memory. This means that the arithmetic organ and the control unit could be multiple checked by having duplicate equipment for these parts, and yet not increase the total equipment very much. The circuit which compares the results from the two computers can also be checked by having a duplicate comparing circuit, thus guarding against failure of this checking device.

To check the unit operations of the computer within any standard arithmetic process seems unnecessary. Reliability is sufficiently assured by the multiple checking of complete arithmetic processes, and checking at a lower level would involve much more equipment and serve only to localize failures. It is believed that such extreme localization is not necessary, since once a computing failure has been noticed either diagnostic test checking or electrical service instruments can be used to locate the fault more precisely.

Methods for checking the memory and switches will not be described here except in very general terms. It is probable that a kind of built-in test checking, requiring only a small amount of equipment, will be found suitable. Sufficient input and output equipment can be provided so that multiple checking can be used.

It is hoped that these built-in checking devices will improve the reliability to such an extent that the operator will not ordinarily have to devote much attention to programmed checking. If, after the EDVAC is put into operation, it is found that this hope is not entirely justified, then a certain amount of programmed checking might have to be done. In any event, all of the failures which are noticed by the built-in checking system will be to some extent localized and service problems will thereby be simplified.

APPENDIX A

Remarks on Arithmetic Operation of the ENIAC

The purpose of this appendix is to present in more detail the arithmetic operation of the multiplier and divider. To do this let us first describe the system for handling negative numbers in the ENIAC. Since the ring counters in the accumulators cycle in but one direction it is desirable to treat subtraction as a form of addition by the introduction of a system of complements.

Let us focus attention on the 10 decade counters of an accumulator. Any integer between 1 and $10^{10}-1$ inclusive, is uniquely representable on the counters but the numbers zero and 10^{10} are indistinguishable. Hence the sum $(10^{10}-N)+N$ appears as zero on the decade counters. The number $10^{10}-N$ is referred to as the complement of N with respect to 10^{10}.

To permit a unique representation both of positive numbers N with $0 \leq N \leq 10^{10}-1$ and of the complements of such numbers there is a so-called PM counter to the left of the decade counters. This counter is a binary one since its function is to distinguish between positive numbers (which carry sign P) and their complements (which carry sign M). In the transmission of a positive number no pulses are sent along the PM lead in a digit tray, whereas nine sign pulses are sent for a complement. The PM counter can receive as its input either the sign pulses or a carry-over pulse from the extreme left hand decade. The reception of an even number of pulses leaves the counter in its original state, the reception of an odd number of pulses has the effect of cycling the PM counter to its other stage. It is now easy to verify that all the usual arithmetic properties of addition and subtraction are obeyed in the system described. There follow a few illustrative examples: The sum of 801 and 527 appears as

$$\begin{array}{l} \text{P } 0,000,000,801 \\ \underline{\text{P } 0,000,000,527} \\ \text{P } 0,000,001,328 \text{ ;} \end{array}$$

the difference 801 minus 527 as

$$\begin{array}{l} \text{P } 0{,}000{,}000{,}801 \\ \underline{\text{M } 9{,}999{,}999{,}473} \\ \text{P } 0{,}000{,}000{,}274 \text{ ;} \end{array}$$

and the difference 527 minus 801 as

$$\begin{array}{l} \text{P } 0{,}000{,}000{,}527 \\ \underline{\text{M } 9{,}999{,}999{,}199} \\ \text{M } 9{,}999{,}999{,}726 \text{ .} \end{array}$$

The ENIAC is so constructed that the complement of a number is formed by subtracting each digit of the number from 9 and then transmitting an additional pulse over the lead associated with the extreme right hand digit as specified by the setting of the significant figures switch. This additional pulse has the effect of subtracting the right hand digit from 10 so that a complement with respect to 10^{10} is thus formed.

As was mentioned earlier, the divider-square rooter is essentially a mechanism for automatically sequencing the behavior of a number of associated accumulators. It performs this sequencing by generating at various times special programming pulses which are transmitted to the appropriate accumulators. The operation cycle, whether for division or square rooting, divides itself into four distinct phases; phase I in which the stage is set for the following phases; phase II during which the operation is performed; phase III, the round-off period; and phase IV the interlock and clear period.

In phase I the divider emits signals which stimulate the numerator and denominator accumulators to receive their arguments and sets up certain programming circuits. During phase II the basic division and shifting sequences are performed. When the numerator and denominator have like signs, the denominator is subtracted from the numerator and the quotient is increased by one in a particular decade; when the signs are unlike, the denominator is added to the numerator and the quotient is decreased by one in a particular decade. When the remainder changes sign, the basic division sequence is interrupted. The remainder is then transmitted from the numerator accumulator to the shift accumulator, shifted one place to the left and returned to the numerator accumulator. The basic division sequence is then resumed. After a shift, however, the unit, added or subtracted from the quotient, is put into the next decade to the right of the place in which it was previously accommodated. Phase II ceases and phase III starts when an overdraft occurs and when the specified number of answer places have been found. We shall not further discuss phases III and IV, but consider a numerical example of division on the ENIAC. See Problem 1, P. A-4.

The multiplier, as was remarked above, carries out its operation by forming successively partial products consisting of the entire multiplicand by one digit at a time of the multiplier. Arrays built into the multiplier store the multiplications tables for numbers between zero and nine in two parts: the tens digit and the units digit. For example, the product of 4 x 9 is remembered as a tens digit of 3 and units digit of 6. During the multiplication process, the tens digits of the partial products are stored in the so-called left-hand partial product accumulators and the units digits in the right-hand partial product accumulators. Upon the completion of the multiplication cycle the process of correction for multiplication by complements is effected. At the end of this phase the left-hand product is transferred into the right-hand accumulators. See Problem 2, P. A-5.

Phase	Quotient Accumulator		Numerator Accumulator		Denominator Accumulator	Shift Accumulator	
	Receives	Stores after receiving	Receives	Stores after receiving	Receives and stores thereafter	Receives	Stores after receiving
I			P0 209 070 000	P0 209 070 000	P0 230 000 000		
II			M9 770 000 000	M9 979 070 000			
	P0 100 000 000	P0 100 000 000					
						M9 790 700 000	M9 790 700 000
			M9 790 700 000	M9 790 700 000			
			P0 230 000 000	P0 020 700 000			
	M9 990 000 000	P0 090 000 000					
						P0 207 000 000	P0 207 000 000
			P0 207 000 000	P0 207 000 000			
			M9 770 000 000	M9 977 000 000			
	P0 001 000 000	P0 091 000 000					
						M9 770 000 000	M9 770 000 000

Problem 1

To perform the division P0 209 070 000 ·/. P0 230 000 000 to four figures without round-off.

	Left-hand Partial Product Accumulator		Right-hand Partial-Product Accumulator	
Multiplier Digit	Receives	Stores after Receiving	Receives	Stores after Receiving
2	P1 000 010 000	P1 000 010 000	P0 226 046 000	P0 226 046 000
0	P0 000 000 000	P1 000 010 000	P0 000 000 000	P0 226 046 000
8	P0 040 201 600	P1 040 211 600	P0 008 840 640	P0 234 886 640
Complement Correction	M7 920 000 000	M8 960 211 600		
Final Product Collection			M8 960 211 600	M9 195 098 240

APPENDIX B

Remarks on Programming the ENIAC

Inasmuch as the problem of programming the ENIAC has not been discussed in much detail above it is desirable to devote more space to this subject. Let us first consider an extremely simple problem as an illustration of an elementary programming procedure. In planning a set-up for the ENIAC it is desirable to link the elementary programming sequences into a complex whole by means of the master programmer. Hence we first consider the problem of programming the elementary steps and then plan the over-all connection of these steps by hierarchies of program sequences in the master programmer.

It is desired to form in two accumulators a tabulation of the function n^2 against n (cf. Fig. 1). Let us proceed inductively and assume that one accumulator contains n and another n^2. We then wish to program the first accumulator to transmit its contents twice into the second one, which must then also be programmed to receive twice. Hence we need to use one transceiver program control on accumulator one to cause the transmission and one such control on the second accumulator for the reception. In addition, we must use a program pulse to stimulate simultaneously each of these controls and we shall indicate below the origin of this pulse. The output signal from either control will then be used to stimulate a unit such as the constant transmitter to send the digit one into each of the accumulators. We need therefore to use one receiver program control on each of the two accumulators to receive this digit. Upon the reception by each accumulator of the unit there appear the numbers $(n+1)$ and $n^2 + 2n \;\; 1 = (n+1)^2$. Finally the program output of the constant transmitter may be fed into the transceiver controls mentioned above to stimulate the reiteration of this process. It remains only to describe the inception of the entire cycle. Initially the accumulators contain the number zero, and the initiating pulse is sent to the constant transmitter for the purpose of stimulating the emission of the digit one and to the accumulators to stimulate reception of the digit. Fig. .1 shows a schematic diagram for the set-up of this problem.

Suppose it is desired to form the above tabulation for $n \leq 13246$ and to stop at $n = 13246$. We then wish to make use of the master programmer to count the number of times we carry out our process. To use the master programmer for this purpose we associate with, for example, the C stepper five decades. The five decade switches associated with the first stage of the stepper are set to 1, 3, 2, 4, 5 respectively. The program output of accumulator 2 in Fig. 2 is now routed to the stepper input and the output of stage 1 is then sent to the program pulse input terminals on accumulators 1, 2 and the constant transmitter (cf. Fig. 2). After the computation of n^2 has been stimulated 13246 times (once by the initiating pulse and 13245 times from the output of stage 1 of the stepper), the stepper cycles to stage 2. The program output pulse from stage 2 of the stepper does not cause any further computations since the output terminal for this stage of the stepper is not connected to any program line.

Evidently we could have used the output of stage 2 to initiate a new and different sequence of events. It is therefore clear that one can, by such simple measures, order the execution of a sequence of subsequences of operations and by suitably interconnecting steppers achieve considerable programming complexity in this manner.

Instead of pursuing this obvious topic further let us discuss other uses of the master programmer. The use of the master programmer is being stressed since it is the mechanism in the ENIAC which enables one to link the simple sequences of instructions given the other units of the computer into a complex whole.

To illustrate another important function of the master programmer, let us consider the problem of programming the ENIAC to discriminate between two program sequences depending on the relative sizes of two numbers a and b. For example, in the integration of a differential equation it may be desired to reiterate a given step if the third difference, a, of a certain quantity is not less than a fixed number, b, and to proceed to the next step in the contrary case. After completing a step of the integration we accordingly wish to form the difference a-b and to examine its sign indication. Recall that if a number is non-negative, no pulses are sent to indicate the sign, whereas, if it is negative, nine pulses are transmitted. Let us now connect the output of stage 1 of a stepper to the program line which carries the stimulating pulse for the sequence of computations involved in reiterating, and the output of its second stage to the line which carries the stimulating pulse for the next step. We now connect the sign indication lead from the accumulator storing the difference a-b to a transceiver whose output is fed to the direct input of our stepper, and connect a program line to the stepper's input. We shall send a program pulse along this line in the same addition time as the transceiver's output will occur. If (a-b) is non-negative, no pulses are applied to the direct input of the stepper since no pulses reach the transceiver. Moreover, the pulse applied to the stepper input causes an output pulse on its stage 1 output, thereby stimulating the reiteration program. If a-b is negative, the stepper is advanced to stage 2 since the transceiver is stimulated. The pulse applied to the stepper input, in this case, causes an output from stage 2, which results in the next line's being computed. Finally a program pulse is sent to the stepper clear direct input so that the stepper returns to stage 1 for the next magnitude discrimination (cf. Fig. 3).

As an extremely simple example of the way in which one may compound sequences of program instructions consider the following problem: Suppose it is desired to read certain data, such as initial conditions from an IBM card, carry out a computational routine of m line steps, print the final results, and then perform the same routine n times. We could then use stage 1 of stepper (with the stage one decade switch set at 1) to signal for the card reading, stage 2 to order the computations to be performed while its associated decade switches count the m repetitions, and stage 3 to control the printing operation (cf. Fig. 4).

Still another master programmer stepper is used to stimulate n repetitions of the read-compute-print sequence. The initiating pulse is sent to this stepper, and the resulting output pulse from stage 1 stimulates the first read-compute-print sequence. The terminal pulse of the read-compute-print sequence is also delivered to this stepper.

In closing let us consider the means at our disposal for performing a large number of multiplications. It is clear that in planning a problem one must determine what is a basic group of elementary arithmetic operations. By this is meant a group of operations, which cannot be built up by iteration of any of its sub-groups. An illustration of this notion can be gained by considering a pair of differential equations. The integration performed on one variable is essentially the same as that performed on the other. Hence it suffices to arrange the computation so that onlv one of the integrations is set up on the arithmetic units and to have the master programmer provide for the iteration. In this fashion, problems involving numbers of multiplications far in excess of 24 can be programmed.

If, however, the basic group of operations itself requires more than 24 multiplications, it is necessary to resort to other techniques to obtain the necessary freedom. These other techniques require the expenditure of a number of transceivers for non-arithmetic purposes. As will be seen below, to achieve six multiplications in the basic group at the expense of but one multiplier control necessitates the use of about 30 of these transceivers, i.e., about 10 per cent of the program control capacity. In general, in planning a set-up of a problem for the ENIAC the inner economy of the machine must be considered in allocation program facilities to various parts of the problem.

In closing let us consider the means at our disposal for performing a number of multiplications in our basic group, which are essentially alike in the following sense: the multipliers a_i ($i = 1, 2, \ldots, n$) and multiplicands b_i are received on the same channels; the products $c_i = a_i b_i$ are transmitted on the same lines; there are the same number of digits in the multipliers; the products c_i are rounded-off to the same number of places; and the multiplier, multiplicand and product accumulators require the same clearing or holding instructions. Evidently the simplest procedure for handling the problem is to devote one multiplier program control to each of the n multiplications. This technique is, however, not always possible as, for example, in the case of a problem requiring more than 24 multiplications.

An alternative procedure is to use but $[n/6] + 1$ multiplier controls, where $[x]$ is the largest integer $\leq x$, and the same number of steppers.

We illustrate by showing how to achieve six products $c_i = a_i b_i$ ($i = 1, 2, \ldots, 6$) by the use of one stepper and one multiplier control and a number of dummy program controls. We first define a dummy program control as any transciever which is used to perform no arithmetic operation but serves only to give an output signal after one or more addition time delays. To return now to our problem let us send the initiating pulse into three dummy controls and use their outputs to stimulate the transmission of a_1 b_1 and to stimulate the multiplier to produce c_1. By means of these dummy controls we have isolated the order to the multiplier from the orders calling for the constants a_1, b_1.

We now send the output signal from the multiplier into our stepper and send the output pulse of stage one to four more dummy controls which stimulate the disposal of the product c_1, the transmission a_2, b_2 and the multiplication a_2, b_2 by the same multiplier control. The output signal of the last multiplication is sent to the stepper and output of stage 6 can be used, as indicated above, to stimulate a new multiplier control (cf. Fig. 5).

As a closing illustration let us consider the previous problem but let us set the problem up in a different way eliminating the use of the master programmer. For simplicity we suppose that our multipliers are always five digit numbers. This time we send our initiating pulse to four dummy controls, the first three of which are used as before and the last of which gives a nine addition time delay. When the multiplication is completed, we do not use the multiplier output pulse. Instead we take the delayed signal from our fourth dummy and use that signal to stimulate five more dummy controls in the obvious way. (cf. Fig. 6.)

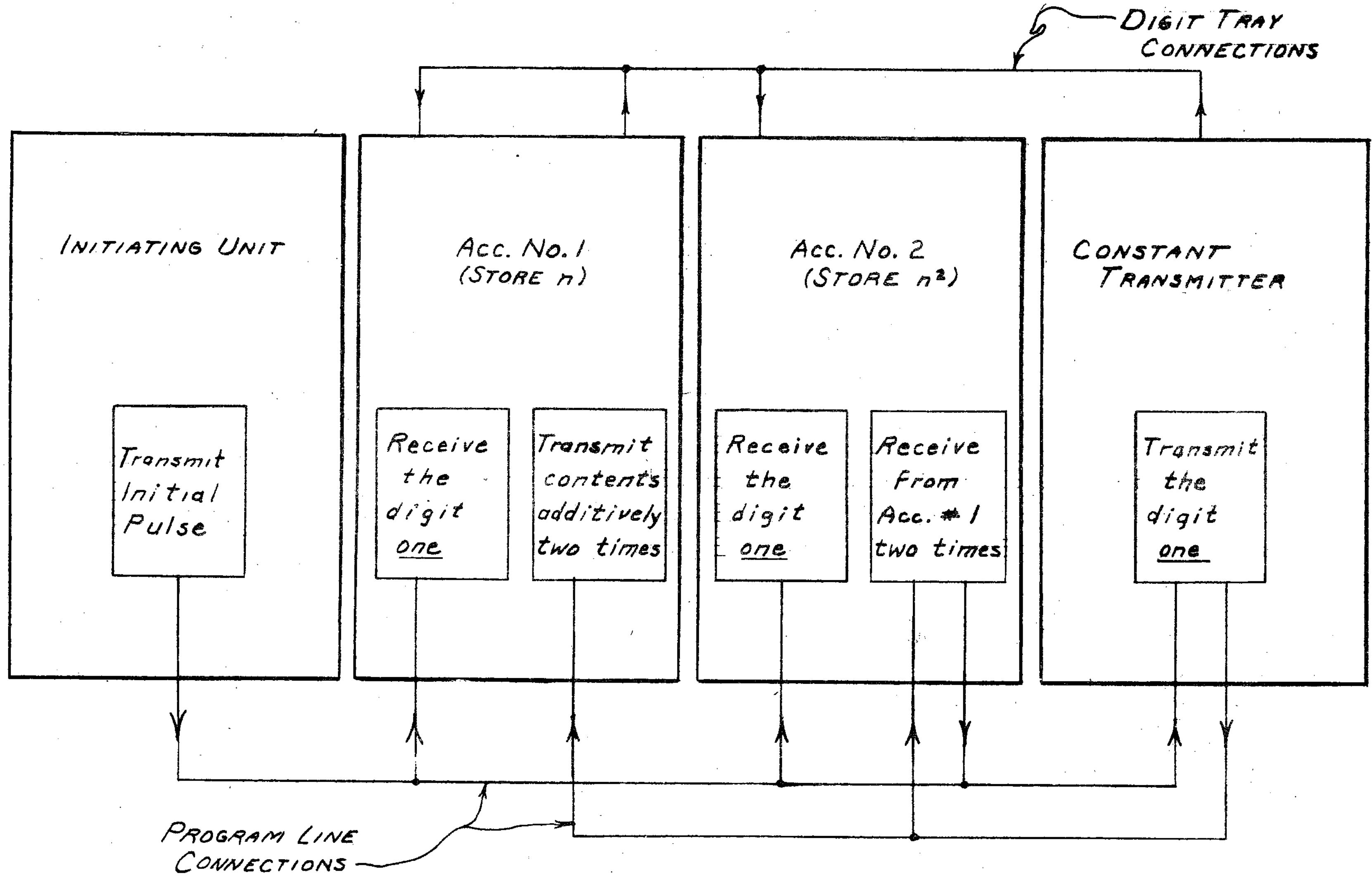


FIG. B-1
SCHEMATIC DRAWING OF PROGRAM SET-UP
FOR GENERATING n^2

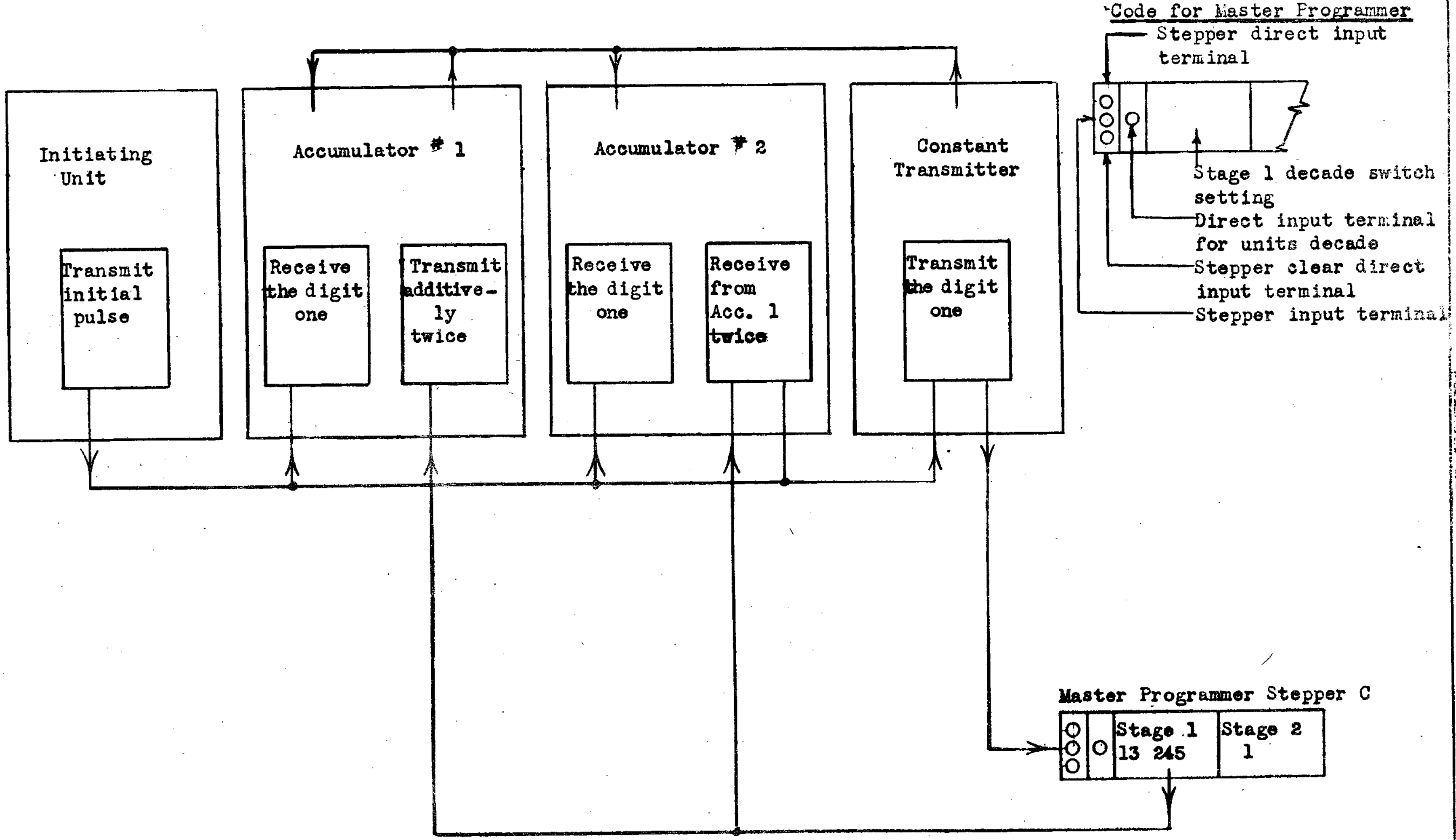
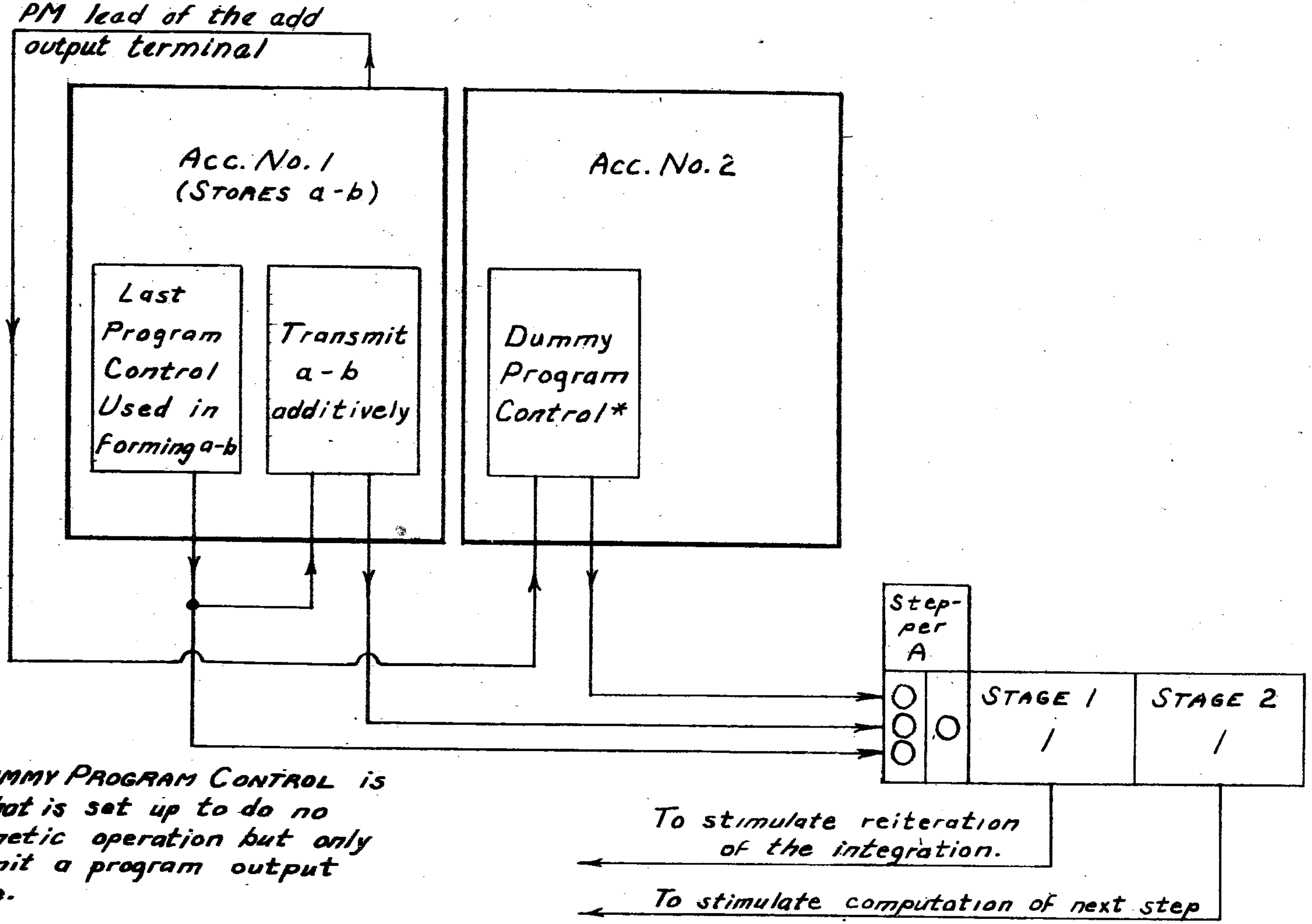

Code for Master Programmer
Stepper direct input terminal
Stage 1 decade switch setting
Direct input terminal for units decade
Stepper clear direct input terminal
Stepper input terminal
Initiating Unit
Transmit initial pulse
Accumulator # 1
Receive the digit one
Transmit additive-ly twice
Accumulator # 2
Receive the digit one
Receive from Acc. 1 twice
Constant Transmitter
Transmit the digit one
Master Programmer Stepper C
Stage 1
13 245
Stage 2
1

Fig. B-2

Schematic Drawing of Program Set-Up For Generating n^2 as long as $n \leq 13\ 246$


PM lead of the add output terminal
Acc. No. 1 (Stores a-b)
Last Program Control Used in Forming a-b
Transmit a-b additively
Acc. No. 2
Dummy Program Control*
Step-per A
Stage 1
1
Stage 2
1
To stimulate reiteration of the integration.
To stimulate computation of next step

* A Dummy Program Control is one that is set up to do no aritmetic operation but only to emit a program output pulse.

Fig. B-3
Use Of The Master Programmer To Choose The Next Program In Accordance With The Result Of Magnitude Discrimination.

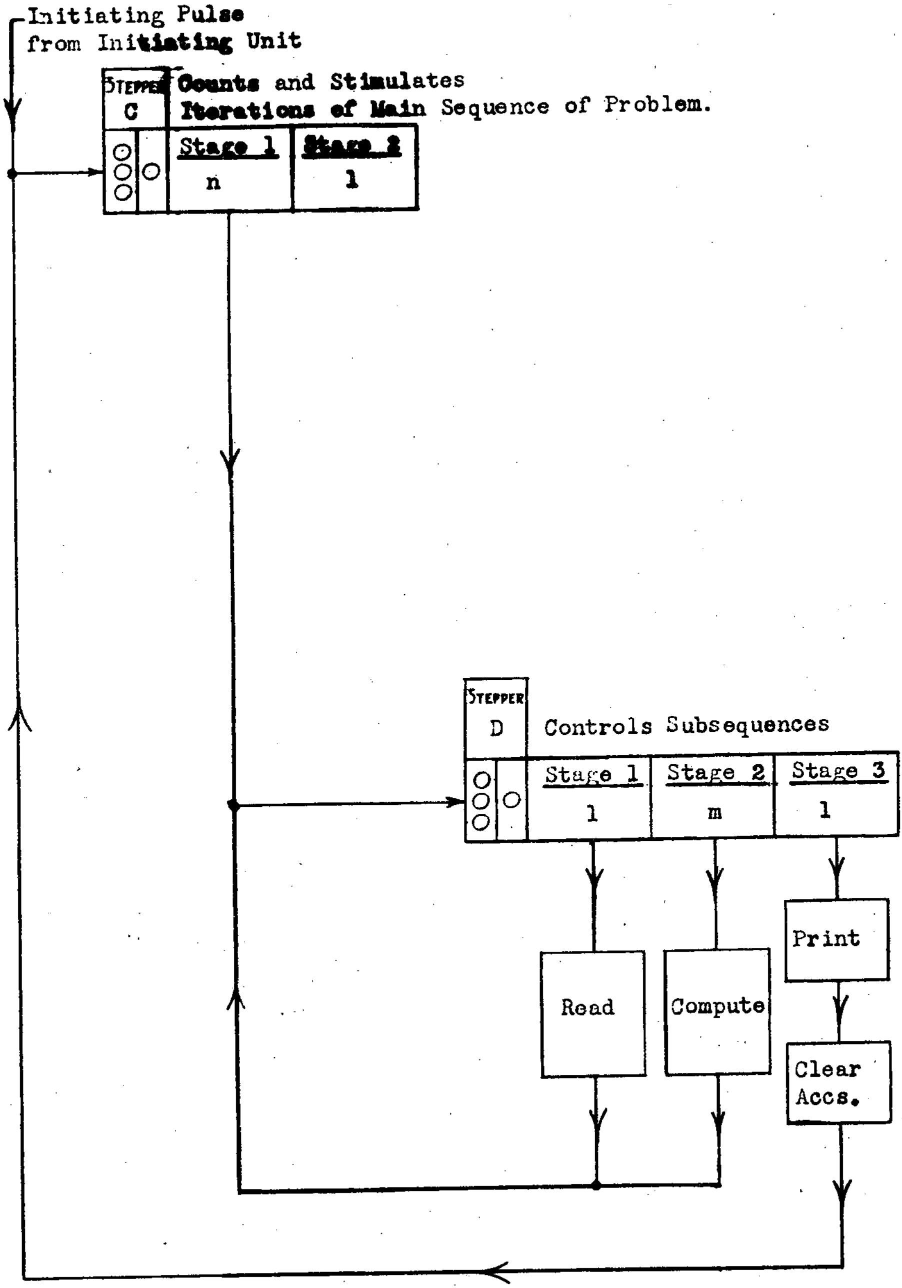
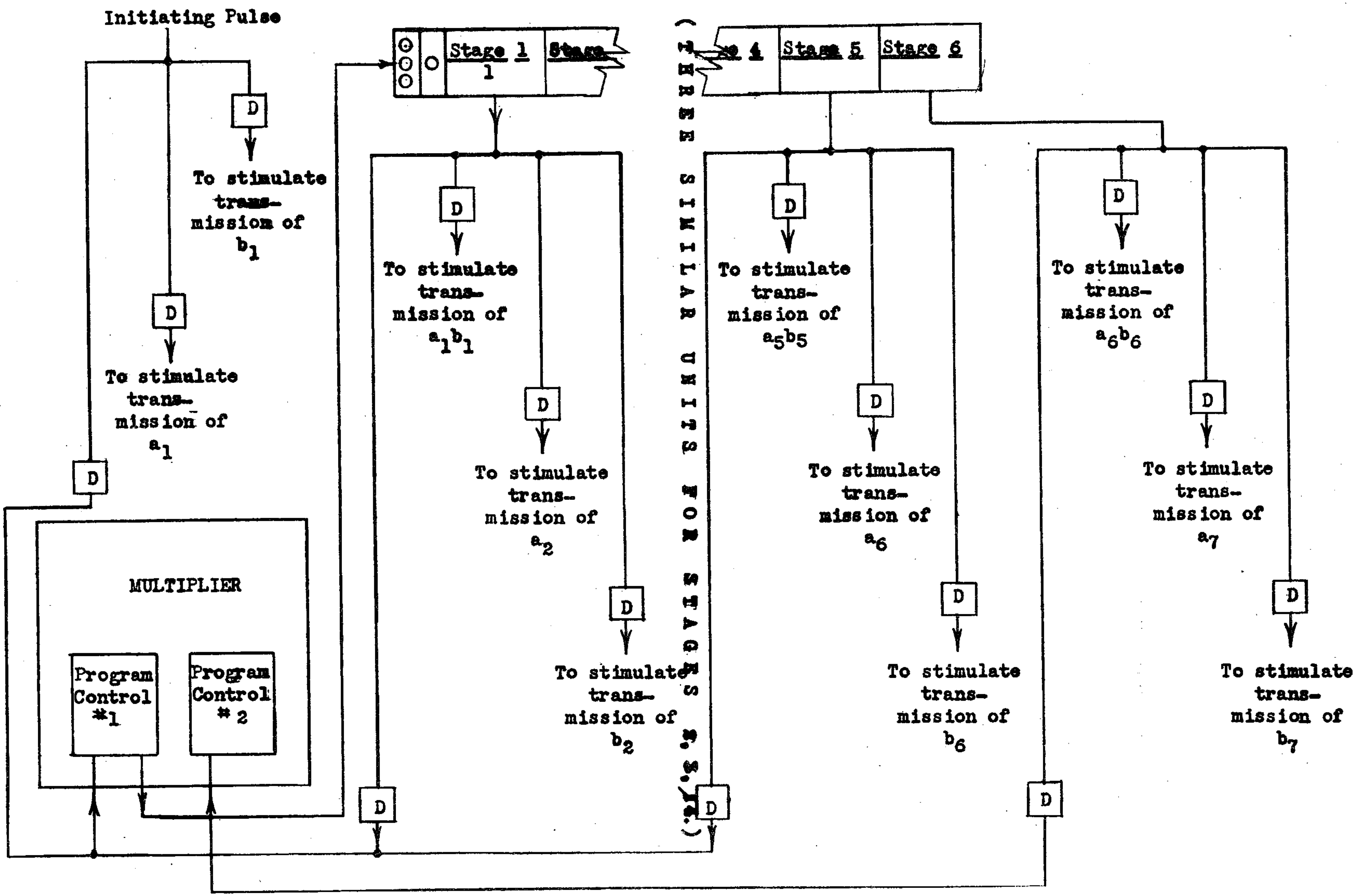


Fig. B-4

Use of Master Programmer to Link Sequences of a Problem.

RESTRICTED



Initiating Pulse
D
To stimulate trams-mission of b_1
D
To stimulate trans-mission of a_1
D
MULTIPLIER
Program Control *1
Program Control # 2
Stage 1
1
Stage
D
To stimulate trans-mission of a_1b_1
D
To stimulate trans-mission of a_2
D
To stimulate trans-mission of b_2
D
(THREE SIMILAR UNITS FOR STAGES 2, 3, 4)
e 4
Stage 5
Stage 6
D
To stimulate trans-mission of a_5b_5
D
To stimulate trans-mission of a_6
D
To stimulate trans-mission of b_6
D
D
To stimulate trans-mission of a_6b_6
D
To stimulate trans-mission of a_7
D
To stimulate trans-mission of b_7
D

Fig. B-5

Use of Master Programmer to enable use of single multiplier program control on six occasions.

RESTRICTED

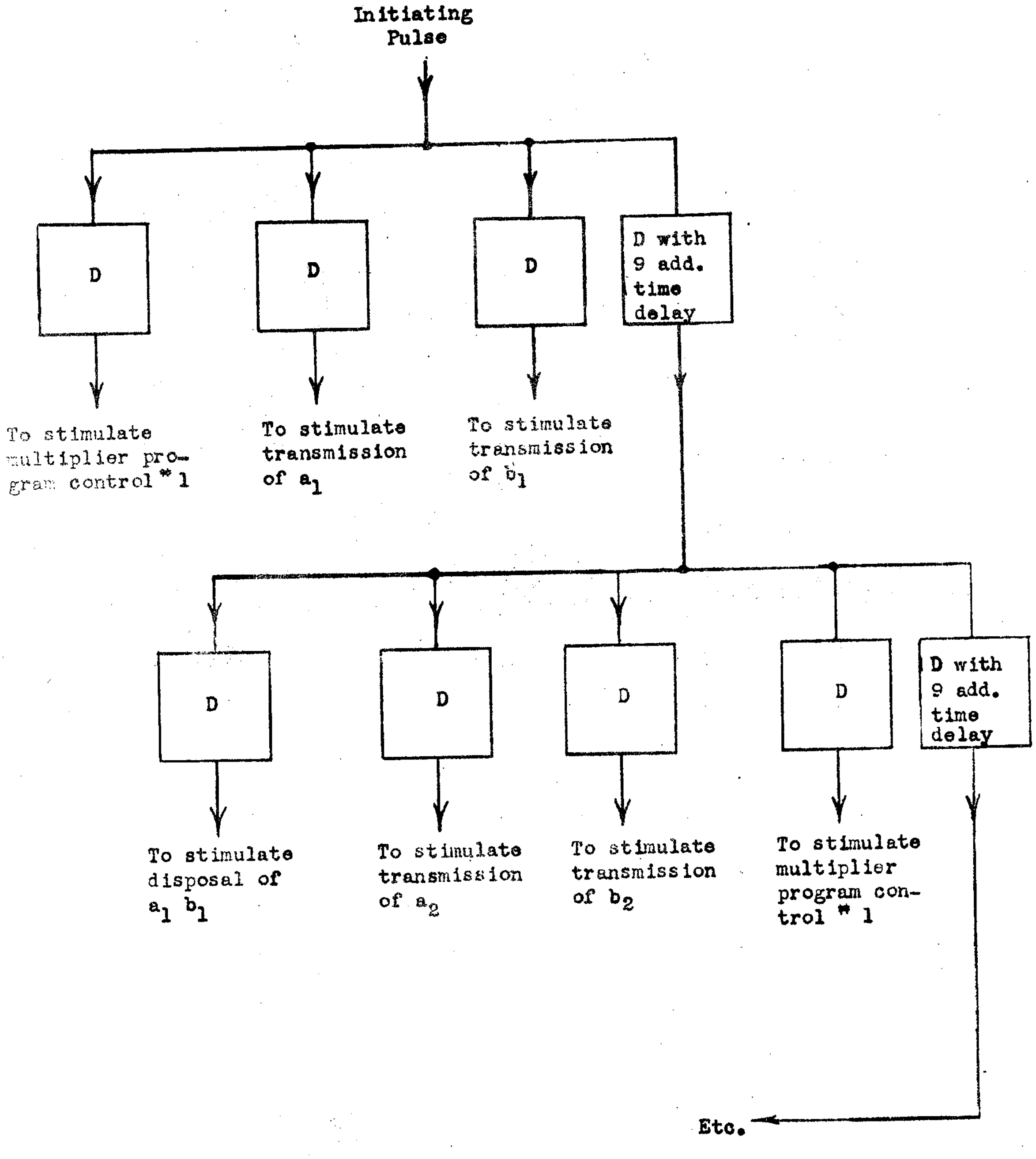
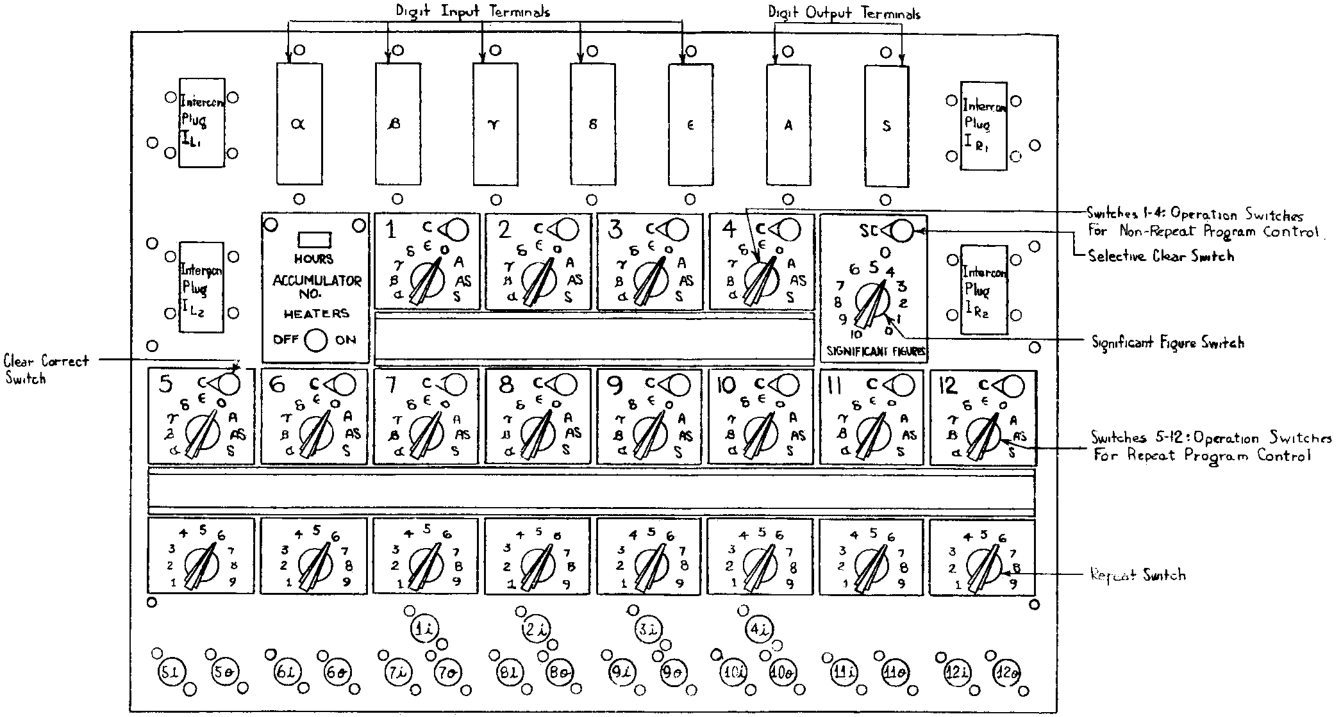

Initiating Pulse
D
D
D
D with 9 add. time delay
To stimulate multiplier program control # 1
To stimulate transmission of a_1
To stimulate transmission of b_1
D
D
D
D
D with 9 add. time delay
To stimulate disposal of a_1 b_1
To stimulate transmission of a_2
To stimulate transmission of b_2
To stimulate multiplier program control # 1
Etc.

Fig. B-6

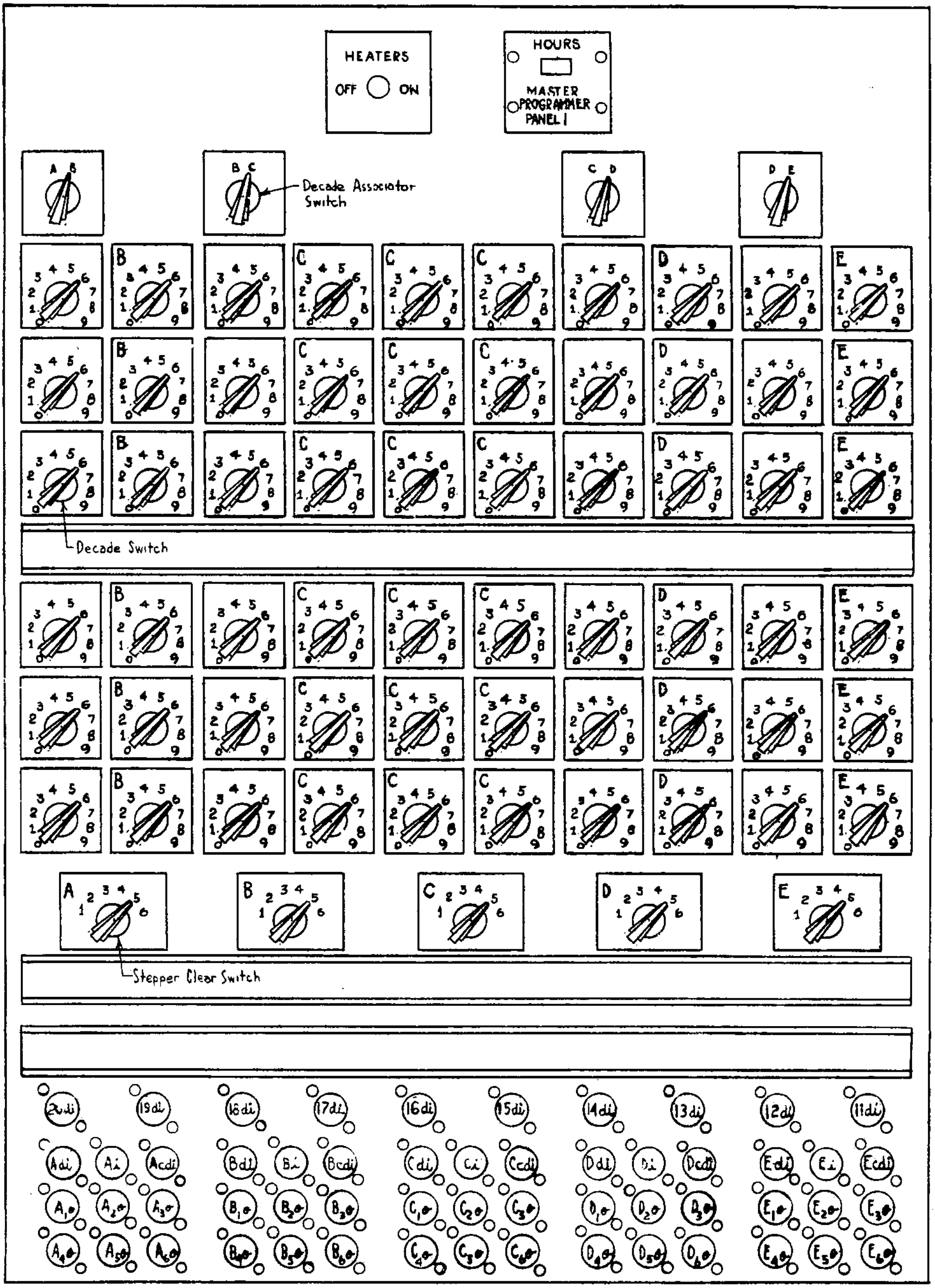
Use of a chain of dummy program controls to make possible the repeated use of a single multiplier program control



TERMINALS 1i, 2i, - - -, 12i
Program pulse input terminals

TERMINALS 5o, 6o, - - -, 12o
Program pulse output terminals

ACCUMULATOR
FRONT PANEL
PX-5-301R

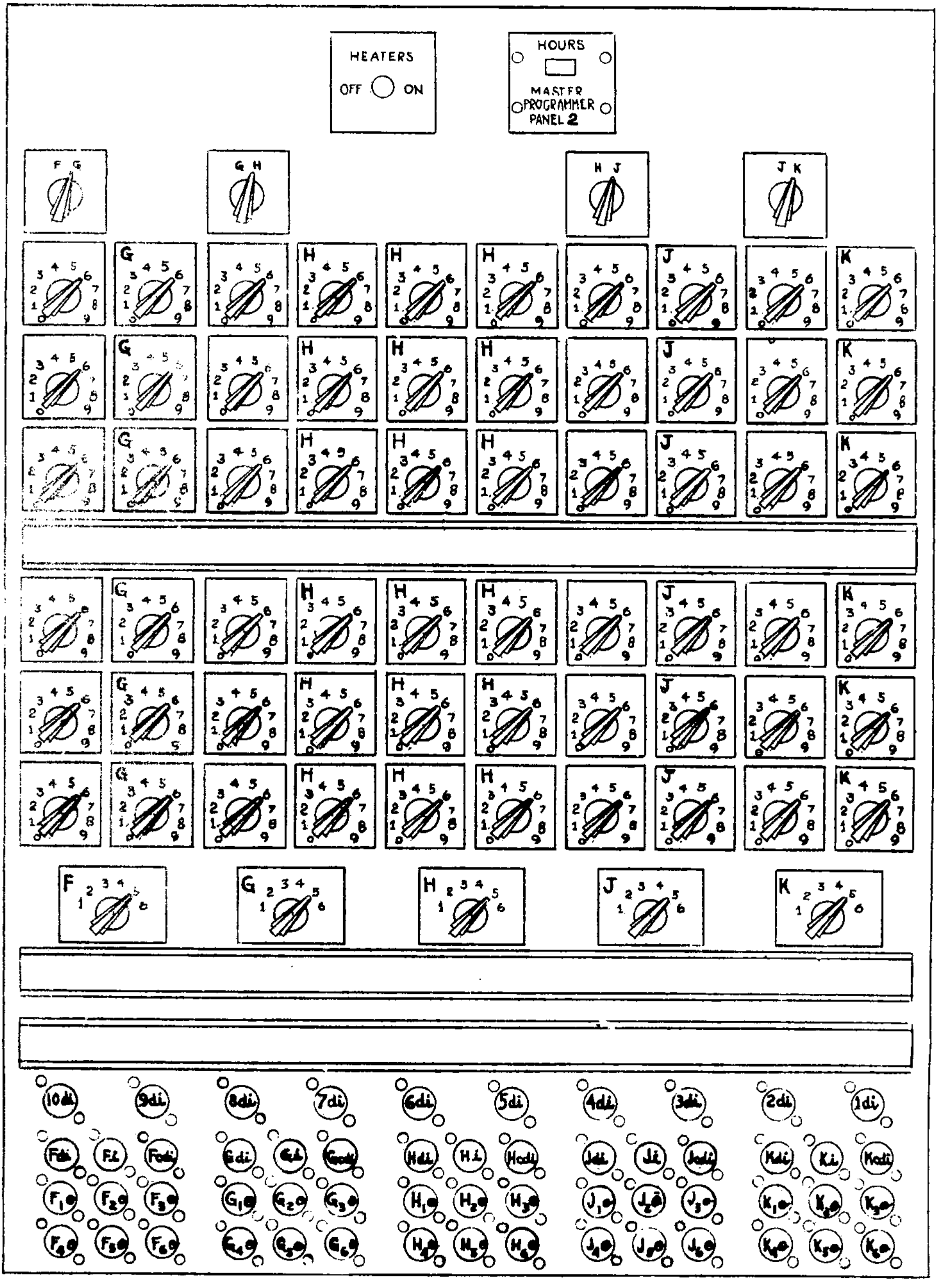


TERMINALS Ai, bi, ---, Ki - Stepper input
TERMINALS Adi ----, Kdi — Stepper direct input
TERMINALS Acdi ---Kcdi — Stepper clear direct input

TERMINALS A_1o, A_2o, ---, A_6o
Stepper output terminals associated respectively with stages 1, 2, ---, 6 of stepper A.

MASTER PROGRAMMER
FRONT PANEL NO 1

~~RESTRICTED~~



TERMINALS 1di, 2di---, 20di — Decade direct input terminals associated respectively with decades 1, 2, ---, 20 (counted from right to left).

MASTER PROGRAMMER
FRONT PANEL NO. 2
PX-8-302R

~~RESTRICTED~~

APPENDIX C

GENERAL CONSTRUCTION DATA ON THE ENIAC

The ENIAC is a very large and complicated machine. The electronic and relay apparatus, exclusive of the portable function tables and the auxiliary card machines, which are separate units mounted on wheels and plugged into the ENIAC, is mounted in forty panels each of which is two feet wide and eight and one-half feet high. Including the cases which house the tubes and air filters, the units are about two and one-half feet deep. Over thirty thousand cubic feet of air is forced per minute through the air filters and around the tubes by ten two-horsepower blowers. The hot exhaust air, which is between ten and twenty degrees Fahrenheit above room temperature, is passed through sheet metal ducts out of the building in which the machine is housed. In addition to the forty panels above, six similar panels of greater depth and two of smaller depth house the power supply equipment.

The forty main panels of the ENIAC are arranged in a large U with sixteen of the panels on each leg of the U and eight panels on the end. Thus U, together with the eight power supply panels, the three portable function tables, and some of the auxiliary card machines, are arranged in a room thirty feet by fifth feet. Since the room has a ceiling which is over eleven feet high, the blower motors and ventilating ducts are suspended from the ceiling over the machine in order to save floor space. Viewing the U from the inside, the forty main panels of the ENIAC arranged from left to right are:

1. Control and Initiating Unit

2. Cycling Unit

3.)
4.) Master Programmer (a two-panel unit)

5.)
6.) First Function Table (a two-panel unit)

7.)
8.) Accumulators 1 and 2 (two one-panel units)

9. Divider and Square Rooter

10.)
11.)
12.)
13.) Accumulators 3, 4, 5, 6, 7, 8, 9, and 10
14.) (Eight one-panel units)
15.)
16.)
17.)

18.)
19.) Multiplier (a three-panel unit)
20.)

21.)
22.)
23.)
24.) Accumulators 11, 12, 13, 14, 15, 16, 17 and 18
25.) (Eight one-panel units)
26.)
27.)
28.)

29.) Second Function Table (a two-panel unit)
30.)

31.) Third Function Table (a two-panel unit)
32.)

33.) Accumulators 19 and 20 (two one-panel units)
34.)

35.)
36.) Constant Transmitters (a three-panel unit)
37.)

38.)
39.) Printer (a three-panel unit)
40.)

Nine digit trunks of eleven wires each and ninety-nine program trunks of one wire each are mounted in trays which are stacked on shelves at the front of the machine. The digit trays are placed on a centrally located shelf, while the program trays are stacked at the base of the machine. Both sets of trays as well as the frame of the machine are made in eight foot, or four panel sections. Front panels containing the input and output terminals and the various program control switches are mounted between the two tray assemblies. The connecting plug and cable assemblies are plugged into sockets on the front panels and on the trays. The neon bulbs on the accumulators are visible above the digit trays, while other neon bulbs, arranged to show the state of the programming equipment, are visible just below and above the front panels.

If more than nine digit channels or ninety-nine program channels are required, they can be obtained by removing the plug and cable assemblies between one or more adjacent pairs of trays, thus giving a greater number of total channels some of which will no longer extend around the entire machine. At the base of the machine, running along the back of the panels, a single trunk constructed of digit trays is provided to connect the cycling unit to the other machines. This is ordinarily a semi-permanent connection - a connection which is only changed for servicing the equipment.

The panel of the control and initiating unit contains two meters and a small oscilloscope for measuring the filament voltages, the DC voltages,and the AC or hum component of the DC voltages. The cycling unit contains a larger oscilloscope which may be used to examine the various signals which are generated by this unit.

The ENIAC contains between seventeen and eighteen thousand vacuum tubes which are mounted along the back surface of thirty-seven of the forty panels. They are mounted horizontally just inside the dust covers which support the air filters.

The other three panels of the ENIAC contain over fifteen hundred automatic telephone exchange relays mounted in a similar manner. Thus each main panel contains either about five hundred vacuum tubes or about five hundred relays. Over seventy thousand resistors, about ten thousand condensers and five thousand switches comprise, with the vacuum tubes, the numerous elements or components of the ENIAC.

The ENIAC consumes one hundred and fifty kilowatts. This power is supplied by a three phase regulated two hundred and forty volt, sixty cycle power line. The power consumption may be broken up as follows: eighty kilowatts for heating the tubes, forty five kilowatts for generating DC voltages, twenty kilowatts for driving the ventilating blowers, and five kilowatts for the auxiliary card machines.

While the ENIAC does not contain any built-in checking equipment, a number of points were given special attention in its design. First, very conservative tube ratings, well below those of the manufacturer, were used. Second, more than sixty per cent of the tubes were mounted in small units which can be easily removed for repair. This was done in all of the more complicated circuits. Third, the signal or voltage level of the circuits was generally maintained at approximately forty volts and in practically all circuits in which sharp pulses were transmitted, shielded conductors were employed. A few exceptions to this rule were made under extenuating circumstances. Fourth, the ventilating system not only cools the room in which the machine is housed (which is only a few degrees warmer than normal) but also insures that the many resistors, which are already operating well under their rating, are functioning at a conservative temperature. Fifth, and finally, not only was great care given to the selection of such parts as switches and plugs, which have their contact surfaces silver plated, but all components and tubes were carefully tested before being installed.

In addition, special test equipment allows many of the circuits to be easily tested. This equipment includes a special test bench with its own power supply and electronic and oscillographic equipment so that the small spare units may be tested without interfering with the operation of the machine.