

THE LOGICAL DESIGN OF
AN INTERMEDIATE SPEED DIGITAL COMPUTER

A thesis submitted to the Graduate School of
the University of Wisconsin in partial fulfillment
of the requirements for the degree of Doctor of
Philosophy.

by
Gene Myron Andahl

Date October 11 , 19 51

To Professors: Peterson, H.A.

Arnold

Sachs

This thesis having been approved in respect
to form and mechanical execution is referred to
you for judgment upon its substantial merit.

C. D. Elvehjem
Dean

Approved as satisfying in substance the
doctor thesis requirement of the University of
Wisconsin.

Harold A. Peterson

K. J. Arnold

R. G. Sachs

Major Professor

Date Oct 18, 19 51

THE LOGICAL DESIGN OF
AN INTERMEDIATE SPEED DIGITAL COMPUTER

by

GENE MYRON AMDAHL

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of
DOCTOR OF PHILOSOPHY
at the
UNIVERSITY OF WISCONSIN

1951

~~INZX
AM 27~~

AW
AM 27

765855

DEC 13 1951 ALA2614

TABLE OF CONTENTS

Chapter	Page
Preface	ii
I. Solution of Problems on a Digital Computer	1
1. Introduction	1
2. Flow of Calculation	3
3. Flow of Information	9
4. Order System	11
II. The Proposed Logical Design	18
1. Introduction	18
2. The Magnetic Drum	20
3. The Use of the Drum	22
4. First Design Feature	24
5. Component Requirements	25
6. Second Design Feature	26
7. Component Requirements	33
8. Modifications	35
III. Choice of Components and Their Operational Requirements	39
1. Introduction	39
2. Division of Drum	40
3. Supplementary Features	43
4. Order Interpreter	49
5. Time and Track Selectors	53
6. Arithmetic Unit	54
7. Input and Output	69
IV. Evaluation	71
Appendix I - Program of a Simple Calculation	77
Appendix II - Survey of Present Machines . .	84
Appendix III - Survey of Computing Machine Components	99
Tables and Diagrams	106
List of References	124

PREFACE

The objective of this thesis is to describe the logical design of a general purpose digital computer presenting the arguments underlying the final choices of operation and components. The scope of the thesis is limited to the logical design, including the requirements of all operational components. It is worthy of note that a comparatively large freedom of physical design exists, satisfying the logical requirements which are imposed.

The appendices will include sufficient background material on existing computing machines and components, so that a reasonably careful comparative evaluation may be made from the material presented.

The sources of material for this discussion are very limited and, in many cases, are rather nebulous. In most cases the sources have furnished a background which facilitated the formulation of the overall views presented. Although portions of the thesis may appear to present material without due credit, in most cases the formulation is entirely independent, and if duplication occurs, it is without the knowledge of the author.

In particular, the author wishes to acknowledge three sources of informative assistance: The Aberdeen Proving Ground, Maryland, for the opportunity of studying the coding system for the EDVAC, thus learning a great deal about generalized logical systems of computing machines;

Engineering Research Associates, Saint Paul, Minnesota, for their assistance in evaluation of the computer and for suggestions for simplification of the controlled delay units in the arithmetic components; Charles H. Davidson, University of Wisconsin, for detailed checking of the logical operation, a number of helpful criticisms and suggestions.

At present work is being done on design and testing of physical components of this machine, and arrangements have been made to build it at the University of Wisconsin. Because of its inherent integral synchronization, the machine has been named the WISC, Wisconsin Integrally Synchronized Computer.

I. SOLUTION OF PROBLEMS ON A DIGITAL COMPUTER*

1. Introduction

Perhaps the simplest and most profitable approach to the understanding and designing of any physical device is a thorough study of the operations the device may be called upon to perform. In the case of a digital computer this entails a study of computation. This chapter will survey the steps involved in most computation and will emphasize in particular the aspects of routines which may be iterative in any one part of the computation or which may recur in several different parts of the computation. Very few detailed examples will be employed, but it is hoped that the generality of the discussion will make evident the generality of the concepts involved.

The initial step in carrying out any computation is the planning of some procedure which leads to the desired result. The initial procedure will in general contain: (1) organization and reduction of initial data into a readily usable form; (2) arrangement of a sequence of operational steps such that data required to perform a particular operational step are available in the form of initial data or as the results of previous steps, these steps leading, in general, to some point of discrimination and/or iteration; (3) repetition of the previous procedure with different

* The original source of this material is reference 6.

initial data to provide a table of results or to provide some means of controlling changes of initial data to approach results possessing desired properties, such as boundary conditions.

The next step in the computational procedure is to make a further breakdown of the initial plan. In general, this will leave a series of elementary operations to be carried out, such as addition, subtraction, multiplication, division, searching tables of functions, or comparison of results at any point to determine convergence. The less elementary operations such as extraction of square roots, integration, or generation of simple functions may, of course, also be reduced to the same set of elementary operations. It is of interest to note at this point that the reduction of these less elementary operations into the elementary ones involves the same three major steps that the initial procedure for the overall solution involved. This simple, but very general, logical scheme of computation may easily be diagrammatically formulated, and, with the introduction of a very limited terminology, the complete breakdown of a computation may be easily described or carried out.

Before going into greater detail on the breakdown of computation, it is advisable to give a very simplified description of a computing machine. A digital computer in general consists of five major components (see Figure 1):

- (1) an INPUT device which supplies information (orders and

numbers) to the computing machine; (2) a MEMORY system which stores information until the desired time and presents the information to the desired place upon instruction; (3) an ARITHMETIC component which performs the desired operations on numbers upon instruction; (4) a CONTROL unit which interprets orders and supplies corresponding instructions to all other units, the orders being supplied by the MEMORY or INPUT or both and under the control of the CONTROL unit itself; and (5) an OUTPUT device which prints or punches the results of a computation.

Numbers and orders are filed in the MEMORY by "location", that is, to obtain a "word" (order or number) from the MEMORY, one must specify the spatial (and in some cases temporal) position of this word in the memory medium.

2. Flow of Calculation

This section contains the diagrammatic formulation of the calculational procedure as it is carried out in a computing machine or in hand computation.

Figure (2) shows the general scheme, which may represent the whole computational scheme and is then called a "sequence", or which may represent an individual complex operation and is then called a "subsequence". This diagram is composed of "blocks of orders" stored in the memory system of the computer, each block performing a particular function, and these blocks being appropriately interconnected; such a diagram is called a "flow diagram". This

diagram represents the flow of calculation and should not be confused with flow of information.

a. Setup Block

This flow diagram of a sequence, or subsequence, begins just after the point (α), where block number (I) is reached. The first block is a "setup block" in which the reduction of initial data and preparation of the following block for the first calculation is carried out. In many subsequences this setup block provides the means of obtaining an initial approximation of sufficient accuracy to reduce materially the number of iterations required to converge to the correct results; for instance in a square root subsequence the determination of the position of the decimal point would be of great assistance. Also, in numerical integration of a differential equation, the process of point by point prediction and correction is handled in a different fashion at the beginning point, or points, than it is throughout the rest of the range of integration. In such cases the setup block would contain the initial calculation procedure.

b. Operation Block

The second block (II) is called an "operation block". In this block the main body of the calculation is carried out. In an iterative process this block performs the complete set of operations required to obtain the $n+1$ st approximation from the n th approximation, or to predict the

value of a function at the $n+1$ st point from the known values at the previous points. In this operation block the calculational process may be sent through a number of subsequences in order to carry out any non-elementary operations which may be required, such as the extraction of a square root.

An example of the programming of a simple problem is given in Appendix I.

c. Alternative Block

Block (III) performs a comparison and is called an "alternative block". The comparison may be between the new approximation and the previous one, to see if the convergence is sufficient. The difference between two approximations is in general required to be smaller than some previously assigned number ϵ . The comparison may be made upon index as well, that is, if a fixed number of steps is required, the number of times that the calculation has proceeded through the operation block can be recorded and compared with the stipulated number of steps. As the name "alternative" implies, there is a choice, a choice of returning and going through the iterative process once more, or of going on to the next stage of the computation.

d. Substitution Block

If the alternative block sends the calculation through the iterative process once more, the next block reached is the "substitution block" (IV). This substitution

block is essentially a restricted version of the "setup block", in that it prepares the operation block for the $n+1$ st operation by changing index and indexed numbers in the required places, just as the setup block prepared it for the first operation. The use of indices, and the necessity of changing them, is to be found in the approximation of a function by the leading terms in its polynomial expansion; in this case each trip through the operation block computes another term in the polynomial by means of the term's index and adds it to the previous sum of terms.

The substitution block must also re-file information which depends upon the index. This can readily be demonstrated in examples where computing the n th value requires use of the previous value ($n-1$ st), thus computing the $n+1$ st value requires the use of the n th. Since numbers are specified by storage location, the n th value must be substituted for the $n-1$ st in order to carry out the computation.

This is a fairly complete description of the substitution block and, as well, extends the knowledge of internal requirements of the setup block. The substitution block closes the "loop" and makes the iterative calculation progressive.

e. Completion Block

If the alternative block sends the calculational process on to the next stage, the last of the blocks (γ) is

reached. This is the "completion block" and its function is the disposition of the result, or results, of the sequence just outlined. The disposition of results may involve having a table of results printed out, if the loop preceding this completion block was the main computation loop. It may similarly involve filing the results into sequential storage locations for later printing or for use in some other part of the computation if this sequence (in this case, subsequence) is but a portion of the main computation. After completion of the operations in this block, the calculation proceeds to the next stage, denoted by (β) .

f. Remote Connectors

(α) and (β) are called "remote connectors". Their function, although perhaps not so easily visualized in some of their forms, is to connect the subsequence into the main computational sequence at any and all desired points. In some computers (α) and (β) will be physical connections, but in many of the newer ones (α) will represent the location of the first order in this subsequence and (β) will be the location of the first of a group of orders to be carried out after the completion of the subsequence. An example of this might be a sequence composed of N orders stored in locations $A_1, A_2 \dots A_{\beta-1}, A_{\beta}, \dots A_N$. The computing machine carries out these orders until it reaches the order in location $A_{\beta-1}$; at this point the calculation has

progressed to the point where a subsequence is to be used, e.g. a square root, so the order in location $A_{\beta-1}$ contains the information that the machine is not to carry out the order in A_{β} next, but is to carry out the order in location C_1 ($C_1 = \alpha$), which is the first order in the square root subsequence (C_1, C_2, \dots, C_M). Since the computing machine follows a sequence of orders with no "memory" of its previous history, some means must be provided for putting the information into the order in location C_M (the end of the square root sequence) that the following order to be carried out is in the location A_{β} ($A_{\beta} = \beta$), thus permitting the original calculation to be carried out. The order which puts this information into C_M must be read prior to $A_{\beta-1}$. There is one additional complication, and this is that the location of the number whose square root is desired must be made available to the square root sequence, for, as presented so far, this number is available to only the A sequence.

The foregoing material has presented the flow of calculation as it is carried out by a human or mechanical computer. The system presented is obviously very flexible, and the ease with which this flow can be directed determines the flexibility of a computing machine. In a number of machines, such as the ENIAC as originally planned and built and in most of the I.B.M. computers, the flow was directed by means of physical connections, and the change of problems required rewiring the connections between computer elements. In most later machines the means of supplying orders to the

machines have been of such a nature that the need for physical interconnections of the former type is eliminated. This method of supplying orders consists of having a numerical code which stipulates the desired operation, and storing these numbers in some sort of memory medium. The machine proceeds to "read" these numbers stored in its memory and carries out their instructions. Since the memory is not destroyed by reading, and since the machine has access to all parts of this memory, it is possible for the machine to follow the same set of orders over and over again, providing iteration without having to supply a new set of orders for each repetition. With this system of order supply, the flexibility of a computing machine is governed mainly by the nature of the orders it is capable of carrying out. The other limitations are the speed with which the machine can carry out a computation, the available memory space, the accuracy with which the individual computations may be carried out, and freedom from mechanical failure. There is one other point to be considered, however, and this is the ease of preparing a problem for solution within the machine; this point will be considered in greater detail in a number of places in the following chapters.

3. Flow of Information

So far this chapter has been concerned with the flow of calculation, while the flow of information has not been considered to any appreciable extent. In order to

investigate the flow of information, some study of the forms in which information is utilized is indicated. Information with regard to numbers alone exists in two forms: (1) the size of the number, and (2) location of the number in the memory medium. Once the location of the number is known, the rest of the information is available upon reading the number at this location. Information with regard to orders is much the same, although it will become necessary to expand upon the information that may be stored in a numerically coded order. Most orders generally consist of requiring a particular operation to be performed upon a number (numbers) stored in a specified storage location (locations). The requirements of the flow of information of this type can be satisfied in either of two ways: (1) placing the desired number into the stipulated storage location prior to its use, or (2) controlling the storage location specified, in such a manner that it is at all times the proper one. Both of these methods of information flow control are used in most of the computing machines, and the choice between the two is generally determined by the number of operations required to carry them out. An example of this can be found in the square root subsequence; in this subsequence several different operations are performed upon the number (y) whose square root is desired, and it is simpler to move (y) into a storage location which can remain assigned to the square root sequence than it is to change all of the locations specified in the various orders.

There is one more type of information flow to be considered here, and that is the flow of control information, i.e., the correct sequence of orders. In the examples, the subsequences have terminated in a remote connector (β), where (β) is the storage location of the order following the order which shunted the calculational process into the subsequence. Since this subsequence should be available to other parts of the computation, the location (β) will differ depending upon which part of the computation required the subsequence at this time; hence (β) will have to be provided before entering the subsequence in order to guarantee proper reentry into the computation. This type of information flow control is similar to the second numerical type, i.e., proper control of the storage location specified.

4. Order System

The remainder of this chapter will be concerned with the properties of a system of orders capable of carrying out this generalized computational scheme. The system of orders described will be those which are to be employed in the computing machine presented in later chapters, but a few references will be made to systems which are in use in other machines. In particular, mention should be made of the possibility of breaking these orders into much more elementary operations which can be combined to correspond to those described below; this is done in a large number of the new machines and its merit depends upon the machine design.

a. The Arithmetic Orders

The first set of orders to be described are the arithmetic orders: Addition, Subtraction, Multiplication and Division. Each of these four orders supply four items of information: (1,2) the storage locations of the two operands, (3) the specified operation, and (4) the storage location in which the result is to be stored.

b. The Information Flow Order

The order controlling the internal flow of information is described next; this is the "Extract" order. The Extract order has the property of being able to extract any part of a word in one storage location and any part of a word in another location, combine these two parts to form a new word which is to be stored in a third storage location. It is also important to mention that the information in the two original storage locations is not affected by this procedure and that the previous information in the third storage location is completely annihilated and replaced. This extract order is, then, obviously capable of transferring a number into any desired storage location or to substitute new storage location specifications into other orders. In addition to satisfying these requirements of information flow, the extract order is capable of performing far more exotic operations such as providing one number with the algebraic sign of another, removing a specified group of digits from a number for separate arithmetical operations,

e.g., arithmetical operations to be performed on exponents, and a host of other possibilities.

The Extract order supplies seven items of information: (1,2) the storage locations of two operands, (3) the statement that this is an Extract order, (4) the position of the first of the group of digits in the first operand which are not to be used in making the new word, (5) the number of digits which are not to be used, (6) the first of the group of digits in the second operand which will be used instead, and (7) the storage location in which this newly formed word will be stored.

c. Flow of Calculation Order

There are two orders controlling the flow of calculation: Transfer and Comparison. These two orders control which chain of orders will be carried out in the immediate future. In the EDVAC the Transfer order occurs as a part of every order, that is, every order stipulates the location of the order that is to follow. In many machines, and in particular in the machine proposed in this thesis, orders are read from sequential storage locations. Since it is desirable to make a single subsequence available to all parts of the main computational sequence where needed, some means must be available to transfer from the sequence to a subsequence and back again, the Transfer order will generally be used to carry out these operations. In an iterative process it is also desirable to be able to use the same

set of orders over and over again until completed, some means must also be provided for this, with the additional provision that this order has contained within itself the ability to determine the necessity of reiteration, such an order is the Comparison order (sometimes called a Conditional Transfer order).

The Transfer order contains two items of information: (1) the statement that this is a Transfer order, and (2) the storage location in which the following order is to be found.

The Comparison order is a combination of the Subtraction and Transfer orders, and contains four items of information: (1,2) the storage locations of two operands, (3) the statement that this is a Comparison order (subtraction specified), and (4) the storage location in which the following order is to be found if the result of the subtraction is zero or negative. Since this machine reads orders from sequential storage locations, a positive result exerts no control and the machine continues reading orders in the original sequence.

d. The Communication Orders

The orders that allow communication between the machine and the exterior world are the "Read-in" and "Read-out" orders. These orders can be made very general, although in the machine proposed, the present plans are, at least for the initial construction, to restrict them to a

minimum. They can, however, be expanded without alteration of the original construction to take on a broader scope. The general formulations, and then the restricted versions will be presented.

The Read-in order can stipulate: (1) read in from a particular source of information (several choices) (2) beginning from a stipulated position in this source and storing this information sequentially into storage locations beginning with (3) the first storage location specified and continue filling up to (4) the final designated storage location, and (5) the location of the order which follows the Read-in order.

The initial plans for the proposed machine are to have a single input mechanism and to arrange material within it so that the desired information is supplied in the proper sequence.

The Read-out order has the same general and restricted formulation as does the Read-in order.

e. Halt Computation Order

The final order required to complete the order system is the Halt order. This stops the operation of the machine. This order has two uses in general, to stop the machine after the computation has been completed, or to stop the machine at any desired point in the computation in order to study the results available to check on the calculation or to make decisions on computational procedures.

A total of ten orders has permitted the computing machine to carry out the general computational scheme formulated earlier in this chapter. They provide for the arithmetic operations, the control of the flow of information, the control of the flow of calculation (including the determination of the necessity of reiteration), communication between the computing machine and the exterior world, and ability to halt the computation at any desired point.

f. Storage of Orders

Up to this point it has been implicitly assumed that some sort of memory system for storing words (orders and/or numbers) has been incorporated, and that the desired word may be obtained by specifying its location within the memory system. No mention has been made of requiring separate memory systems for orders and numbers, and, indeed, separate memory systems are not required. It is common practice to store numbers and orders interchangeably throughout the memory system, not even reserving some section of the memory for orders and the rest for numbers, except that sequences of orders must be placed in sequential storage locations. The advantage of this system of storing "words" is found in the maximum utilization of storage capacity, for if a sequence of orders have been used and are no longer required, this space may be used for the storage of intermediate results, new data or new orders, whichever is most desirable.

Perhaps the first reaction to the intermingling of numbers and orders is to raise the question: how can the machine tell the difference between a number and an order? The answer is: it cannot and does not need to, for in programming a problem the machine is furnished a connected calculational path and the "order reader" reads only the words which form the steps of this path, thus coming into contact with orders alone.

II. THE PROPOSED LOGICAL DESIGN

1. Introduction

The objective in the design of this computing machine is simplicity, simplicity of construction and of problem preparation. In achieving this objective the requirements of reasonably high speed of operation and general applicability must necessarily be satisfied, if an instrument of considerable utility is to result.

In achieving general applicability and simplicity of problem programming, great care was exercised in choosing a set of orders which had extreme flexibility and yet were very few in number with easily and well defined functional properties. The set of orders chosen is dependent upon the design of the machine itself, in particular, it is dependent upon whether the arithmetic unit is self-programmed or not. The chosen set of orders described in the first chapter are for a self-programmed arithmetic unit and are, consequently, a much simpler set than would have otherwise been required.

In the formulation of this logical design, two types of interrelations must be considered: (1) the interrelation of simplicity of construction and simplicity of problem programming, i.e., a reduction of complexity of the machine should not shift a corresponding burden upon the programmers; and (2) the interrelation between operating components making up the machine. These two will be considered in greater detail in the arguments to follow.

The criteria to be satisfied by the design are: a minimum of parts are to be used and all component's capabilities are to be fully exploited. The arguments leading to the chosen design will not be presented in detail, but sufficient information to indicate the line of reasoning will be included.

Parallel operations on numbers in a machine which uses a cyclic memory is wasteful of arithmetic equipment, for the average time required for locating numbers in the memory system is long compared to the time required for performing the arithmetic operations.* This does not allow efficient utilization of the arithmetic unit's capabilities.

Similarly, use of a static or rapid cycle memory in a machine which performs serial operations is wasteful of memory equipment, for the memory supplies numbers more rapidly than they are treated. Use of a longer cycle memory would offer a considerable increase in memory capacity without appreciably affecting the total operation time.

It is evident that to attain a reasonable balance in operating times, two designs are possible: (1) parallel operation with a static memory system or (2) serial operation with a cyclic memory in which the time required for the arithmetic operations is approximately equal to the memory cycle time.

* Refer to Appendix III for the properties of the components which provide for these operations.

In order to utilize a minimum of parts, the chosen design will be the second, for both the serial arithmetic unit and the cyclic memory system require a considerably smaller amount of equipment than do the others. However, this combination is much slower than the first, and represents an intermediate speed rather than a high speed computer. It should be pointed out that the actual speed of the proposed machine is dependent on the memory medium utilized, and that the use of a memory medium of the same nature as that employed in several "high speed" computers will yield approximately equal operating speeds.

2. The Magnetic Drum

In order to expand upon the logical design, and to go into the details of the requirements of the associated components, it is advisable to choose a physical model which can be described with relative ease. The actual model to be chosen will be the magnetic drum; however, it should be borne in mind that the magnetic drum is but a vehicle for exposition, and that the context and diagrams may be readily translated into any cyclic memory system.

The magnetic drum itself is perhaps a sufficiently new device that some explanation of its operation should be included. Physically, it consists of a cylindrical drum which is rotated about its axis. Mounted on a fixed framework about the drum are reading and writing heads, each head reads and/or writes onto a peripheral track about the

drum as it turns by. A large amount of information may be stored in any peripheral track and a large number of such tracks may be spaced along the length of the drum.

The surface of this cylindrical drum is sprayed with a thin coating of magnetic iron oxide, which can record a pulse of magnetic polarization in either of two opposing directions. In practice, the entire surface is magnetized uniformly in one direction to provide a "zero" background, and the presence of a "one" is indicated by a small region of opposite polarity. A reading and/or writing head consists of a magnetic core with one or more windings having an air gap mounted in very close proximity to the surface of the drum (see Figure 5). To write a number, a pulse of current is sent through a winding at the desired instant, producing a momentary high magnetizing force at the air gap, and hence magnetizing a tiny region of the surface. This region is called a cell. These pulses are in one direction for a "one"; the opposite for a "zero". Since the momentary magnetizing forces are sufficiently high to override the previous condition of the cell area, it is never necessary to erase as such, for the action of writing a new number automatically replaces any old information with the new. In reading, the sequence of cells (or digits) passing the air gap induces voltage pulses in the windings, which are amplified, shaped and supplied to the other components of the computing machine.

3. The Use of the Drum

In the proposed computing machine the drum is divided longitudinally into four sections: (1) timing tracks, (2) standard memory, (3) short memory, and (4) arithmetic tracks (see Figure 4).

In the timing section several (four, as will be shown) tracks are used to provide timing pulses for synchronization of the operating circuits.

The standard memory section corresponds to the memory system of the general computing machine described in Chapter I and Appendix III, and will be used to store words (numbers and/or orders) for the computing machine. The standard memory consists of a number of parallel tracks around the drum. In each track several words can be stored, these words are stored in serial form, for the arithmetic unit is of the serial type and must receive numbers in this form. A given word stored in this memory will pass one of the reading heads at a certain time during each cycle of the drum so the word may be "read" by specifying which track this word is in and at which angular position about the drum.

A rather unique feature is obtained by wiring a pair of these reading and writing heads in a feedback circuit (see Figure 4). If the reading head furnishes its information to the writing head which leads it by an angle θ , information once supplied to the pair will be repeated

over and over with a period $\theta/2\pi$ times the period of the drum. Such a circuit is called a "recirculator". This recirculator becomes in effect a short memory where a limited amount of information may be stored to be continuously available. The actual function of the short memory will be to store numbers between the standard memory and the arithmetic unit, for if a number is read out of the standard memory before the arithmetic unit can accept it, this number must be recirculated until it is accepted. A consequence of this function of short memory is, that the angle θ between reading and writing heads should be equal to the angle subtended by a word in standard memory, so that a recirculating word will always be in phase with the word positions in standard memory. (The requirement of a short memory is unique to the proposed computing machine.)

The arithmetic section of the drum will also consist of recirculators, these recirculators will be used for storage of the operands and the partial results during the arithmetic operation, providing a relatively inexpensive storage medium for this purpose. The rest of the logical design must be known before the requirements of this arithmetic section can be described satisfactorily.

In operation, the computing machine will go through four basic steps: (1) locate the desired order in standard memory and interpret this order, (2) locate the two operands in standard memory which are called for in this order, (3)

perform the specified operation on these two operands, and (4) file the result of this operation into the stipulated storage location in standard memory. It should be noted that three words are located and read from standard memory and that one word is being written back into it.

4. First Design Feature

Steps 1, 2 and 4 of the above operation may require one full revolution of the drum, for the word can be obtained only when the specified location in standard memory is passing the reading and writing heads.

The first design feature will be to allow one full revolution of the drum for carrying out each of the four steps outlined. This guarantees that steps 1, 2 and 4 will be completed during their allotted times, but demands also that the time required for carrying out the most difficult arithmetic operation (step 3) be less than or equal to the revolution time of the drum.

The operation now consists of four revolutions of the drum, each revolution guaranteeing the successful completion of one of the basic steps. These four steps may now be carried out simultaneously (see Table 1): (1) order N is being located, (2) the two operands called for in order N-1 are being located, (3) the operation specified in order N-2 is being carried out in the arithmetic unit, and (4) the result of order N-3 is being filed away into the stipulated storage location. There are, then, four simultaneous steps

being carried out, each referring to a different order or operation, so that the result is an accomplishment of a complete operation every revolution of the drum.

5. Component Requirements

Returning to the standard and short memory sections of the drum, it may be seen that during a given revolution four words are communicated between the standard memory and the rest of the computing machine. An order and two numbers (or operands) are read out of standard memory, the order being destined for an "order interpreter" (see Figure 6) which will translate the serial digital information into static control voltages for the rest of the machine, and the two numbers destined for the arithmetic unit to be operated upon. The fourth word communicated is a number, the result of an arithmetic operation, which is written into standard memory.

The order and two operands read out of standard memory are located some time during the memory cycle and must be stored until the end of this cycle, at which time they are delivered to their respective destinations. The "result" number is delivered from the arithmetic unit at the beginning of the memory cycle and must be stored until it can be written into the proper position in standard memory. The function of the short memory section is to provide the storage for these four words, and it is obvious that four recirculators are required for this purpose.

It might also be noted at this point that some sort of unit will be required to "connect" the desired location in standard memory to the short memory recirculator which is (or is to be) temporarily storing the information (see Figure 6). The properties of such a unit are: it must be capable of arranging for connection of the recirculator to the desired track in standard memory, and to close this connection during the proper part of the memory cycle to pick out only the one desired location. Such a unit is called a "time and track selector", and four of these units are required if it is desired to be able to connect to storage locations which are passing the reading and writing heads at the same time, one being required for each recirculator. The information of which track and time (angular position) to contact is supplied by the "order interpreter", for this information is contained in the orders.

6. Second Design Feature

The requirement of the first design feature was that the time for the longest arithmetical operation be less than or equal to the period of rotation of the drum (or memory period). The second design feature is concerned with maximizing the efficiency of the arithmetic unit under such restrictions. The efficiency is greatest when the operation time is precisely equal to the period of the cyclic memory, and the achievement of this constitutes the second design feature. The combination of the two design features can

best be termed "INTEGRAL SYNCHRONIZATION", for all units of the computing machine form an integrated system, carrying out their functions simultaneously and in identical periods of time.

This second design feature immediately imposes difficulties in efficient utilization of the operation time, for with "fixed point" number representation the various arithmetical operations are performed in different lengths of time; multiplication and division take about n times longer for n -digit numbers than addition and subtraction. In addition to this, of all number representations (with respect to base) only the binary system permits division to be performed as rapidly as multiplication may be performed; and then only under rather stringent conditions, i.e., the dividend must be less than twice the divisor.

These are not essential difficulties, however, for they are removed by the introduction of two numerical features, both of which are desirable from the point of view of simplicity of construction or simplicity of problem preparation. These features are: (1) binary representation of numbers and (2) a floating binary point. The binary representation allows multiplication and division to be performed in the same length of time, provided the conditions for division are satisfied. The floating binary point makes it possible to satisfy at all times the condition on the division process, thus speeding it up. The floating

point also makes the operations of addition and subtraction require the same length of time as multiplication and division, but accomplishes this by reducing the rate of performance of the first two. The reduction in speed of addition and subtraction should not really be considered a loss however, for the time is available and the additional operations are certainly of a useful nature.

The introduction of the floating-point feature does require some additional equipment for addition and subtraction of exponents and for control of shifting numbers, but the increase is not very large relative to the amount of equipment already involved. In addition greater storage capacity is required to store an equal quantity of numbers, for the values of the exponents must also be stored; however, a corresponding increase in useful information is made available.

To complete the requirements that must be satisfied before integral synchronization may be achieved, an analysis of the relationship between the memory and arithmetic units must be made. The first analysis will ignore the requirements of providing short periods of time between numbers, sufficient to perform switching operations.

Multiplication furnishes the simplest example and will be used to initiate the analysis. In multiplication of two n -digit numbers, each of the n digits of the multiplicand must be multiplied in turn by each of the n digits

of the multiplier, forming a total of n^2 partial products with their corresponding powers of two (binary system). The product is obtained in binary form by summing the n^2 partial products. In actuality the process of summation of the partial products is accomplished by summing each partial product as it is formed, thus requiring no additional time. The formation of partial products occurs as rapidly as digits are presented to the multiplication component, so n^2 digit-times are required for the formation of the product. It is desirable to have the memory system supply the n digits of each operand to the arithmetic unit at the same rate as the internal storage (arithmetic section of the drum) of the arithmetic unit supplies them to the multiplier, so that a speed matching device is not needed; hence, to maintain integral synchronization between the arithmetic and memory systems, the period of the memory system must also be equal to n^2 digit-times. The addition of the exponents, which must also be performed, is carried out in a special exponent register and is done simultaneously with the rest of the operation.

In the following chapter it will be shown that the division process will require precisely the same length of time, i.e., n^2 digit-times.

Addition and subtraction do not always require n^2 digit-times, the actual time varying from $2n$ to n^2 digit-times, but n^2 are required to assure completion, hence these

two operations will be considered to utilize the full capabilities of the arithmetic unit as well as multiplication and division.

The requirement of n^2 digit-times making up one period of the cyclic memory imposes relationships between the number of digits of information in a word (significant digits, exponent value and the algebraic signs of the number and exponent make up a word), the number of significant digits in a number, and the number of words which fit into one period of the memory. These relationships must be satisfied if maximum utilization of the memory and arithmetic units are to be realized. An additional condition, for maximum use of order information, might also be imposed: that the number of words in a memory period be equal to an integral power of two. Designating m as the number of digits in a word, n as the number of significant digits, and 2^j as the number of words in a memory period, the relationships are: (1) $n^2 = mx2^j$, (2) $m > n$, and (3) m, n, j are integers. The spectrum of m and n for a given value of j may be generated by a single integer k , where $k=1, 2, 3, \dots$ ($k=0$ generates $m=n=2^j$). There are two cases, j -odd and j -even:

j -odd

$$n = 2^j + k \times 2^{\frac{j+1}{2}} \quad k = 1, 2, 3, \dots$$

$$m = 2^j + k \times 2^{\frac{j+3}{2}} + 2k^2$$

j-even

$$n = 2^j + k \times 2^{j/2}$$

$$m = 2^j + k \times 2^{\frac{j+2}{2}} + k^2$$

The choice of j and k must be made in such a fashion that n and m are reasonable values, and the choices will be taken to be: $j = 5$, $k = 1$, yielding $2^j = 32$, $n = 40$ and $m = 50$. This represents 32 words in a memory period, 40 significant digits in a number and 50 digits to represent a word (number or order). Forty of the 50 digits in a number are the significant digits, 8 are used as the exponent value, and 2 are to be used for the algebraic signs of the number and the exponent. The numbers are to be represented as $\pm p \times 2^{\pm q}$ where $1 > p \geq 1/2$ and $255 \geq q \geq 0$. The 40 significant binary digits are approximately equivalent to 12 decimal digits, and the $2^{\pm 255}$ range in number size is approximately equivalent to $10^{\pm 75}$.

It was mentioned earlier that provision of some "dead space" for performing switching operations is required. This switching time must be provided between words in the memory system and between significant digit groups in the arithmetic unit. Let the switching time provided in the arithmetic unit be equal to α digit-times, and that between words in the memory unit be equal to β digit-times, where α and β are integers. The set of altered relationships are: (1) $n(n + \alpha) = 2^j(m + \beta)$, (2) $m > n$, and

(3) n, m, j, α, β are integers. The spectrum of m and n are yielded by the original set of equations, and an additional set of equations for generating the spectrum of α and β by a second integer, h , is provided:

j -odd

$$\alpha = h \times 2^{\frac{j-1}{2}}$$

$$\beta = h \times (2^{\frac{j-1}{2}} + k)$$

$$h = 1, 2, 3, \dots$$

j -even

$$\alpha = h \times 2^{j/2}$$

$$\beta = h \times (2^{j/2} + k)$$

The above equations do not yield the complete spectrum for even k , but the complete spectrum may be obtained by choosing a new factor $h' = 2^g \times h$, where g equals the number of factors of two in k , not exceeding $\frac{j-1}{2}$ for odd j or $\frac{j}{2}$ for even j . These cases are not of great interest, for the numbers yielded do not represent cases of practical interest.

h is again chosen to be 1, yielding $\alpha = 4$ and $\beta = 5$. Actually the choice of h is determined by the operation time of the switches, and α and β are very reasonable values. In addition to this $\beta - \alpha = 1$ could be used as additional storage space in the memory if switching required about the same length of time in either place, but the switching operation is more complex in the memory system than in the arithmetic unit so the additional time is useful.

The result of the choices yield an integrally synchronized memory system and arithmetic unit with an operating period of $40 \times 44 = 32 \times 55 = 1,760$ digit-times, so the magnetic drum must have space to store exactly 1,760 digits in each peripheral track. The two digit-time groupings, 44 and 55, will constitute a "minor arithmetic cycle" and a "minor memory cycle" respectively; and the total of 1,760 digit-times will constitute a "major memory cycle".

7. Component Requirements

The component requirements of "standard memory" and "short memory" were fairly well established from the first design feature, except that the number of words in a particular track in standard memory was not known, but has just been chosen to be equal to 32. This fixes the requirements of the spacing angle between the reading and writing heads of the short memory recirculators, for they must subtend the same angle as a word in standard memory, or $2\pi/32$ radians (55 digit-spaces). The requirements of the other units previously described ("order interpreter" and "time and track selectors") remain as stated, but will be enlarged upon in the following chapter.

The second design feature fixes the requirements of the arithmetic and timing sections of the magnetic drum. The recirculators in the arithmetic unit must span an angle about the drum equal to that of 44 digit-spaces, for these recirculators must store the 40 significant digits of a

number and also the 4 digits of "dead space" for switching operations must be included. Three such recirculators are required: two for the two operands which are read into the arithmetic unit at the beginning of the major memory cycle, and one for storage of the results at intermediate points in the arithmetic process (such as the sum of partial products at any point in the multiplication, the quotient digits generated at any point in division, the sum or difference in addition and subtraction).

Since the arithmetic section of the drum stores only the significant digits of the numbers, the complete arithmetic unit must provide additional storage space and operational equipment for storing and adding the exponents. These exponents must also be used to provide control of the relative shift between addend and augend or minuend and subtrahend (this will be discussed in the next chapter).

Although it was just stated that the spacing of the recirculators in the arithmetic section should be 44 digit-spaces, actually a control of this spacing must be exerted, allowing values of 43-44-45 digit-spaces, for shifting the numbers to the right, none at all, or to the left of the binary point, respectively, one digit per minor arithmetic cycle.

The timing section of the drum is provided to synchronize the three other sections, supplying "markers" indicating the stage of the various cycles to the

corresponding units concerned. These markers must be of four kinds: (1) digit-time markers, which are required by all units to synchronize inter-storage communication; (2) minor memory cycle markers, which synchronize the standard and short memory systems and also the "time and track" selecting switches; (3) minor arithmetic cycle markers, which synchronize the operation of all the elements in the arithmetic unit; and (4) a major memory cycle marker, which signals the beginning of each operational step, including transfer of information between the arithmetic and short memory sections, transfer of newly located orders from the short memory unit to the "order interpreter", and a number of other applications.

Although not properly introduced in this section, the "input" and "output" units must establish communication with standard memory, and will do so through two of the four short memory recirculators which are provided. To establish communication these units must also be properly synchronized with the short memory system, and will require digit, minor memory, and major memory markers from the timing section. These functions will be considered in the next chapter.

8. Modifications

a. Input and Output

It was stated in the first chapter that the two communication orders (Read-in and Read-out) have possibilities

for expansion to include several input and output units. These additional units would be of use in supplying or producing tables of functions for the machine. They can be added without appreciable change in the original construction.

b. Storage Capacity

So far no statements have been made about the storage capacity available in standard memory, except that each "track" in the system can store 32 words and that a number of tracks may be used in parallel. The maximum number of tracks, N , is dependent on the order system, and $N=32$ is a maximum for the order system previously defined, since the number of digits available in an order for specifying addresses (storage locations) is limited. A change in the Extract order, involving the replacement of one of the original operands by the new word constructed, would allow $N=512$ as a maximum.

c. Time and Track Selectors

In the component requirement subsection of the first design feature it was stated that four time and track selectors were required if it were desired to connect to storage locations which passed the reading and writing heads at the same time. An alternative to this would be to have two time and track selectors, one for reading and one for writing into standard memory. The single reading time and track selector could function for all three of the reading

operations by providing an additional switch for connecting to any of the three short memory recirculators. The programmer would have to prepare the computations so that none of the three words being read from standard memory would appear during the same minor memory cycle. This would place a certain additional responsibility on the programmer, but the flexibility of the machine would not be appreciably impaired.

d. Special Storage Locations

Before concluding this chapter it is of interest to point out a difficulty which is peculiar to the simultaneous performance of operational steps (refer to Table 1). If order N calls for either or both of its operands to be the result of the N-2nd order or the N-1st order, a dilemma exists, for these results have not yet been filed into standard memory, so are not available during the corresponding "locate operands" cycle. It will be noted, however, that the result of the N-2nd order is available during this time in the "result" short memory recirculator*, and will remain there until replaced during the first minor memory cycle of the next major memory cycle (the word not being destroyed when written into standard memory), which is the operation cycle for order N. Thus it is possible to just read this

* The availability of the result in short memory was first recognized by C. H. Davidson, University of Wisconsin.

result from its short memory recirculator into the desired arithmetic section recirculator during that first minor memory cycle.

Similarly the result of the N-1st order is being generated in the arithmetic unit during the "locate operand" cycle of the Nth order, so it will be available to read back into the desired arithmetic section recirculator for the following operation cycle of order N.

The introduction of these two special "memory addresses" into the order system allows simpler programming procedures and more efficient use of the computing machine.

III. CHOICE OF COMPONENTS AND THEIR OPERATIONAL REQUIREMENTS

1. Introduction

In the design of the first model of the "integrally synchronized computer", which is to be built at the University of Wisconsin*, it has been decided to incorporate components which would require a minimum of developmental research. The memory unit which appeared to most nearly satisfy the requirement was the magnetic drum, particularly because of the ease of synchronization between the three memory sections. Another advantage is that commercially available magnetic drums have digit-periodicities of very nearly the value required, i.e., their maximum storage capacity per "track" is about 15% in excess of the required 1,760 digits.

The ease of synchronization when utilizing the magnetic drum is a property of the system itself, for the complete memory system consists of a uniformly moving medium, and timing markers placed on this make the medium serve as its own "clock" independent of its speed (within the limits of the marker-sensing equipment utilized).

There is an additional advantage and a disadvantage in using a magnetic drum storage system. The advantage

* This model has been named "WISC", Wisconsin Integrally Synchronized Computer.

is that the information stored on the magnetic drum is non-volatile, allowing the computing machine to be shut down overnight even though the problem is unfinished; or, if a check shows an error has developed in the computer, it may be repaired and the problem may frequently be started from the point of the previous check rather than starting at the beginning. The disadvantage of the magnetic drum is that it is essentially a low speed device, presenting digits at a frequency of 100 kilocycles in comparison to the 1 megacycle frequency achieved in mercury delay lines. However, in a way this is advantageous for building the first model, for the development of circuits for handling the lower frequencies is considerably easier.

2. Division of Drum

The division of the drum into sections according to function is shown in Figure 5. The five sections are: (a) Timing, (b) Standard Memory, (c) Short Memory, (d) Arithmetic Memory, and (e) Input and Output. With the exception of section (e) these sections are identical to those required in Chapter II. The use of the Input and Output section will be discussed in the latter part of this chapter.

a. Timing

The Timing section will contain four reading heads, which will read the timing markers for digit-times (1760 per

revolution), minor arithmetic cycles (40 per revolution), minor memory cycles (32 per revolution), and major memory cycles (1 per revolution).

To maintain precise synchronization throughout, without the use of precision equipment, all digits will be written onto the drum in coincidence with digit timing-markers; similarly the reading of digits from the drum will be done in coincidence with these timing-markers, but delayed an amount equal to the inherent delay of the system.

b. Standard Memory

The Standard Memory section will contain 32 combination reading and writing heads, each serving to read or write information in a particular track around the drum. In each track there are spaces for storing 32 words, so the total storage provided totals 1024 words. Connected to these 32 reading stations are three parallel track selection units, each of which can connect an amplifier to any of the 32 tracks. The outputs of these three amplifiers are the three words: (1) the next order, (2) and (3) the two operands called for in the present order. Connected to the 32 writing stations is a single track selector which connects the "writing amplifier" to the desired track for recording new information on the drum. In general the new information will be the result of an arithmetical or extract operation, but the words supplied by the input mechanism will be recorded on the drum through this same channel.

Similarly, the output of one of the three track selectors connected to the 32 reading stations will be supplied to the output mechanism for permanent external recording.

c. Short Memory

The Short Memory section consists of four tracks with reading and writing heads separated by 53 digit spaces (Refer to Figure 4). Three tracks store the new order and the two operands of the present order from the time of receipt from Standard Memory until they are required at their respective destinations. The fourth track stores the result of an operation from the time of receipt from the Arithmetic Unit until erased by the receipt of a new result one major cycle later, the result being available all during this cycle for duplication in the desired location in the Standard Memory. The new order will be transferred from the Short Memory storage track into the Order Interpreter (see Figure 6) during the last minor memory cycle of the major cycle in which it is read from Standard Memory. In the event that this new order is not located until this last minor memory cycle (at which time success of location is assured) the new order is read directly from Standard Memory into the Order Interpreter without requiring the intermediate storage. The two operands will be transferred from Short Memory into the Arithmetic Unit during the first minor memory cycle of the following major cycle; simultaneously, the result just obtained in the Arithmetic Unit is

transferred into the fourth Short Memory track for filing into the required Standard Memory position sometime during this major cycle.

d. Arithmetic Memory

The Arithmetic Memory section consists of three tracks with reading and writing heads separated by 42 digit spaces. Two of these tracks will receive the two operands from Short Memory and the third will be used to store the result of the operation until delivered to Short Memory at the beginning of the next major cycle.

e. Input and Output

The Input and Output section will consist of four tracks with combination reading and writing heads. Two of the tracks will be used to store information read from the Input mechanism, but which Standard Memory is not ready to receive. The other two tracks store information delivered by the Standard Memory, but which the Output mechanism is not ready to receive.

Since each track holds 32 words, a total of 64 words of intermediate storage is supplied for each purpose.

3. Supplementary Features

a. Recirculators

The magnetic drum is provided with a single inherent period, which is the time for one revolution, but two additional periods are required, short memory requiring

55 digit-times and arithmetic memory requiring 44. These additional periods can easily be produced by the feedback circuit in Figure 4. In such a circuit the information written onto the surface of the drum at the writing-head location is carried past the reading-head location a number of digit-times later. This information is now fed back to the reading-head, amplified, and supplied to external circuits and back to the writing-head again. In this fashion a number of digits, not exceeding the digit-time delay between reading and writing heads, may be made available periodically. Such a system is termed a "recirculator", and recirculators having delay times of 53 and 42 digit-times will be used for the short and arithmetic memories, respectively. (The 53 and 42 digit-times rather than 55 and 44 will be explained when inherent and controlled delays are taken into account).

b. Delays

Before proceeding any farther, the question of inherent and controlled delays must be considered, for they are necessary to the operation of the Arithmetic Unit and to the synchronization of the machine. In the process of "writing" on the magnetic drum a pulse of current is used to set up an intense magnetic field at a spot on the surface of the drum (see Figure 5), permanently magnetizing it; this current pulse is triggered by a digit marker, so the current pulse reaches its maximum a short time later than

the digit marker. Later, when read back, an additional inherent delay is to be found in the amplifier. As a result of this delay it is difficult to use the output of the amplifier to directly recontrol the writing process, for the inherent delay is less than one digit-time. The simplest means for solving this problem is to increase the delay of the amplifier so that the total inherent delay becomes exactly equal to one digit-time; any additional delay that may also be desired is to be added in increments of one digit-time. The obvious difficulty that arises at this point is that the digit read from the drum is one digit-time later than when written, which still makes it impossible to control writing processes. The way out of this apparent dilemma is to run portions of the computing machine one digit-time behind the rest, with means for regaining the lost time in communication between these sections. The system of "time stabilization" proposed here is comparatively simple, and is shown in Figure 7. It will be noted that the regaining of the digit time is accomplished by means of recirculators.

c. Shifting Registers

In concluding the discussion of delay, it might simplify later discussion to state that the one-digit delay units which will most probably be used are special "flip-flop" units (units having two stable states of conduction) which take on the state of conduction determined by two

input trigger pulses. One of these pulses is called a reset pulse, and the receipt of this pulse causes the "flip-flop" unit to take on the state of conduction designated as "zero". If the unit was in the "zero" state prior to the application of the reset pulse, no change occurs, but if it was in the "one" state, a change in the state of conduction occurs. The change in state of conduction may be used to generate a trigger pulse similar to the reset pulse but later in time. The "change of state" trigger pulse is used as the second input trigger pulse to an adjacent "flip-flop" unit and causes this unit to take on the "one state". Considering a chain of such "flip-flop" units having a given configuration of conduction states, the simultaneous application of reset pulses to all units causes each unit to take on the "zero" state. Those units which were previously in the "one" state generate "change of state" pulses which are applied to the following unit in each case, causing the following unit to take on the "one" state. In effect, then, the configuration of conduction states moves one unit down the chain each time the reset pulses are supplied. Such a chain is called a "shifting register", and converts a serial input of information into a static setting on the flip-flop units. In application in the Wisconsin machine, the present plans are to attain the full one-digit inherent delay in reading by having such a flip-flop unit as the output stage of the amplifier.

One more point to be considered about the "flip-flop" units is that the units are to be symmetrical, so that the "zero" state and the "one" state differ only in the respect that the potentials at two corresponding points in the circuit are reversed; this will be useful for obtaining complements of a number by "reading the opposite side" of the flip-flop units (the complement also requires the addition of 1 in the least significant position).

d. Assignment of Digits

In order to discuss the operation of the Order Interpreter the assignment of digits in the orders should be described. In Figure 8 a set of four diagrams are shown which picture in reasonable detail: (a) assignment of digits in a number, (b) general assignment of digits in an order, (c) detailed assignment of digits in an order, and (d) precedence of control information. There are a few items of information regarding the assignments which are of interest. In the numbers the arrangement of the digits are such that the least significant of the digits in both the numerical and exponent groups are at the left; and appear first in time throughout the computer; in addition all numbers which are to be handled in the Arithmetic Unit will have a 1 in the 40th digit (most significant) space, with the exception of numbers for special purposes (i.e., which are to be added or subtracted only). In the orders the 40th digit is retained as a 1 for all but the extract order, this is

provided so that addition of constants to the order addresses (locations in Standard Memory) for operations on sequences of numbers may be accomplished without modifying the Arithmetic Unit; similarly the exponent section is kept free in all but the extract order so that a single step is all that is needed for addition of constants to the order addresses. Keeping the exponent section free does require dual assignment of digits in several cases and the significance of digits in these positions depends on the presence of other digits, as shown in the last diagram in Figure 8. More compact grouping of the order digits could be attained by using binary notation, but additional circuitry would be required for this. The "Short Memory" digit assignments are for specifying either of the operands to be results of the previous or second previous operations; only one such Short Memory assignment is available in the extract order, but this is sufficient to allow consecutive extract operations on a given word.

Sufficient information is now available to cover the operational requirements of the rest of the computer units. These units will be taken in the order: (1) Order Interpreter, (2) Time and Track Selectors, (3) Arithmetic Unit (covering each operation required), and (4) Input and Output Units.

4. Order Interpreter

The Order Interpreter is shown in Figure 9. Its main components are "shifting registers" which receive the new order in serial form from the new order Short Memory track, or from Standard Memory if located in the last minor memory cycle. This information transfer takes place during this last minor memory cycle and remains as static information on the shifting register for at least one major memory cycle. It should be noted that the information is divided into 5 parallel 10-digit groups (A, B, C, D, E), and that the (C) group consists of three series units (C, C', C''); also the (D) and (E) groups consist of two series units (D, D') and (E, E'). The number of units in series is determined by the number of major cycles during which the information must be kept available. The input to the 5 parallel groups is governed by a cyclic switch, allowing 10 digits to be read into each.

a. The (A) and (B) Groups

The (A) and (B) units contain (for addition, subtraction, multiplication, division, comparison, and extract orders; A, S, M, D, C, and E) the locations in Standard Memory of the two operands called for in the order, and are required only during the following major memory cycle. In the case of the specification of a Short Memory address, any location may be specified in the (A) and (B) section (zero being the simplest), and the word in this location will be

located and placed in Short Memory, where its future will be governed by the Short Memory address specified in the order.

In the cases of the other four orders (Read-in, Read-out, Transfer, and Halt; I, O, T, and H) the (A) and (B) units contain information of a different nature. For I and O orders the (A) unit holds the address in Standard Memory into or out of which the first of a sequence of words will be transferred. This address will be increased by 1 each major memory cycle (except for a special case discussed in the Input and Output section), allowing the transfer of a new word every major cycle from Input to Standard Memory or Standard Memory to Output. The (B) unit, for I and O orders, will hold the address up to which this process will continue. For T and H orders any information may be placed in the (A) and (B) units, but the words in these addresses will be located and sent to the Arithmetic Unit where some uncontrolled operation will be performed, however, the result of this operation will not be allowed to be written into Standard Memory.

b. The (C) Group

The three (C) units serve two purposes. In the A, S, M, D, and E orders, the (C) unit holds the address in Standard Memory into which the result of the operation will be written. The input to the (C) section causes the information in (C) to be shifted to (C') and that of (C') to be

shifted to (Cⁿ); making the information in these units available for three major cycles. This is necessary, for the address into which the result of an operation is to be written cannot be used until this result is available for storage, which is during the third major memory cycle following the receipt of the order by the Order Interpreter. The second purpose served by the (C) unit is the storage of the address of the following order in the case of T, I, and O orders; the C order uses this position for storing the address of the following order under the condition that the result of a subtraction is zero or negative, but the information as to the result of the subtraction is not available until the end of two major memory cycles, so this must be kept available for transfer into the Order Selector section until the result is known. In the T order this new address must be transferred immediately to the Order Selector, and in I and O order a similar transfer must be made, but the new order cannot be read until the completion of the I and O operations, which may take a large number of major cycles.

Returning to the C order and the operation of the Order Selector section of the Order Interpreter, it was stated that the address of the following order was to be found in the (C) unit in case of a zero or negative result of a subtraction. The function of the Order Selector is to supply the addresses of the orders to the Order 'Time and

Track Selector', and it accomplishes this function by adding 1 to the previous order address every major memory cycle, thus providing orders from sequential storage locations; transfer from this sequence of orders is provided for by the C, T, I, and O orders and is accomplished by replacing the address in the Order Selector by the new address supplied in the order. Since the result of the subtraction in the C order is not known for two whole major memory cycles, the Order Selector will continue reading orders from the same sequence for the two following cycles; in the case of a positive result two of the desired operations have then already been started, and in the case of a zero or negative result the new order address is placed in the Order Selector and the results of the two undesired orders are not allowed to be written into Standard Memory (they become available for writing after the result of the comparison is known), and it is as though they had never been carried out. In addition to the above, the result of the subtraction in a Comparison order must not be written into Standard Memory.

c. The (D) and (E) Groups

The (D) and (E) units of the Order Interpreter shifting register contain the information of the order type. The control digits in these sections determine the type of information contained in the (A), (B), and (C) units as well as in the (D) and (E) units and are not generally needed until the operands for the order have been located (except

for T, I, O and H), which means they must be stored until the second major memory cycle during which they are used.

As shown in Figure 9, the statement that this is an Extract order (digit 40=0) causes a particular interpretation to all of the digits in the (D) and (E) sections. The statement that this is not an Extract order but is one of the arithmetic orders (A, S, M, D, C) (digit 40=1 and one of digits 31-35 are 1) causes a particular interpretation of the (T, I, O, H) group of digits (no interpretation is made of the (E) group in any but the Extract order, so they may be left blank or given values for special operations in the Arithmetic Unit). The statement that this is not an Extract or an Arithmetic order (digit 40=1 and digits 31-35 are zero), allows the T, I, O, and H group to represent these four orders. The properly interpreted digits supply control voltages wherever needed throughout the computer, and are used generally for the setting of electronic switches.

5. Time and Track Selectors

The four Time and Track Selectors are essentially alike, at least in function, and a representative functional diagram is shown in Figure 10. The Time and Track Selector consists of two main units, one of these units is a track selector, which in effect chooses the correct position of a 32-position switch. In actuality this is a binary to unitary translator, closing one of 32 possible contacts determined

by the binary number in a 5-digit register. The other unit is a coincidence detector, which completes the connection between the amplifier and the reading or writing station when the setting of a 5-digit binary "minor memory cycle" counter coincides with the 5-digit register setting which stipulates the angular position of the address. In practice the reading and writing stations will be numbered 0 through 31, and the minor memory cycles, or angular positions, will also be numbered 0 through 31. The two 5-digit registers will actually be one of the 10-digit units in the Order Interpreter shifting register, the specification of track and angular position each require 5 of the 10 digits and actually designate the physical position of the word in Standard Memory.

Some sort of "lock" will be used to maintain connection between the track and amplifier for a full minor memory cycle, for in the last minor memory cycle the information in the Order Interpreter shifting register will be annihilated by the introduction of a new order, but the Time and Track Selector must continue functioning throughout this minor memory cycle with the previous information. Sufficient time to set the switches is available before the old information is destroyed.

6. Arithmetic Unit

This section covers the operation of the Arithmetic Unit. Numbers (or orders) are read into this unit

during the first minor memory cycle of a given major memory cycle, two numbers are read in at this time and have been called operands No. 1 and No. 2. These operands are of the form $\pm A \cdot 2^{\pm a}$ and $\pm B \cdot 2^{\pm b}$, respectively (A, a, B, and b are positive), where the first 40 digits of operand No. 1 constitute A, the next 8 digits constituting a, and the last two digits giving the signs of a and A, respectively; operand No. 2, of course, comes in precisely the same order. As stated earlier in the assignment of digits, the earliest digits in A and a are the least significant digits. A and B are accepted in the Arithmetic Memory during the first minor arithmetic cycle, and the succeeding ten digits (a and signs and b and signs) are accepted by two "exponent shifting registers" for their operations.

It should be noted that the exponents and signs appear in the exponent shifting registers during the second minor arithmetic cycle (actually a 6 digit-time overlap occurs), so they are not available for control until the beginning of the third minor arithmetic cycle. This causes no difficulty in the case of multiplication and division, but in addition and subtraction considerable difficulty ensues, for decisions on shifting and on the actual arithmetic operations are required. This difficulty arises in the four cases in addition and subtraction which require the full operational time; these cases will be taken up later on.

To investigate the operation of the exponent shifting registers and the addition and subtraction cases, it is advisable to first take up the method of handling numbers in these operations. The methods of addition and subtraction in the exponent registers will be about the same as those used in the rest of the Arithmetic Unit.

a. Addition and Subtraction

There are four cases in addition and also in subtraction, depending on the algebraic signs involved. To handle these in the simplest way, one of the numbers (A) will be considered the "major signed number", and all results will first be obtained with the sign of A. Subtraction will be performed by adding the complement, with respect to 1, of the subtrahend. (Only bracketed numbers are recorded in the result track). See Figure 12.

The number, A or B, with the least positive exponent is to be shifted to the right of the binary point by an amount equal to the exponent difference or 39, whichever is the smaller.

Addition

$$(+)A + (+)B = +[A+B]$$

$$(+)A + (-)B = +[A + (1-B)] - 1$$

$$(-)A + (+)B = - [(+)A - (+)B] = -[A + (1-B)] + 1$$

$$(-)A + (-)B = - [(+)A - (-)B] = -[A+B]$$

Subtraction

$$(+)A - (+)B = +[A + (1-B)] - 1$$

$$(+)A - (-)B = +[A+B]$$

$$(-)A - (+)B = -[(+)A + (+)B] = -[A+B]$$

$$(-)A - (-)B = -[(+)A + (-)B] = -[A + (1-B)] + 1$$

It should be noted that the algebraic sign of the "major signed number" controls the indicated operation, that is, a negative sign associated with A converts an addition to a subtraction or vice versa; this is in addition to indicating the trial sign of the result. The "minor signed number" has control over its complement, that is, a negative B inverts the "complement decision" of the indicated operation (the indicated operation being the result of the originally specified operation controlled by the sign of A). It is interesting to note that only two controls are required in this method, and that the control voltages are available in the exponent shifting registers; negative A inverts the indicated operation, and negative B inverts the complement decision.

So far the possibility of negative results has been ignored, i.e., the sign of the result \neq the sign of A. This can occur under only one condition, namely, when the complement of B is added to A; this condition occurs four times in the eight addition and subtraction cases.

The condition on A and B (including shift) is:

$$1 > A, B \geq 0, \text{ where } 0 \text{ is considered to be } 2^{-41} \text{ smaller.}$$

Case 1: $A \geq B$

$$A + (1-B) = 1 + (A-B) = 1 + \epsilon$$

where: $\epsilon \geq 0$

so: $A + (1-B) \geq 1$

Case 2: $A < B$

$$A + (1-B) = 1 - (B-A) = 1 - \epsilon'$$

where: $\epsilon' > 0$

so: $A + (1-B) < 1$

Case 1 illustrates the condition that the sign of A is equal to the sign of the result, and Case 2 the condition that the sign of A is opposite to the sign of the result. Since the addition of the complement of B to A represents the operation of taking the difference of two positive numbers, the $|A-B|$ is less than the larger of the numbers (A, B), and hence is less than unity. This makes it permissible to inspect the digit-position of 1, and if a 1 occurs in this position (Case 1), the result is positive and the correct result is obtained by failing to write the 1 in the "result memory track" (failure to write 1 being equivalent to subtracting 1 from the result); if the digit in the 1 position is 0 (Case 2), the result is negative and the number recorded is the complement of the desired number, the correct result is now obtained by taking the complement of this recorded number and changing the sign in the exponent register. (The inspection of the digit-position of 1, which is the 41st digit-time of each minor arithmetic cycle, is required

for ascertaining signs only when the complement of B is added to A.)

The result of an arithmetic operation is written as $\pm R \cdot 2^{\pm r}$, where $1 > R \geq 1/2$ and $255 \geq r \geq 0$. In the addition and subtraction cases, where $1 > A, B \geq 0$ and the unshifted number (A or B) is greater than or equal to one-half, the condition $[A+B] = R$ satisfies $2 > R \geq 1/2$, implying the need for inspecting the digit-position of 1 again to determine the need of shifting the result (a required shift in this case requires the addition of 1 to the most positive exponent of A and B, either $\pm a$ or $\pm b$, to obtain $\pm r$); the condition $[A + (1-B)] = 1 \pm R$ yields an R which satisfies $1 > R > 0$, implying the need for inspecting digit-positions 2^{-1} , 2^{-2} , ..., 2^{-40} successively until a 1 is located or all are determined to be zero, each successive zero requires a shift and the subtraction of 1 from the most positive exponent in order to obtain $\pm r$ (in the event that all digits are zero, the present plan is to make $\pm r = -255$ and the sign of the result negative, so zero is carried as a negative number.)

b. Exponent Unit

The Exponent Unit is shown in Figure 11; the main components are two 10-digit shifting registers, an exponent adder with switching controls, and coincidence detector with shifting controls. The two 10-digit shifting registers serve two purposes. They receive the two exponents with the algebraic signs during the first minor memory cycle, and

constitute digits 41 through 50 of the numbers; since the first minor arithmetic cycle requires 44 digit-times, the information is not completely received until digit-time 6 of the second minor arithmetic cycle. If the arithmetic operation requires it to be addition or subtraction, the algebraic difference of the exponents must be obtained for purposes of controlling shifting of the numbers A or B. The second purpose served by the two shifting registers is to hold the largest positive exponent and this algebraic difference between exponents. Since the exponent information is too late for control during the second minor arithmetic cycle, the rest of this cycle is available to be used to perform the arithmetic operations on the exponents and obtain the desired numbers as settings on the shifting registers. The means by which these operations are accomplished are very similar to those applied in the rest of the Arithmetic Unit (except that complements are on 2^8 instead of 1).

The arithmetic operations on the exponents will be controlled so that the most positive exponent will occupy the shifting register formerly occupied by (a), and the difference between the exponents will occupy the shifting register formerly occupied by (b). Shifting control for the rest of the Arithmetic Unit will be supplied by the (b) register, and will be accomplished by allowing the Arithmetic Unit to continue shifting A or B until the "Coincidence Detector" shows that the number in (b) matches the setting

of a minor arithmetic cycle counter, at which time the arithmetic operation is performed and the result shifted the required number of places. Each shift of the result requires adding or subtracting 1 from the (a) register, depending on the direction of the shift.

For multiplication and division, the exponent unit will be controlled to yield the signed sum or difference, respectively in the (a) register; in addition the sign of the result of the arithmetic operation on A and B will be stored there. The (b) register is not needed for control in these two operations.

c. Shifting in Addition and Subtraction

The method of shifting numbers in the arithmetic unit will be described with the aid of Figure 12. In this figure the output of the reading heads of the recirculators are shown to be obtainable at three different times (in the result track there are five), they are: 43, 44, or 45 digit-times later than the time of writing. Since the length of a minor arithmetic cycle is 44 digit-times, the use of the output with 44 digit-time delay to recontrol the writing will cause each of the digits to reappear at the same point in every minor arithmetic cycle. The use of the output with 43 digit-time delay to recontrol the writing will cause the digits to appear one digit-time earlier each minor arithmetic cycle. Since the earliest digits are the least significant digits, the 43 digit-time delay causes a

shift of the number to the right of (away from) the binary point and at the rate of one digit per minor arithmetic cycle. The use of the 45 digit-time delay causes a shift to the left of the binary point.

Returning to the addition and subtraction operations (Figure 12), the four extreme cases can be classified according to two operational requirements: (1) maximum shift of one of the operands, or (2) maximum shift of the result. Class (1) requires 39 shifts of one of the operands before performance of the arithmetic operation, and inspection of the 41st digit (if addition occurs) for a possible "carry" in the most significant position, or inspection of the 40th digit (if subtraction occurs) for a possible "borrow" in the most significant position; the possible shift of the result requires the 40th minor arithmetic cycle. Class (2) requires either zero or one shift of one of the operands, performance of the operation (only subtraction causes difficulty) and sequential inspection of the lesser significant figures until a 1 is found or until all digits are shown to be zero. The operation of the Arithmetic Unit is such that three of four cases may be planned without difficulty, but the fourth case requires the addition of a delay unit to the result memory track (there are a number of ways to take care of the fourth case, but this appears to be the least complex). The three cases contain all of class (1) and "no shift" in class (2). To prepare

for the three cases, the arithmetic unit will be initially instructed to shift both numbers one place to the right of the binary point during the second minor arithmetic cycle, and during this same cycle the sum $[A + (1-B)]$ is to be stored in the result memory track with one shift to the left of the binary point. This leaves the Arithmetic Unit in a position to continue shifting one of the operands, and to shift the other one back, upon receipt of instructions prior to the third minor arithmetic cycle; the result which may require 39 shifts is also available if the instructions require it. The fourth case cannot be planned in any simple manner, so it will not be carried out until instructions appear, and the additional shifts which the result requires will be performed by an extra delay unit. This will be taken care of during the last minor arithmetic cycle, when special control voltages are made available for several other operations. (Actually two extra delay units will be provided in the result memory track to simplify the Extract operation, so no appreciable contribution to complexity is required.)

d. Multiplication

The multiplication circuit is shown in Figure 13. In each minor arithmetic cycle the multiplication circuit multiplies each digit of the multiplicand by one digit of the multiplier (successively more-significant digits), and the new set of partial products are added to the shifted

previous sum of partial products (shifted one place to the right of the binary point). The multiplication process is particularly simple, for multiplication by 1 or by 0 are the only cases; multiplication by 1 requiring only the addition of the multiplicand to the shifted previous sum of partial products; multiplication by 0 requires nothing to be added.

Since $1 > A, B \geq 1/2$, $1 > A \cdot B \geq 1/4$, this implies the necessity for inspecting the digit-position $1/2$ when the operation is completed; if zero, the result must be shifted one place and a 1 must be subtracted from the algebraic sum of exponents in the exponent register.

e. Division

The division circuit is shown in Figure 14. The condition on the division process which enables carrying out the process in 40 steps, is that the dividend is less than twice the divisor. Since the numbers, A and B, satisfy $1 > A, B \geq 1/2$, then $2B > A$ (exponent values ignored). The process of division will be carried out essentially by continuous subtraction, but with a slight modification:

Let: $A = R_0$ (remainder after zero steps)

$B = D$ (divisor)

q_n be the nth generated quotient digit
(n 1, 2, ...)

$Q_N = \sum_{n=1}^N q_n \cdot 2^{1-n}$ be the N place quotient.

The normal process:

$$R_n - 2^{-n} \cdot D = R_{n+1}$$

If R_{n+1} is negative, make $q_{n+1} = 0$ and:

$$R_n - 2^{-n} \cdot D + 2^{-n} \cdot D = 2^{-n-1} \cdot D = R_{n+1} - 2^{-n-1} \cdot D = R_{n+2}$$

The modified process:

$$R_n - 2^{-n} \cdot D = R_{n+1}$$

If R_{n+1} is negative, make $q_{n+1} = 0$ and:

$$R_n - 2^{-n} \cdot D + 2^{-n-1} \cdot D = R_{n+1} + 2^{-n-1} \cdot D = R_{n+2}$$

The results are identical, but the steps required in the latter case are equal to the number of quotient digits generated.

As proof that the condition on division is sufficient when utilizing this divisional process: (only 40 digits can be considered)

$$Q_{40} = \sum_{n=1}^{40} 2^{1-n} = 2 - 2^{-39} \quad (\text{all } q_n \text{'s being unity yields largest } Q)$$

where $2 - 2^{-39}$ is the largest possible 40-digit number, N , satisfying $N < 2$, so:

$$(2 - 2^{-39}) \cdot D = R_0$$

A properly controlled division circuit is, then capable of carrying out the divisional process with the numbers A and B . It follows from the above that $R_n \cdot 2^n < 2 \cdot D$, and also that $|R_n| \cdot 2^n < 2 \cdot D$, which will be of great importance in setting up the divisional process. To attain maximum accuracy the full 40 significant digits of D will be retained, the remainder, R_n , will be shifted

one place toward the binary point each minor arithmetic cycle, yielding:

$$R_n \cdot 2^n - D = R_{n+1} \cdot 2^n \quad (\text{for } R_n \text{ positive}).$$

Although R_n satisfies $|R_n| \cdot 2^n < 2 \cdot D$, it does not necessarily satisfy $|R_n| \cdot 2^n < 1$ (this is necessarily satisfied only for $R_n > 0$), so that complements must be taken on 2 rather than on 1. Since $D < 1$:

$$R_n \cdot 2^n + (2-D) < 2 + D < 2 + 1$$

also: (for positive R_n)

$$R_n \cdot 2^n + (2-D) \geq 2 - D > 2 - 1$$

This makes possible the use of either the digit position 2 or 1 of the remainder (digit-times 42 or 41) as a check digit (to check sign of R_n and to obtain the quotient digit) for the result. The digit in the digit-position of 2 is equal to the quotient digit, for if $R_n \cdot 2^n \geq D$, then $R_n \cdot 2^n + (2-D) \geq 2$, and if $R_n \cdot 2^n < D$, then $R_n \cdot 2^n + (2-D) < 2$. The digit in the digit-position 1 of the remainder is as good an indication, since it is just the alternative to the quotient digit. The case of a negative R_n resulting from the subtraction (addition of the complement on 2) of D , yields the complement of R_n . Since the result of this operation yields actually $2 - |R_n| \cdot 2^{n-1}$, the value $|R_n| \cdot 2^{n-1} < 1$, so the complement may be considered to be

taken on 1, which is accomplished by not writing either of the digits in the digit-positions of 1 and 2. The shift of $1 - |R_n| \cdot 2^{n-1}$ yields $2 - |R_n| \cdot 2^n$ for the next operation, which is the desired value.

The complete divisional process is accomplished by reading in A and B, recording the first 40 digits of $A + (2-B)$ in the "A" memory track and B in the "B" memory track. The digit in the 42nd digit-time (or the opposite of the 41st digit) is retained in the "carry" unit and will be written in the "result" (or "R") memory track during the 3rd digit-time of the following minor arithmetic cycle. This quotient digit is available for control for the next operation, and if it is a 1, the following operation is again to be the sum of $2R_1 + (2-B)$, where $2R_1$ is the shifted 40-digit value of $A + (2-B)$; if the quotient digit is 0, the following operation is to be $(2 - 2R_1) + B$, where $(2 - 2R_1)$ is the shifted 40-digit value of $A + (2-B)$ as before. Again the new quotient digit is found in the 42nd digit-time and is held over until the 3rd digit-time of the following minor arithmetic cycle, the previous quotient digit being shifted one place toward the binary point. This process yields 40 binary digits and $1 > A, B \geq 1/2$, $2 > Q \geq 1/2$ implies the need of inspecting the digit position of 1, to determine the possible shift required. If a shift is needed, the addition of 1 to the algebraic sum of $(\pm) a - (\pm) b$ is required.

f. Extraction

The extract operation is also performed in the Arithmetic Unit (see Figure 15). In the extract operation the full 50-digit numbers are to be kept circulating, and they are to be kept circulating with the 55-digit minor memory cycle period. This is accomplished by connecting the 10-digit exponent shifting registers in series with the presently available 45-digit arithmetic memory tracks, totaling 55 digits of delay. The two tracks used in this case will be the "A" track and the "R" track, the "R" track being used because it has two additional delay units, totaling 57 digit-times. The shift of B (in the "R" track) is under the control of the Order Interpreter, which has stored in the D and E units the information: (1) digits 31-36 specifying the digit in A, with which the replacement begins; (2) digits 37, 38, 41-44 specifying the number of digits in A which are to be replaced; and (3) digits 45-50 specifying the digit in B which begins the replacement. To obtain the proper control information, the digits 45-50 are to be replaced by $(45-50)-(31-36)$, where () means the number specified by the included digit-spaces; the result of this subtraction yields the number of places that B must be shifted, and the sign of the result yields the direction of the shift. Since shifts up to 50 places may be required, and since only 40 minor arithmetic cycles are available, the allowance of 2-place shifts afforded by the 57 available digit-time delay in the "R" track, is necessary.

The process carried out is: (1) shift B the required number of places (B is in the "R" track), (2) write A into the "R" track until reaching the digit specified by (31-36), (3) retain B for the number of digits specified by (37, 38, 41-44), and (4) recommence writing in A until 50 digits have been recorded. The desired result is now available in the "R" memory track, and will be read into Short Memory during the first minor memory cycle of the following major memory cycle.

7. Input and Output

This section covers the Input and Output Units (see Figure 16). It is planned at present to use punched-tape in input and output devices, but to speed up the transfer of information between the magnetic drum and these devices by utilizing a portion of the magnetic drum as intermediate memory. The method which will be employed, is to supply two 32-word memory tracks for each of the two functions, furnishing a 64-word memory for each. The input memory system will receive several digits at a time from the input tape, recording these digits in a particular word position, when a complete word has been received and stored, the following word position will be the recipient of the next digits. This will continue until the 64-word positions are filled with new information, at which time the input tape is stopped. When a Read-in order is received by the computer, these words will be read into the "result" Short

IV. EVALUATION

In evaluating the proposed computing machine, the greater emphasis will be placed upon the model to be built at the University of Wisconsin, the WISC, essentially because more concrete statements may be made about its expected operational characteristics. The evaluation will be made with respect to problem preparation and programming, accuracy, speed of operation, and physical complexity.

In simplicity of problem preparation and programming, the integrally synchronized computer combines some of the better points of a number of the other computers. It incorporates the "floating point" feature, which has previously been included only in the relay computers. This will greatly reduce the labor of determining the most desirable arrangement of operational steps and scale factors, for with a fixed point the numbers must not only be kept smaller than a fixed limit, but, for accuracy, they must not be allowed to become much smaller. It should be noted that this feature may be programmed in many of the other computing machines by a series of logical operations, but the resultant speed of operation is greatly reduced.

The second feature which simplifies programming is the uniform accessibility of the memory; that is, no preferential sequencing of orders and numbers within the memory exists, for the full memory capacity is made available for the procurement of each desired number and order. This

feature is shared with the IAS Computer (Princeton, N.J.) and the EDVAC (Aberdeen Proving Ground, Md.). The uniform accessibility of the memory completely eliminates the study of the interrelationships between operational sequences and apportionment of storage locations which yield the highest rate of operation the computing machine is capable of attaining.

The third feature does not have such an evident advantage, the feature being a three-address order system rather than a single-address system; its actual advantage being dependent on the operational properties of integral synchronization. In the majority of the computing machines a saving of time may be achieved by use of a single-address order system, for numbers which are to be used in successive operations may be retained in the arithmetic unit, saving the steps of storing and/or recalling them from the memory system. In the integrally synchronized computer the full set of steps for the three-address order system are done simultaneously, requiring precisely the same length of time that would be required if the orders were of the single-address type. The three-address system has, as a result, at least one advantage for programming simplicity, for no attention need be paid to the order of arithmetical operations leading to the most efficient use of time. A second, but more tenuous advantage, is that the three-address order system corresponds to a complete arithmetical operation

which is generally a more familiar concept than the sub-arithmetic operation employed in the single-address system; the argument of familiarity is, of course, only valid in the case of a relatively inexperienced programmer. A four-address order system is used in the EDVAC. The fourth address system automatically reads orders from sequential addresses.

In fairness, it should be admitted that the advantages given in the foregoing are dependent on the variety of problems which are to be solved in the computing machine, for if the same type of problem is to be solved for a large number of cases, the ease of programming becomes relatively unimportant. The uses of the WISC at the University of Wisconsin are expected to consist of a large number of greatly differing problems, each to involve a relatively small number of cases, hence the importance of simplicity of problem preparation and programming.

In accuracy of computation the integrally synchronized computer compares very favorably with all of the rest of the computing machines. This machine handles numbers in the form $m \cdot 2^{\pm q}$, where p is expressed by 40 binary digits, approximately equivalent to 12 decimal digits; q ranges from 0 to 255, yielding an approximate variation of $10^{\pm 75}$. The 40 binary digits are about as many as are employed in any of the computing machines, and the floating point feature allows the sustained utilization of the 40 significant

figures throughout the computation, accruing errors due to round-off but not due to variation in size of numbers. If greater accuracy is needed, it is possible to program arithmetical operations to achieve 80 binary digits accuracy, as is possible in most of the other computing machines.

The speed of operation of the integrally synchronized computer is directly proportional to the presentation rate of digits, since an arithmetical operation is performed every 1,760 digit-times. If mercury delay lines were to be employed in the memory system, presentation rates of one digit per microsecond could practically be realized, yielding an operation time of 1.76 milliseconds for all operations. The WISC will employ a magnetic drum, presenting digits at the rate of one digit per ten microseconds, yielding an operation time of 17.6 milliseconds (actually about 17 milliseconds is closer to the operation time). The 1.76 milliseconds is about the average time for the electronic computers which utilize the mercury delay line as a memory device. The operation time of the WISC is considerably slower, but no figures are available for comparison with other computing machines utilizing a magnetic drum as a memory device (the comparison must be made with the average operation time during computation, not with the minimum achievable time). Comparison of operation time is quite difficult, for in the integrally synchronized computer it is constant, whereas all of the other electronic computers

require different amounts of time for each of the arithmetical operations, and each of these times must be weighted with their frequency of occurrence in computation before an average operation time may be determined. It should be stated that the WISC will definitely achieve the speed of operation of a number of electronic computing machines, even though it cannot be considered as a high-speed electronic computer.

The comparison with regard to physical complexity can only be made with predicted requirements; these predictions are based on preliminary studies of component requirements and upon the design and testing of a number of basic units. These preliminary studies and tests have led to a prediction of requiring in the neighborhood of 400 vacuum-tube envelopes, but the actual number cannot yet be fixed.* The use of magnetic amplifiers is being considered, but no conclusions have yet been reached.

It seems clear that the integrally synchronized computer represents a logical design which incorporates a number of desirable features: (1) simplicity of problem preparation and programming, (2) sufficient accuracy, enhanced by the floating point, for solution of most problems,

* This number of vacuum tubes will not be supplemented by a large quantity of germanium diodes, such as are employed in many of the electronic computers, for the functions performed by most of the vacuum tubes will be those of shifting registers and amplifiers rather than performing logical operations.

(3) a speed of operation comparable to that attained by most electronic computers (the WISC being a relatively slow version of the integrally synchronized computers), and (4) a very considerable reduction in the physical components needed to satisfy the requirements.

(3) a speed of operation comparable to that attained by most electronic computers (the WISC being a relatively slow version of the integrally synchronized computers), and (4) a very considerable reduction in the physical components needed to satisfy the requirements.

APPENDIX I

PROGRAM OF A SIMPLE CALCULATION

As an illustration of the process of programming in a computing machine of the nature of the one described in this thesis, a very simple calculation will be employed. This calculation will be programmed twice, the first time ignoring the necessity for acquiring some of the numbers from "Short Memory" addresses and the second time including them.

The calculation chosen is the evaluation of the polynomial $f(x) = x^3 - 7x - 6$ for values $-2.5 \leq x \leq 3.5$, where x is to be taken in steps of 0.1 throughout this range.

The programming procedure will first be outlined and then carried out.

a. Outline

The Setup block will consist of placing $x_0(-2.5)$ in the proper storage location, i.e. where it is called for in the various orders, and that the location for the storage of $f(x_0)$ is properly specified.

The Operation block will consist of generating $f(x_j)$ from x_j and storing $f(x_j)$ in the specified address.

The Alternative block will consist of comparing x_j with $x_f(3.5)$ to see if the range of x has been covered. Provision for the transfer of the calculational process to

the proper following block, depending on the results of the comparison, must be made.

The Substitution block will consist of adding 0.1 to x_j to obtain x_{j+1} , changing the specified address for storage of $f(x_{j+1})$ so that it will not replace and annihilate $f(x_j)$, and then transfer to the beginning of the Operation block for the following calculation. Sequential storage locations will be chosen for the $f(x_j)$.

The Completion block will consist of ordering the printing of the 61 values of $f(x)$ and then halting the machine.

b. Programming

Storage of constants

- 0 - save this for results of intermediate operations.
- 1 - store $x_0(-2.5)$ in this address.
- 2 - store $\Delta x(0.1)$ in this address.
- 3 - store the coefficient (7) in this address.
- 4 - store the coefficient (6) in this address.
- 5 - store $x_f(3.5)$ in this address.
- 6 - store 1×2^{-20} in this address. This number is used to increase the storage location of $f(x_j)$ by unity for $f(x_{j+1})$

Setup Block

This block really needs no programming, for the value x_0 can be obtained directly from the assigned storage location and replaced by x_j for the rest of the points.

Operational Block

Since the constants and space for storage of intermediate results only occupy the first 7 addresses, the first order in this group can be placed in location 8, and the rest placed in sequential locations. (This is advisable, since the Read-in order reads information into sequential storage locations, non-useful information would be required to fill the gap between constants and orders, or else two Read-in orders would be required.)

Illustration of Programming Symbolism

7 [1, 1, 0, M,] the order in address 7 specifies multiplication of the number in address 1 by itself and to store the result in address 0.

Program

7 [1, 1, 0, M,] x_0^2 stored in 0.
 8 [0, 3, 0, S,] $x_0^2 - 7$ stored in 0.
 9 [1, 0, 0, M,] $x_0(x_0^2 - 7)$ stored in 0.
 10 [0, 4, 17, S,] $x_0^3 - 7x_0 - 6 = f(x_0)$ stored in 17.

This completes the coding of the Operation Block.

Alternative Block

Again these orders should go into the following addresses.

11 [5, 1, 15, C,] $3.5 - x_0$ is checked for sign.

If the result is positive, the machine reads the order in address 12, if negative, the machine reads the order in address 15.

Substitution Block

- 12 [2, 1, 1, A,] $x_0 + 0.1 - x_1$ stored in 1 (x_{j+1})
replaces x_j
- 13 [6, 10, 10, A] $17x_2^{-20} + 1x_2^{-20}$ stored in 12
[address for $f(x_{j+1})$] replaces the
address for $f(x_j)$.
- 14 [, , 7, T,] transfer to the order in address 7.

The machine now repeats the former process with a new value of x .

Completion Block

- 15 [17, 78, 16, 0] Read-out the numbers in addresses
17 to 78 (61 numbers), then read
the order in address 16.
- 16 [, , , H] Halt the machine.

Remarks

The machine calculates the values of $f(x_j)$ and stores them in the locations 17, 18, - - -, 77, then prints out these results.

The information required by the machine is the set of constants and the orders, these must be provided by the input device in the proper order so that they are stored in the specified memory locations. The machine must also be given a manual instruction to read this information into itself and then proceed to carry out the first order. The Read-in order is:

- [1, 17, 7, I,] Read-in words into locations 1 up to 17,
then carry out the order in location 7.

The information must be supplied by the input device in

this order:

-2.5	These will be stored in:	1
0.1		2
7		3
6		4
3.5		5
1×2^{-20}		6
[1, 1, 0, M,]		7
[0, 3, 0, S,]		8
[1, 0, 0, M,]		9
[0, 4, 17, S,]		10
[5, 1, 15, C,]		11
[2, 1, 1, A,]		12
[6, 10, 10, A,]		13
[, , , T,]		14
[17, 78, 16, O,]		15
[, , , H,]		16

c. Program with Short Memory Addresses (see Chapter II)

The arithmetical operations in the Operation block frequently call for numbers which are the results of the previous or next previous operation. In such cases these numbers must be obtained from the output of the Arithmetic unit or from the "Result" recirculator in the Short Memory, respectively, for they have not yet been filed into Standard Memory. The destination of the result must be stipulated

in each order, but this result is frequently required before reaching its destination.

The "Short Memory" addresses will be stipulated in this fashion:

[a, b, c, (1,0)M,] ignore address a, get number from the Arithmetic unit.

[a, b, c, (0,1)M,] ignore address b, get number from the Arithmetic unit.

[a, b, c, (2,0)M,] ignore address a, get number from Short Memory.

etc.

These may occur in pairs as well:

[a, b, c, (1,2)M,] ignore a and b, get these numbers from the Arithmetic unit and Short Memory, respectively.

New Operation Block Program

- 7 [1, 1, 0, (0,0)M,] x_0^2 stored in 0.
 8 [0, 3, 0, (1,0)S,] $x_0^2 - 7$ stored in 0.
 9 [1, 0, 0, (0,1)M,] $x_0(x_0^2 - 7)$ stored in 0.
 10 [0, 4, 17, (1,0)S,] $x_0^3 - 7x_0 - 6 = f(x_0)$ stored in 17.

This is all of the reprogramming required. The new set of information supplied to the input device will be (Read-in, Read-out, Halt and Transfer orders have no "Short Memory" addresses):

-2.5	to be stored in:	1
0.1		2
7		3
6		4

3.5	5
1×2^{-20}	6
[1, 1, 0, (0,0)M,] multiply	7
[0, 3, 0, (1,0)S,] subtract	8
[1, 0, 0, (0,1)M,] multiply	9
[0, 4, 17, (1,0)S,] subtract	10
[5, 1, 15, (0,0)C,] compare	11
[2, 1, 1, (0,0)A,] add	12
[6, 10, 10, (0,0)A,] add	13
[, , , T,] transfer	14
[17, 78, 16, 0,] read-out	15
[, , , H,] halt	16

APPENDIX II
SURVEY OF PRESENT MACHINES

The survey of computing machines presented in this chapter is not intended to be exhaustive, with regard to either the machines included or the information pertaining to them, but rather to present a few characteristics appropriate to the evaluation of the proposed computer. These characteristics will include operating speeds, internal memory capacity, ease of programming and complexity. Three relay computers, two mixed relay and electronic computers and four electronic computers will be considered.*

1. Harvard Mark I Calculator (IBM Automatic Sequence Controlled Calculator)

The Mark I employs relays for switching and IBM counter wheels for storage and for arithmetical operations. The Machine's elements are (1) the sequence control, which directs the programming from punched tape, (2) three interpolators, which are tape-fed units for selecting data required in the interpolation process, (3) functional counters for controlling the interpolation of functions, and for computing logarithmic and trigonometric functions and printing, (4) a multiplying-dividing unit for 23-digit accuracy, (5) 72 storage counters for the intermediate storage of results

* These statistics were obtained from "High-Speed Computing Devices", by the staff of Engineering Research Associates, Inc., Chapters 9 and 10.

up to 23 digits capacity, used for arithmetic operations and accumulation, and (6) a storage unit with a capacity of 60 23-digit decimal numbers which can be introduced to the machine by manually setting dial switches.

Arithmetic operations are performed to 23 decimal digits, but instructions can be introduced for 12 or 46 digit operation. Addition and subtraction are accomplished in any of the 72 adding-storage registers and require 0.3 seconds. Multiplication and division are performed in the multiplication-division unit and require about 5.7 seconds and 15.3 seconds respectively. Calculation of logarithmic and simple trigonometric functions are generated by arrays of relay elements and require about 1 minute to be generated.

The programming involves controlling the operations of the 72 adding-storage registers as well as the rest of the components. (A register is a mechanical or electronic device for storing a number, and may or may not be able to perform the function of addition.)

2. Harvard Mark II Calculator

The Mark II employs relays for the internal storage of numbers, for the transfer of numbers, for performing the basic arithmetical operations, and for sequence control of these processes; the relays totaling 13,000. The machine handles 10-digit decimal numbers and uses the floating decimal point. About 100 10-digit decimal numbers can be stored in the machine; additional internal or intermediate storage

may be supplied by punching data on tapes for introduction to the machine at a later time.

A number N is represented by the calculator as $N = ip \times 10^j$ where $1 \leq p < 10$ and $-15 \leq j \leq 15$. Addition and subtraction require about 0.2 seconds for the complete operation, and multiplication requires about 0.7 seconds. Division is accomplished by multiplying the reciprocal of the divisor.

Six algebraic and transcendental functions are stored permanently within the machine; the reciprocal, the reciprocal square root, the logarithm, the exponential, the cosine and the arctangent. These functions may be computed to eight or nine significant digits in from 5 to 12 seconds if the function and argument lie within the digit capacity of the machine.

The machine has 12 input mechanisms, each employing punched tape. Four of these supply commands, four supply numbers, and the other four are used for supplying the calculator with coded tables of functions.

It is interesting to note that the Mark II can be operated as a whole on one problem or can be split into two parts and each operated separately.

Programming again involves controlling the computer components, but in this machine there are only two adding registers compared to the 72 adding-storage registers in the Mark I and four multipliers compared to one.

3. Bell Telephone Laboratories' Model V

The BTL Model V consists of two computers, each containing all the components needed for dealing with a computational problem with the exception of tape reading devices which are used in common. The BTL Model V contains more than 9,000 relays and about 50 pieces of teletype equipment.

Each of the computers contains 15 storage registers; 8 sign registers; a calculator for the basic arithmetic operations and square root extraction; a routine control for directing the execution of instructions to the computer; a BTL register, for "block-forming", or number-grouping, trigonometric, and logarithmic operations; a table register and table control for receiving and storing numbers read from table tapes; a problem-registering problem control for receiving and storing numbers from sections of the problem tape; a printer-registering printer control for receiving numbers from various parts of the machine and directing the process of printing and perforating; a discriminator, which selects one out of several possible methods of computation and directs the machine to perform one; a recorder table, which contains a printing reperforator, a tape transmitter, and a distributor.

Like the Harvard Mark II, the BTL machine uses the floating decimal point, each number consisting of a decimal number of seven significant digits multiplied by an appropriate power of 10. In this machine the exponents include

the range -19 to +19. Addition or subtraction requires about 0.3 seconds, multiplication 1.0 seconds, division 2.2 seconds and square roots about 4.3 seconds.

For problems of medium complexity, usually one computer is used for carrying out the computation. For large problems, the two computers are associated by the problem programming, and both are used simultaneously during the computation. The setup time has been reduced by provision of more problem setup positions than can be used at one time. In the Aberdeen installation there are two computers, but there are four problem positions. With this arrangement two problems can be set up on the two idle positions while the computing equipment is actively using the other two. As soon as either one of the two problems has been completed, one of the standby problems is automatically connected in.

As in the other machines, programming requires controlling the operations of all of the registers.

4. IBM Card-Programmed Electronic Calculator

This calculator combines several IBM units to form the complete machine.

It uses the Type 402 or the Type 417 Accounting Machine as the master control unit and printer. The program for a calculation is contained on a deck of cards which are fed into this Accounting Machine. Each card carries an eight-digit instruction which specifies the locations in

storage of the factors to be operated upon, the operation to be performed, and the disposition to be made of the result.

For internal storage, the calculator uses the 80 mechanical counters with which the accounting machine is normally equipped. A complete record of a calculation can be printed at the rate of 150 lines per minute.

Cards are used for external storage. A Calculator Punch, Type 521, is used to prepare these cards and also to record on cards the results of the calculations. Either one or two Supplemental Storage Units, Type 941, may be used to provide storage for 16 or 32 10-digit signed numbers relayed to it from the Type 604 Electronic Calculator. The Type 604 Electronic Calculator is used to perform the various arithmetic operations. Additions or subtractions are performed faster than 2,000 per second, multiplications or divisions at 86 a second, and any of these may be combined. The total time required to carry out the instruction on a card, however, is determined by the rate at which cards are supplied, i.e., 2.5 per second, but the instruction may require a rather complex series of arithmetical operations.

The programming for the IBM-CPEC requires two different kinds of operations: (1) wiring plugboards for each of the units so that the machines will carry out the desired operations upon receipt of coded instructions, and

(2) punching cards with the coded instructions for the desired operations at any point in the calculation.

This machine is at present in use within the University of Wisconsin Computing Service.

5. The IBM Selective Sequence Electronic Calculator

This calculator employs vacuum-tube circuits for arithmetic operations and for high-speed storage, switching, and other control orders. Relays and punched tape form secondary storage units.

The calculator uses card-stock tape of width equal to the length of the standard IBM cards, so ample space is provided for representing 19 digit decimal numbers and algebraic sign per line across the tape (representation of the number is in binary or binary-coded decimal).

The internal storage capacity of this calculator totals about 20,000 19-digit numbers with their algebraic sign. This internal storage is divided among electronic, relay and punched-tape units. Electronic storage is provided for eight numbers, relay storage is provided for 150 numbers, and the punched-tape units provide storage for as many as 19,840 numbers. In addition to the punched tape internal storage the machine has table-consulting units which utilize punched-tape, and can store an additional 5,400 numbers.

The internal punched-tape storage system is comprised of three units, each having a punching unit and ten

reading stations. The table-consulting units consist of 36 punched tapes and corresponding reading stations.

Addition, subtraction, multiplication and division are performed by electronic-vacuum tube counting circuits, and this calculator contains more than 100 electronic counting units which can be combined for arithmetic operations. The counters will add or subtract two 19-digit numbers in less than 10 milliseconds, they will multiply two 14-digit numbers to yield a 28-digit product in 20 milliseconds, and they will perform division of two 14-digit numbers to form a 14-digit quotient in about 30 milliseconds.

The programming, as before, requires the controlling of the counter units and the other components.

6. The ENIAC (Electronic Numerical Integrator and Computer)

The ENIAC is the first large scale machine to make use of electronic circuits for general operations except for input, output, and certain switching functions. There are 40 separate panels with a total of approximately 1,500 relays and 18,000 vacuum tubes.

The input and output units for the ENIAC are standard IBM equipment for reading and punching cards. The numbers read in by the card reader are stored in relay banks, called the "constant transmitter", which makes these numbers available to the operating components of the machine when required.

The ENIAC is provided with two types of internal storage in addition to the "constant transmitter", one type is a set of 20 electronic counters or accumulators which perform the arithmetical operations as well as serving as an intermediate storage medium; the other type of internal storage is provided by two portable function tables. The basic difference between these two types of storage is that data can be stored in the electronic accumulators by the machine itself, whereas only the operator can store data in the function tables. The storage space available in the function tables is determined by the complexity of the problem, for this space, 3,000 decimal digits, is shared by the orders which are supplied to the machine.

Addition and subtraction are performed in any of the 20 electronic accumulators, each of which holds a signed 10-digit decimal number, and are carried out in 6 milliseconds. Multiplication is performed in 4 or 6 of the accumulators, for 10 or 20 significant digits, and is controlled by 3 additional panels in the computer. The time required for multiplication is 8 milliseconds. Division and extraction of square roots are performed in 4 accumulators, controlled by one additional panel, and require about 20 milliseconds to complete.

The ENIAC was originally provided with hand-set plug connectors to wire the various units into proper sequence, but this arrangement has since been changed by the

addition of a "converter". The converter energizes any one of 100 lines, determined by a pair of decimal digits, thus initiating the desired operation. The programming requires manually setting the two-digit orders, in the desired sequence, on the portable function tables.

7. The EDVAC (Electronic Discrete Variable Computer)

The EDVAC is unique among the computers considered in this thesis in that it utilizes completely serial operation and also because all arithmetic processes are self programmed, i.e., it is not necessary to bear in mind the operational properties of accumulators in order to program the computations.

The computer has an internal memory capacity of 1,024 44-digit binary numbers, one of which is the algebraic sign. This memory consists of 128 acoustic delay lines (mercury-filled tubes), each of which holds 8 numbers in continuous circulation. The internal memory is used for storage of both numbers and orders, and the division is at the discretion of the programmer.

Addition and subtraction require about 1 millisecond, including all transfers of numbers. Multiplication and division require about 3 milliseconds. Arithmetic operations may be performed to double precision, or to 86 significant binary digits. The EDVAC has two separate arithmetic units which may be connected to a comparison unit, and hence can detect errors.

Input and output for the EDVAC are furnished by punched tape, but higher speed devices are being designed and constructed. The total number of vacuum tubes in the EDVAC is about 3,500.

The programming is accomplished by supplying the desired set of 4-address orders to the machine. The 4-address order in general specifies a complete arithmetical operation, that is, it specifies the location of the two operands, the operation to be performed upon them, the location in which to place the result and the location of the following order to be read. In contrast to the 4-address order, most computers use a single-address order, specifying the desired operation to be performed with each individual number, and consequently requires a much larger array of orders, approximately four times as many orders are required.

8. The UNIVAC (Eckert-Mauchly Computer Corporation)

The UNIVAC is unique among all computers for two reasons: (1) the large number of self-checking features provided and (2) the large amount of intermediate speed storage. These computers are being built to handle large problems of a statistical nature (built for the U. S. Census Bureau, USAF, and Army Map Service), and hence require both of these unique features to a high degree.

Part of the self-checking features are provided by the representation of numerical and alphabetic data. A character, number, or letter, is represented by 7 binary digits

and, since 6 digits are sufficient to cover the range included, the seventh digit is used as a check digit. The check digit is chosen as a zero or a one such that the sum of digits in a character is odd. All data handled by the machine is checked in this way, and periodic checks are made of all characters stored in the high-speed memory. In addition the arithmetic operations are performed in duplicate and compared. Any error causes the machine to halt the computation and to indicate the nature of the failure.

The input and output units for the UNIVAC are magnetic tape readers and printers. The magnetic tape furnishes the exceedingly large memory capacity, totaling more than one million characters per reel of tape and provision for connecting the computer to any of ten such reels. The high speed internal storage is supplied by acoustic delay lines, and furnishes storage of 1,000 11-digit numbers.

Addition and subtraction are executed in less than 0.6 milliseconds, multiplication requires about 2.5 milliseconds, and division less than 4.0 milliseconds. Either a 22-digit product may be obtained or an 11-digit product with proper round off, according to the instruction used.

Information regarding the amount of equipment or the programming for handling alphabetical data is not immediately available.

9. The IAS (Institute for Advanced Study Computer)

The IAS computer is one of the highest speed machines to date, performing complete additions and subtractions in less than 100 microseconds. Multiplication and division require about 300 to 400 microseconds. This machine is not yet in operation.

Mixed input and output systems are planned.

Punched tape will be used for the initial coding of problems, and the data will be transferred from the punched tape to magnetic wire for input to the machine. The same system will be used in reverse for output. This system allows essentially continuous transfer from punched tape to magnetic wire at a low speed and a discontinuous, intermediate speed transfer from the magnetic wire to the internal high speed memory of the computer.

The internal memory of the IAS computer is furnished by a bank of 41 cathode-ray tubes, which store binary digits as charged areas on the screen (Williams tubes). Storage for 1,024 40-digit binary numbers is supplied by the memory. The operation of this memory is completely parallel, that is, each tube stores one digit of a number, and in corresponding positions of the Williams tube screen.

The total number of vacuum tubes in the IAS computer is expected to be about 2,000.

Programming for the IAS computer is similar to that of most, that is, the operation of accumulators must be controlled for each number handled in the computation.

The foregoing survey of computing machines has presented the trend of design in speed of computation and in internal high-speed memory capacity. In actuality these two properties are not independent for high speeds of computation require a correspondingly rapid supply of orders and numbers, and, since in most calculations the operations are iterative, a sufficiently large supply must be maintained. It is also interesting to note that the electronic computers have not incorporated the floating decimal (or binary) point, as did the relay computers. The introduction of a floating point would require a great deal of additional equipment for most of the electronic computers so, since problems can be arranged for solution without it in most cases, they have been built with a fixed point. Nevertheless, the floating point is of great advantage for almost all problems, for intermediate steps do not have to be investigated for sizes of numbers, and functions which possess a large range of variation do not require changing the problem setup for the different ranges.

The methods of programming of problems for the EDVAC were noted as being different from those of the other machines. The EDVAC programming involved specification of complete operations and hence the programmer did not need to understand the operational properties of the accumulators. This system has both advantages and disadvantages. The advantages are found in the ease of programming, for very

little experience is necessary before an individual may properly program rather complex calculations. The disadvantages are to be found in the maximum usage of the computer's potentialities, for in the other systems of programming it is possible to save steps in various parts of the computation by retaining the result of an operation in the accumulator as an operand for the next step, thus not requiring recall of this number from the memory.

APPENDIX III

SURVEY OF COMPUTING MACHINE COMPONENTS

A digital computer in general consists of five major components (see Figure 1): (1) an INPUT device which supplies information (orders and numbers) to the computing machine; (2) a MEMORY system which stores the information until the desired time and presents the information to the desired place upon instruction; (3) an ARITHMETIC component which performs the desired operations on numbers upon instruction; (4) a CONTROL unit which interprets orders and supplies corresponding instructions to all other units, the orders being supplied by the memory or input or both and under control of the CONTROL unit itself; and (5) an OUTPUT device which prints or punches the results of a computation.

1. Memory

The component which has received the greatest attention in the past is the memory system. The various devices which are serving in present machines can best be classified by attributes for proper evaluation.

Erasable - Nonerasable -- In the erasable class information which is no longer needed may be erased to provide space for storage of new data, whereas in the nonerasable class additional memory space always must be supplied if new information is to be recorded. For example, a magnetic tape is erasable, a punched card is not. It follows

that an erasable system requires a smaller physical capacity to handle a given amount of information, since part of it can be used over and over.

Volatile - Nonvolatile -- In the volatile class the permanency of the information depends upon the continued supply of power to the memory device, and in the event of a failure of the power supply the information is lost. A nonvolatile memory device retains recorded information indefinitely without requiring power. Typical of the volatile class is the electron-tube relay; nonvolatility is represented by a mechanical relay, with a self-latching device. Other things being equal, volatility is certainly an undesirable feature, since it requires redoing at least a portion of a problem in event of power failure at the storage device.

Static, Movable, and Cyclic -- In the static class an item of information stored in the memory is fixed in space, and may be obtained at any instant by closing the right switch. In the movable class the memory medium is movable in space, and to obtain a particular item of information that portion of the memory must be moved to the reader before the switch is closed. In the cyclic class the information in the memory is kept in constant circulation so that all portions are passed by the reader in sequence, and each item is available at a definite instant of the cycle, each cycle. Reading information from such a

device consists of closing switches at the right time. A bank of relays typifies the static class, a stack of IBM cards the movable, and a rotating magnetic drum the cyclic. In general, static devices are the fastest, but most expensive of both space and equipment; movable memories are very compact for their capacity, but may require prohibitively high access time. Cyclic devices furnish various degrees of compromise between these extremes.

With these factors in mind, some of the commoner devices can now be evaluated with regard to cost, access time and reliability. Access time to an item of information varies from a few seconds for the slower devices, through the millisecond range for the intermediate types, down to perhaps ten microseconds for the very fastest memories. Table II illustrates in greater detail such a classification of these memory devices. The conclusions might be summarized as follows:

Punched cards and tapes: slow in operation, comparatively quite inexpensive, and very reliable.

Mechanical relays: slow in operation, very expensive and bulky, and quite reliable.

Vacuum-tube relays: fast in operation, very expensive, require large amounts of space and power, and not too reliable due to volatility.

Electrostatic storage (Williams tube): very fast in operation, much less expensive per digit than individual tubes

but still subject to considerable development for entirely satisfactory operation, and not too reliable due to volatility.

Magnetic wires and tapes: intermediate in speed of operation, inexpensive, compact, and quite reliable.

Magnetic drum: intermediate in speed of operation (faster than wires and tapes), inexpensive, compact, and quite reliable.

Sonic delay lines: fast in operation, inexpensive, but less reliable due to volatility and temperature control requirements.

2. Input and Output

Input and output devices in general consist of movable and nonvolatile memory devices such as punched cards or tape or magnetic wire or tape. The punched cards and tapes have the advantage of allowing sight-reading and checking. Magnetic wire and tape have the advantage of higher rate of transfer of information and erasability. The choice of input medium depends upon the input or output speed requirements. Perhaps the best input-output systems for high speed memories are mixed systems where the magnetic wires or tapes are supplied with information from the punched cards or tape fairly continuously, and in turn transfer this information at a much higher rate, but very discontinuously, to the high speed memory. This is done in reverse for the output. In a system of this type the high-speed memory is

not slowed down to the input device's speed during reading-in periods, for the intermediate speed device functions as a speed-matching unit between the two.

3. Control

The control unit is a component that must be tailored for a particular machine, and depends upon the choice for all the other units in the computing machine.

4. Arithmetic

In the arithmetic component there is a similar range of possibilities. The choices to be made at any point might well depend upon the choices made at any other point. There are essentially three major factors to be considered.

(1) Parallel or serial operation; i.e., addition or subtraction of all digits in the two numbers simultaneously, or corresponding digits a pair at a time. Parallel operation is accomplished in a much smaller period of time, for example, to add two ten digit numbers using serial operation takes about ten times as long. However, the amount of equipment required for parallel operation is about ten times as large.

(2) The number system to be employed. Decimal notation has the advantage of familiarity and corresponding ease of interpretation, but has the disadvantage of requiring fairly complex circuitry to carry out the arithmetical operations. A number of ways of representing the ten

digits have been employed, such as ten-position counters, biquinary or mixed base two and base five, binary-coded decimal, and binary-coded decimal excess-three. Each of these systems has its advantages and corresponding disadvantages in simplicity of circuits, circuit design or checking.

Binary notation, using only two distinctive digits, zero and one, possesses two important advantages for machine simplicity. The first advantage is the much smaller number of machine components required to produce analogues of arithmetic operations; part of this advantage is due to the properties of circuit elements and part is due to the properties of the binary system itself. The second advantage of binary notation is that it requires less storage space to store a given number of items of information than does any other number system. The disadvantage of unfamiliarity with binary numbers can be greatly reduced, for arithmetic sequences may be employed to change number systems within the machine; i.e., numbers may be supplied to the machine in the form of binary-coded decimal numbers and be translated internally to binary numbers for use in calculation, the results of the calculation may again be translated from binary to binary-coded decimal numbers prior to printing out.

(3) The use of a fixed or floating point in positional notation; i.e., the use of a fixed number of decimal

(or binary) places, or a fixed number of significant figures. In floating point notation, a number is written (for example) with the first significant digit immediately following the point, followed by the given number of significant figures, and the order of magnitude of the number indicated by an appropriate power of ten (or two). Such notation not only increases the overall accuracy of the machine, utilizing maximum accuracy for every calculation, but it also greatly simplifies the preparation of problems for the machine. Without it, much care must be exercised in so adjusting scale factors that no intermediate result in the calculation will exceed the machine's capacity and so that the largest possible number of significant digits will be maintained. The floating point feature may generally be programmed in the electronic computers, but a considerable decrease in speed of computation and storage capacity results. Inclusion of a floating point, then, requires a more complex machine and a greater storage capacity.

This survey of machine components should serve to emphasize the more salient features involved in choosing elements for a computing machine.

OPERATION STEP MAJOR CYCLE NUMBER	READ ORDER NUMBER	Locate Operands called for in ORDER NUMBER	Perform Operation called for in ORDER NUMBER	File Away the Result of ORDER NUMBER
N	N	N-1	N-2	N-3
N+1	N+1	N	N-1	N-2
N+2	N+2	N+1	N	N-1
N+3	N+3	N+2	N+1	N
N+4	N+4	N+3	N+2	N+1
etc.				

Table I - This table shows the overlapping of the four operational steps. The four steps in any row are being carried out simultaneously, and are steps referring to four separate orders.

Table II

<u>Type of Memory</u>	<u>Speed</u>	<u>Erasable</u>	<u>Volatile</u>	<u>Static</u>	<u>Movable</u>	<u>Cyclic</u>	<u>Cost</u>
Punched Cards Tapes	slow	no	no		✓		low
Mechanical relays	slow	yes	no	✓			high
Vacuum-tube relays	fast	yes	yes	✓			high
Electrostatic (Williams)	very fast	yes	yes	✓			medium
	fast					✓	high
Magnetic wires tapes	medium	yes	no		✓		low
Magnetic drum	medium-						
	fast	yes	no			✓	low
Sonic delay lines	fast	yes	yes			✓	low

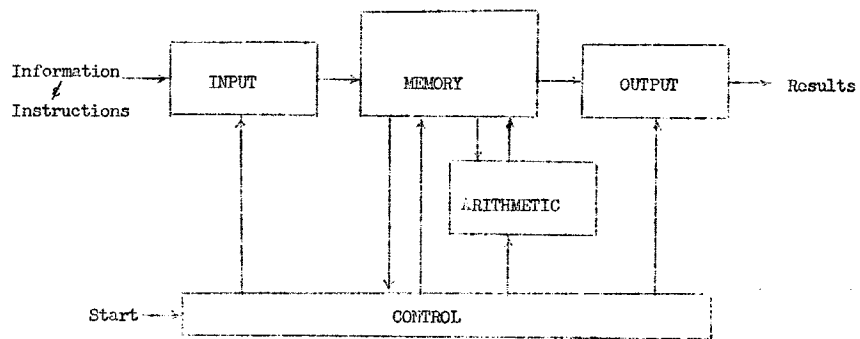


Figure 1 - Basic Components of a Computing Machine

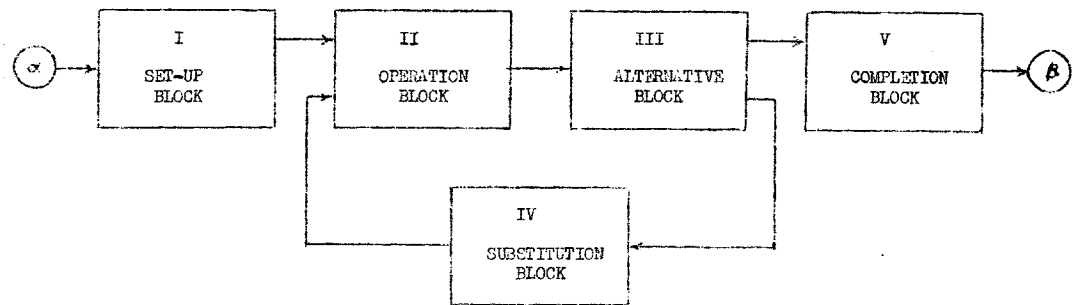


Figure 2 - Flow Diagram of a Sequence or Subsequence

Each block represents a group of orders required to carry out the designated operations. The interconnections show the flow of calculation.

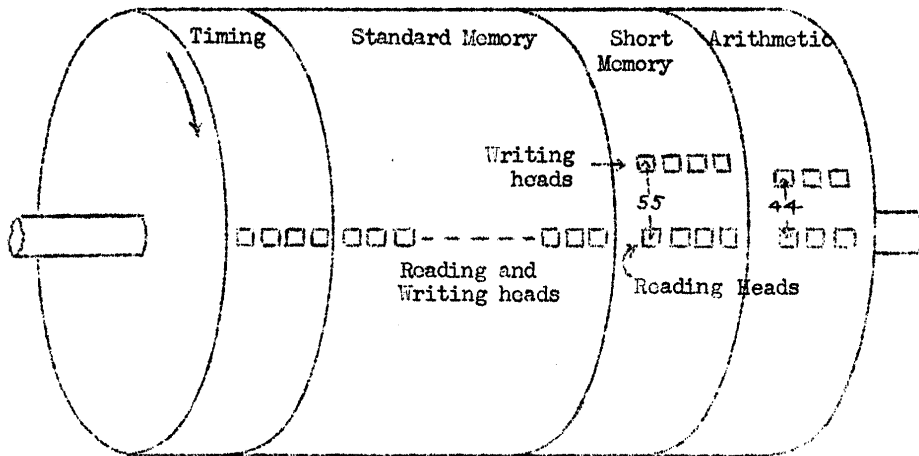
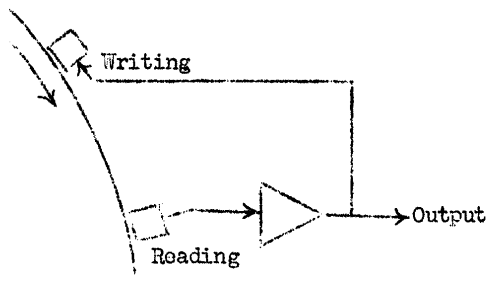


Figure 3 - Division of Magnetic Drum



Definition of symbols
Figure 4b

Figure 4a - Recirculator

The recirculator is essentially a delayed feedback circuit, the information stored in the delay section being periodically presented for external use and for recirculation through the delay section.

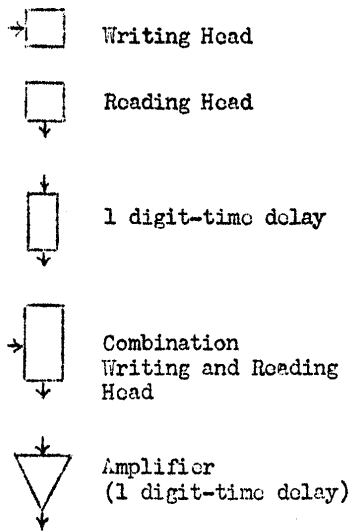


Figure 4b - Definitions of symbols

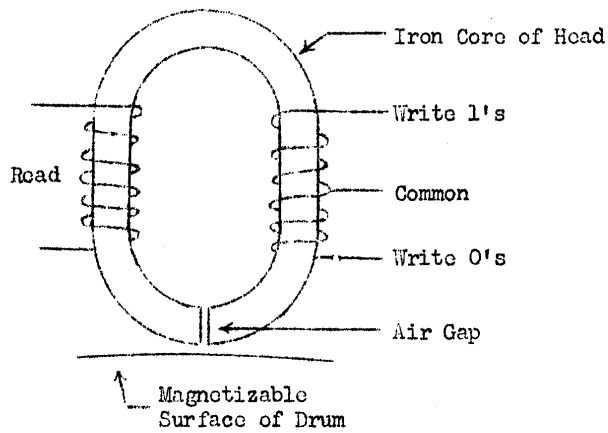


Figure 5 - Reading and Writing Heads

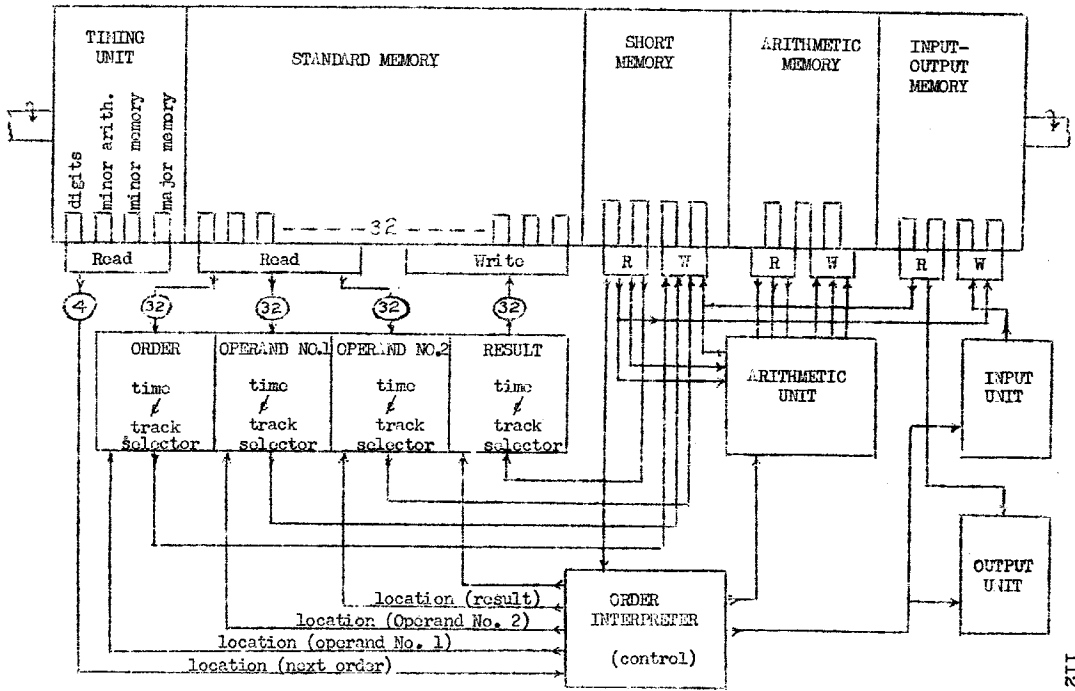
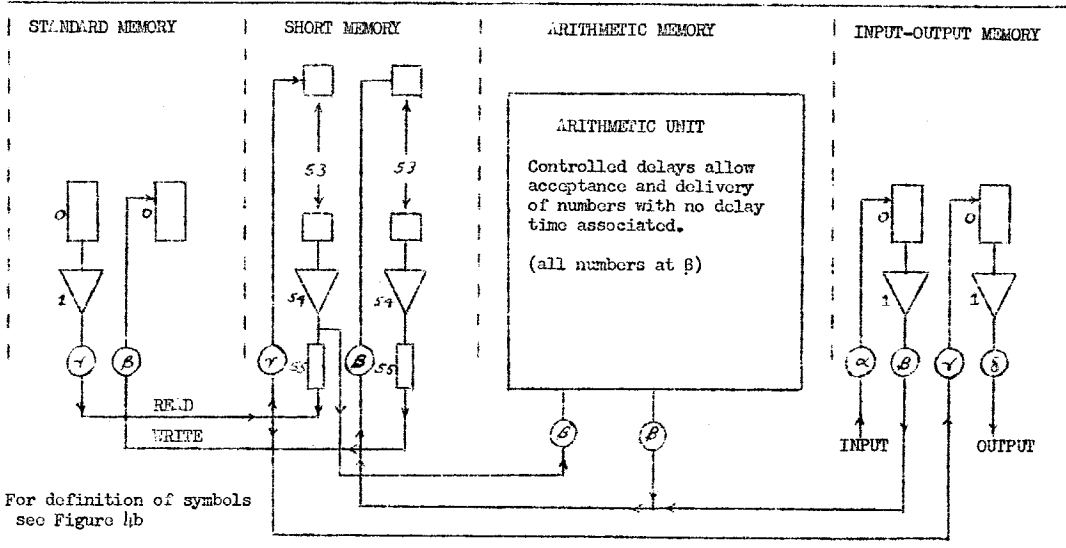


Figure 6 - Block Diagram of Computer



For definition of symbols see Figure 1b

Connected lines are lines of equal time (switches not shown)

Reference time is time of writing into standard memory (β)

- α one digit-time delay
- β reference time
- γ one digit-time lag
- δ two digit-time lag

Figure 7 - Inherent and Controlled Delays

Significant digits (40)	exponent (8)	(1)(1)
← least significant digit	sign of exponent	↗ sign of number

Figure 8a - Assignment of Digits in a Number

A	B	C	D	E
location (10)	location (10)	location (10)	order type (10)	order type (10)
Operand No. 1 (1-10)	Operand No. 2 (11-20)	Result (21-30)	All orders (31-40)	Extract only (41-50)

Figure 8b - General Assignment of Digits in an Order

Section	A	B	C	D	E
Order	Location	Location	Location	Order type and Short Memory Address	Address
Addition	augend	addend	sum	31 order (36-39)S.M.	---
Subtraction	minuend	subtrahend	difference	32 order (36-39)S.M.	---
Multiplication	multipli- cand	multiplier	product	33 order (36-39)S.M.	---
Division	dividend	divisor	quotient	34 order (36-39)S.M.	---
Comparison	minuend	subtrahend	next order	35 order (36-39)S.M.	---
Read-in	first word	last word (+1)	next order	36 order	---
Read-out	first word	last word (+1)	next order	37 order	---
Transfer	---	---	next order	38 order	---
Halt	---	---	---	39 order	---
Extract	Operand NO.1	Operand No.2	new word	40 order 39 S.M. (A only) Extract specification (31-36)(37,38,40-44) (45-50)	

Figure 8c - Detailed Assignment of Digits in an Order
(S.M. in D section refers to short memory locations).

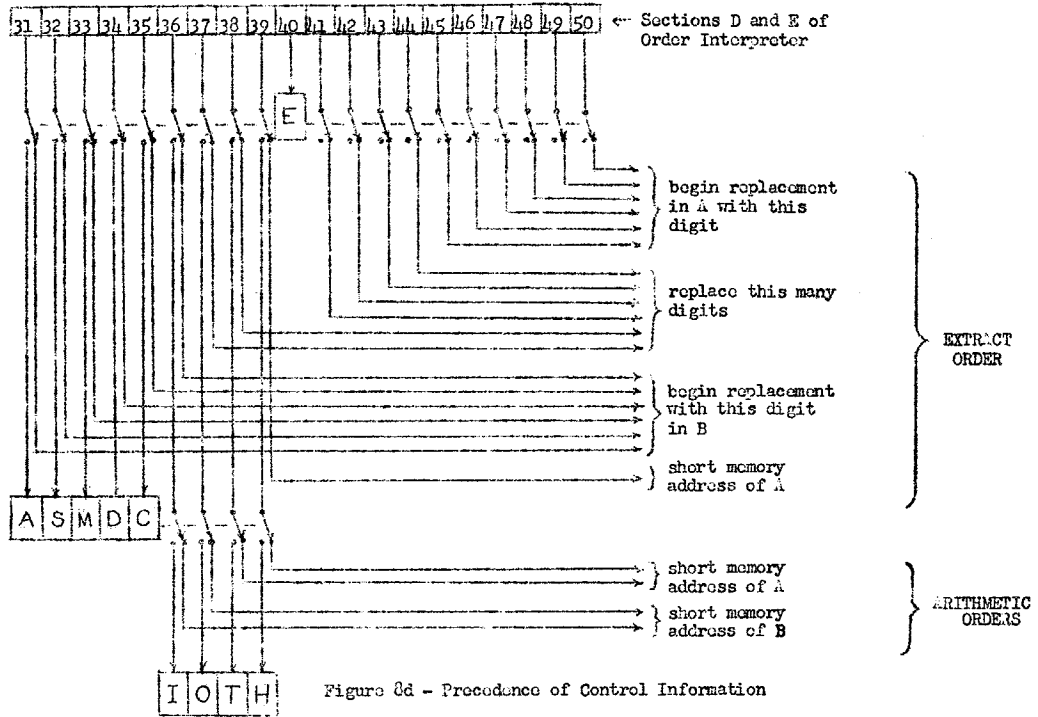


Figure 8d - Precedence of Control Information

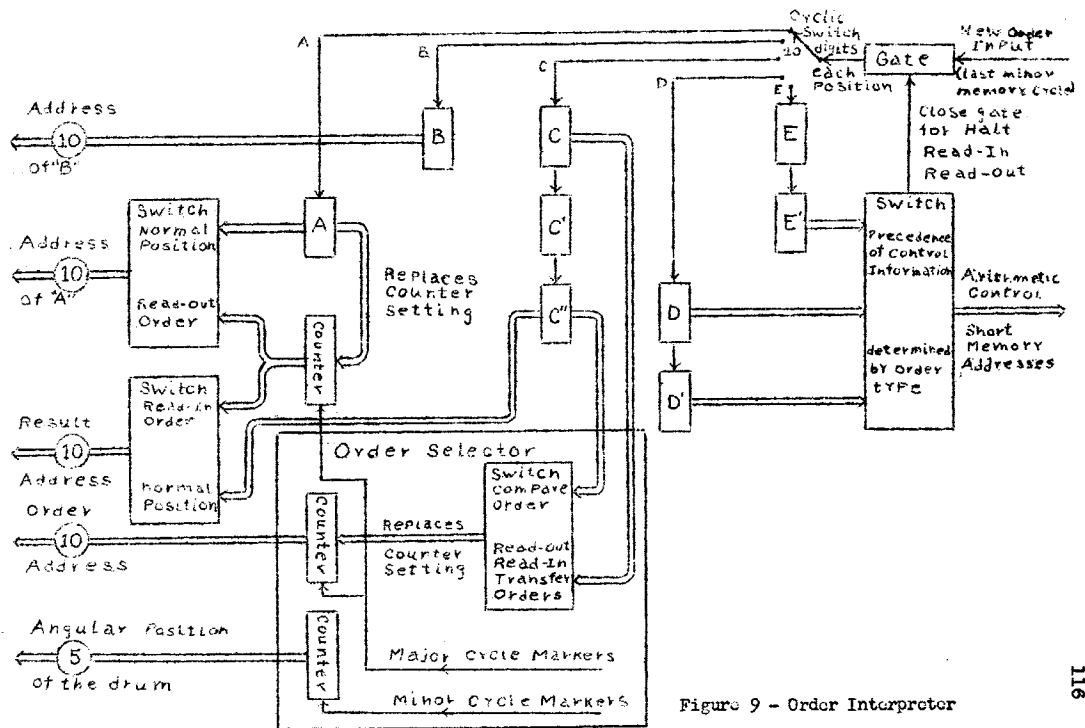


Figure 9 - Order Interpreter

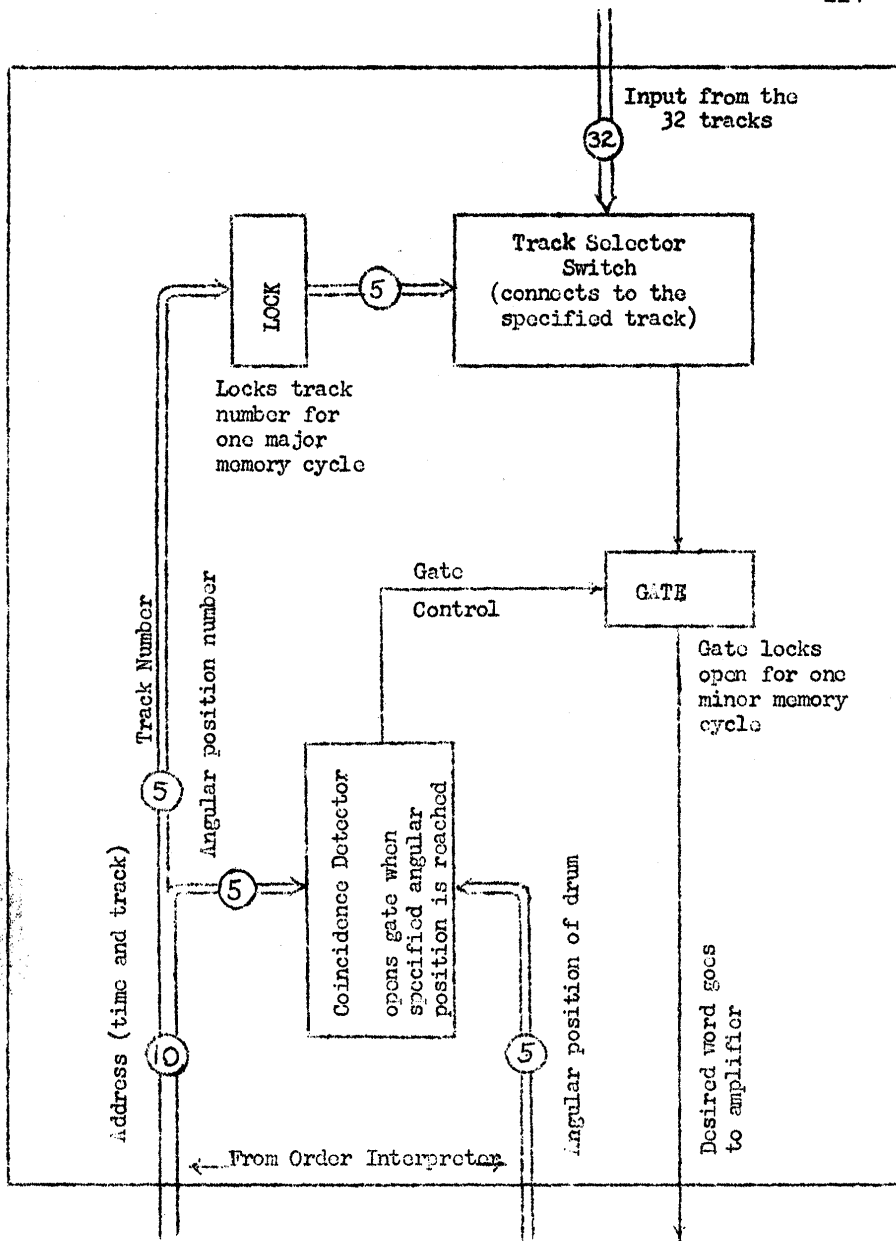
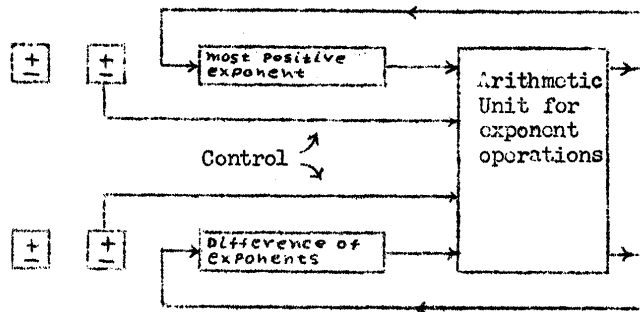
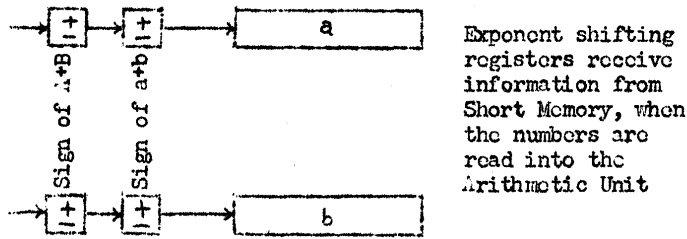


Figure 10 - Time and Track Selector



Operations on exponents are performed during the second minor arithmetic cycle to provide control information

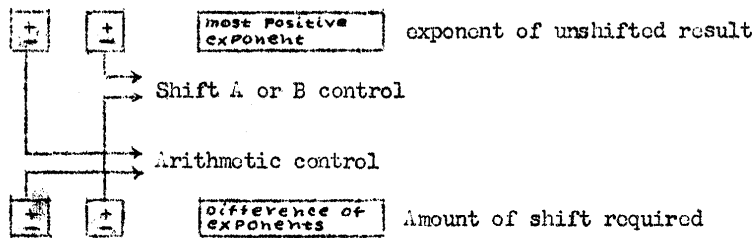


Figure 11 - Exponent Unit (shown for Addition and Subtraction Operations)

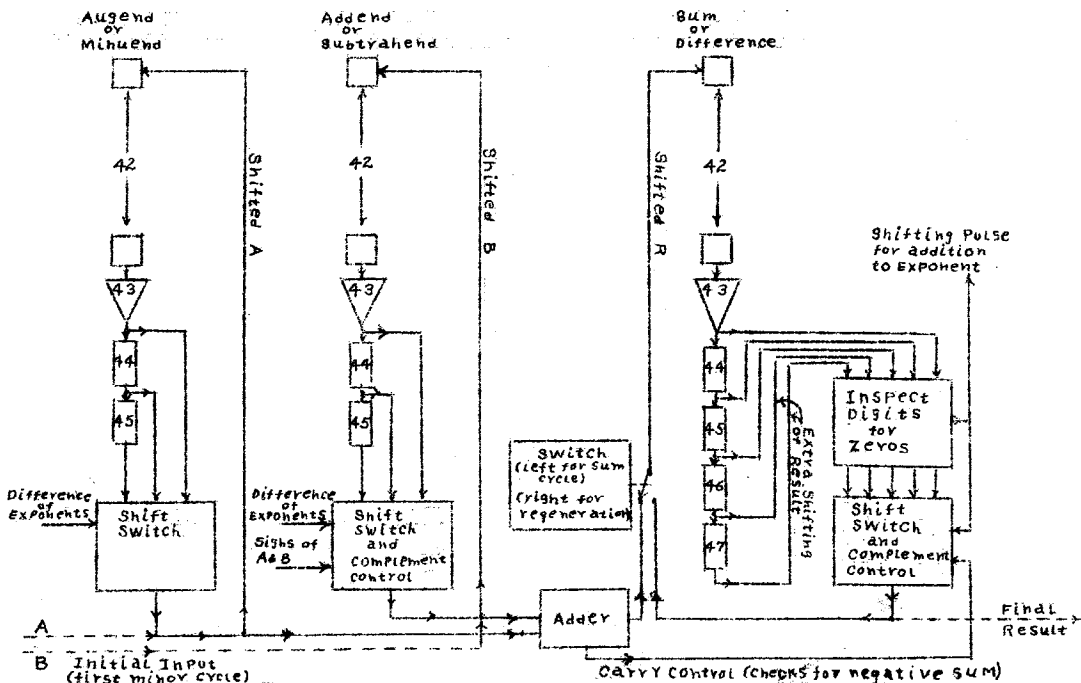


Figure 12 - Addition and Subtraction Circuits

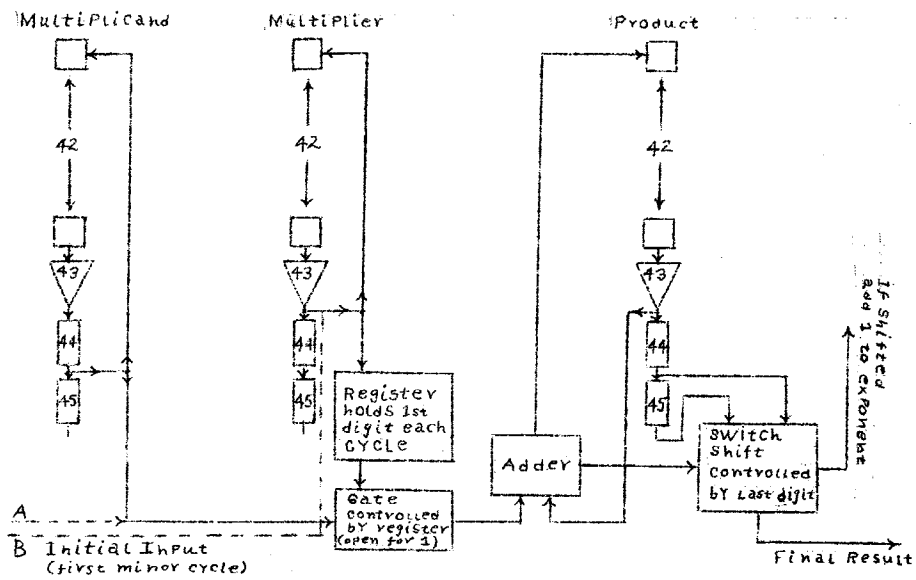


Figure 13 - Multiplication Circuit

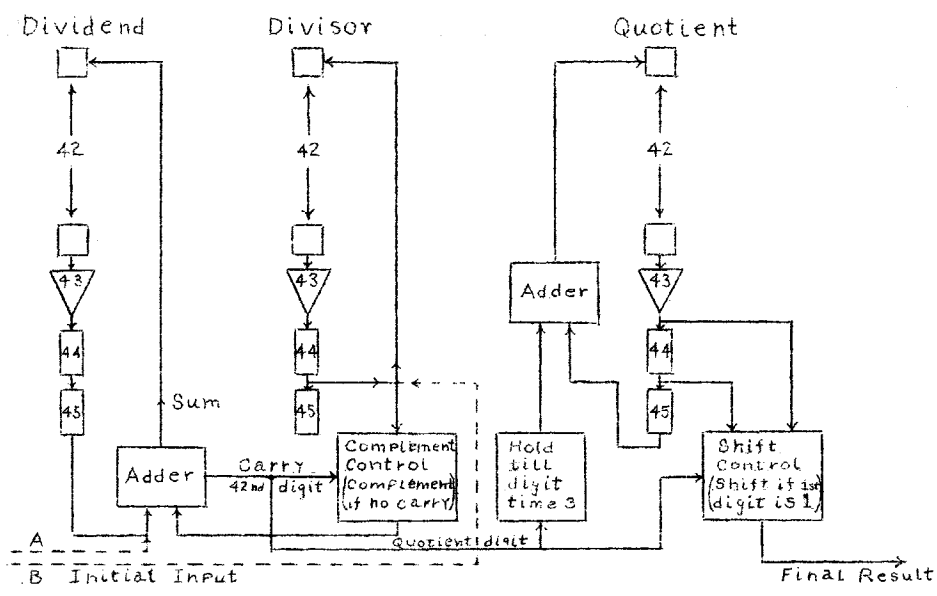


Figure 14 - Division Circuit

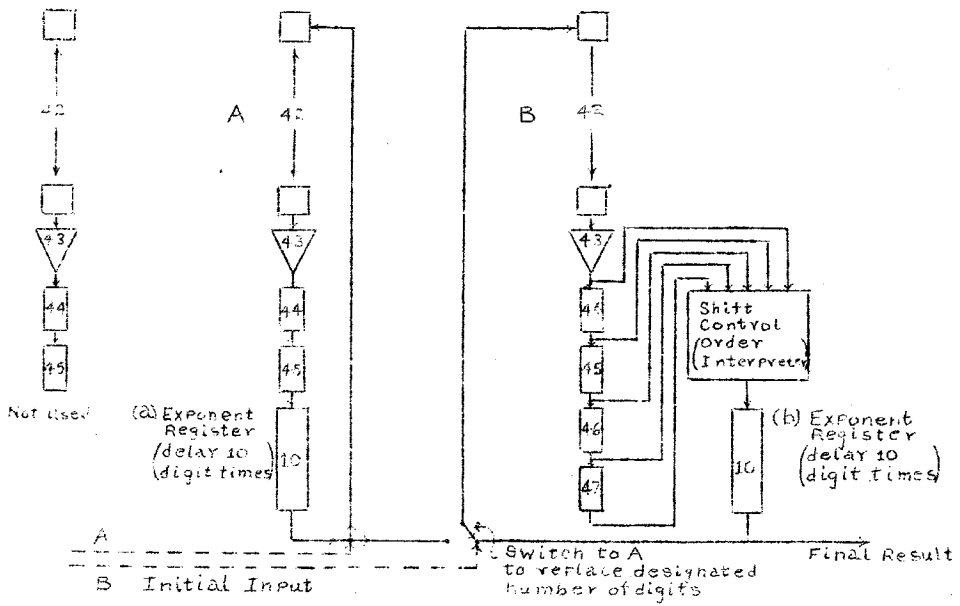


Figure 15 - Extract Circuit

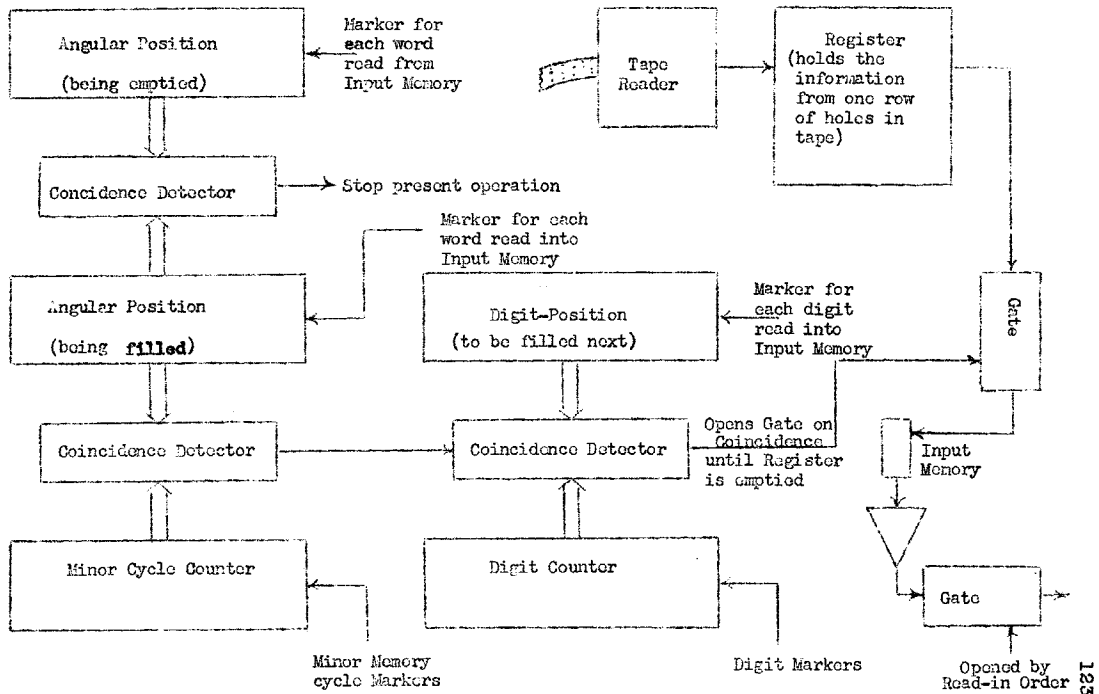


Figure 16 - Input Unit

LIST OF REFERENCES

Books

1. Calculating Machines, D. R. Hartree, Macmillan and Co. Ltd., 1947.
2. High Speed Computing Devices, Staff of Engineering Research Associates, Inc., McGraw-Hill Book Co., Inc., 1950.
3. Proceedings of a Symposium on Large-Scale Digital Calculating Machinery (Jan. 7-10, 1947), Annals of the Computation Laboratory of Harvard University, Vol. XVI, Harvard University Press, 1948.

Periodicals

4. Bloch, R. M., Campbell, R. V. D., and Ellis, M., The Logical Design of the Raytheon Computer, Mathematical Tables and Other Aids to Computation, Vol. III, No. 24, pp. 286-295, October, 1948.
5. Goldstine, H. H. and Goldstine, Adele, The Electronic Numerical Integrator and Calculator (ENIAC). Mathematical Tables and Other Aids to Computation, Vol. II, No. 15, pp. 97-110, July, 1946.

Reports and Pamphlets

6. Burks, A. W., Goldstine, H. H., and von Neumann, J., Preliminary Discussion of the Logical Design of an Electronic Computing Instrument. Report prepared under contract W-33-034-ORD-7481 between the Research Development Service, Ordnance Department, U. S. Army, and the Institute for Advanced Study, Princeton, N. J., June 28, 1946.
7. The Eckert-Mauchly Computer Corporation, The UNIVAC System, Pamphlet prepared by the Eckert-Mauchly Computer Corporation, Philadelphia, Pa., 1947.
8. Engineering Research Associates, Inc., Selective Alteration of Digital Data in a Magnetic Drum Storage Computer Memory. Report prepared for ONR, under contract N6onr-240 Task I with Engineering Research Associates, Inc., Saint Paul, Minn., Dec. 1, 1947.

9. Goldstine, H. H., and von Neumann, J., Planning and Coding of Problems for an Electronic Computing Instrument, Part II, Vol. I, 1947, Part II, Vol. II, 1948. Prepared under contract No. W-36-034-ORD-7481 between the Research and Development Service, U. S. Army Ordnance Department, and the Institute for Advanced Study, Princeton, N. J.
10. International Business Machines Corporation, IBM Selective Sequence Electronic Calculator, 1948. Brochure of IBM Corporation, New York.
11. von Neumann, J., The Principles of Large-Scale Computing Machines. Office of Research and Invention, Navy Department No. 6553.
12. University of Pennsylvania, Staff of Moore School of Electrical Engineering, Functional Description of the EDVAC, Research Division Report 50-9. A report of development work under contract No. W-36-034-ORD-7593 with the Ordnance Department, U. S. Army, and the University of Pennsylvania, Philadelphia, Pa., Nov. 1, 1949, 2 vols.

TITLE OF THESIS The logical design of an intermediate
speed digital computer.

Full Name Gene Myron Amdahl

Place and Date of Birth Flandreau, S. Dak., Nov. 16, 1922

Elementary and Secondary Education Blinsmon Grade School, Flandreau,
S. Dak. (1928-1936); Egan, S. Dak., Consolidated High
School (1936-38); Augustana Academy, Canton,
S. Dak. (1938-40).

Colleges and Universities: Years attended and degrees

S. Dak. State College, Brookings S. Dak.
(1942-44) — (1946-48) B.S. (1948)

University of Wisconsin, Madison, Wis.
(1948-1951) M.S. (1949)

Membership in Learned or Honorary Societies American Physical Society,
Sigma Xi, American Institute of Physics.

Publications None

Major Department Physics

Minors Mathematics.

Date 10/11/51

Signed Robert S. Sachs
Professor in charge of thesis

Approved R. G. Sachs
10/11/51

The Library

of the



University of Wisconsin

