UNPUBLISHED MATHEMATICAL TABLES

In this issue there is a reference to an unpublished table in RMT **1015**

147[A, P].—LEO STORCH, *Admittance-Impedance Conversion Tables*, Technical Memorandum No. 274, Hughes Aircraft Co., Research and Development Laboratories, Culver City, California. 10 p. manuscript tabulated from punched cards. Copy deposited in UMT FILE.

The table gives 4S values of $(1 + s^2)^{-1}$ and $s(1 + s^2)^{-1}$ for $s = 0(.001)1$. It is intended to facilitate the calculation of the reciprocal of a complex number. The table is an extension of a table of JAHNKE & EMDE. [4th ed., appendix, p. 13.]

148[F].—F. GRUENBERGER, *Lists of primes*. Six sheets tabulated from punched cards. Deposited in the UMT FILE.

These lists of primes are for the ranges 10100009 to 10132211 and 50000017 to 50040013. The lists were computed on a CPC as a fill-in project, without attempting to program for speed. A graph showing the distribution of differences between consecutive primes in the ranges 1000003–1024523, 10100009–10132211 and 50000017–50012839 is included.

F. GRUENBERGER

University of Wisconsin
Madison, Wisconsin

149[F].—A. GLODEN, *Table of solutions of the congruence* $x^{128} + 1 \equiv 0 \pmod{p}$ *for* $p < 20000$. Manuscript, 2 p., deposited in the UMT FILE.

The table gives for each of the 16 primes p of the form $256k + 1$ less than 20000, the 64 solutions of the congruence mentioned in the title which are less than $\frac{1}{2}p$.

A. GLODEN

11 rue Jean Jaurès
Luxembourg

AUTOMATIC COMPUTING MACHINERY

Edited by the Staff of the Machine Development Laboratory of the National Bureau of Standards. Correspondence regarding the Section should be directed to Dr. E. W. CANNON, 415 South Building, National Bureau of Standards, Washington 25, D. C.

TECHNICAL DEVELOPMENTS

AN AUTOMATIC COMPUTER IN AUSTRALIA

An automatic computer, the "C.S.I.R.O. Mk. I Digital Computer," designed and constructed by the Radiophysics Division of the Commonwealth Scientific and Industrial Research Organization in Sydney, Australia, is now in service. It is of the all-electronic, serial-binary type with a main store consisting of a group of ultrasonic delay lines with a total capacity of 1,024 words of 20 binary digits each. An auxiliary store in the form of an unsynchronized magnetic drum is incorporated with a capacity of 1,024 similar words, later to be extended to 4,096 words.

No attempt has been made to obtain very high speeds of operation, the

objective being to construct a computer which is simple and sufficiently flexible from the engineers' and mathematicians' point of view. Thus, equipment can be removed or added without dislocating the mode of operation. The device is intended at this stage primarily for research into numerical methods and programming techniques.

Standard electronic components are used throughout the 1,500 or so tubes and circuits. The clock pulse rate is 333 kcs. A minor cycle of one word length covers a period of 60 μsec, and the major cycle of 16 words, equal to the delay of each ultrasonic storage channel, covers a period of 1 millisecond. On the average the time occupied by the selection of a command and the corresponding operation is about 2 major cycles.

Numbers and Commands. Numbers are registered in the straight binary scale, with each of 19 digits together with a sign digit in the most significant place, and with negative numbers being stored in complementary form. The convention with regard to the position of the binary point is only significant in the operation of the automatic multiplier, where it is placed immediately to the right of the sign digit of the total product. Although each number consists of only 20 binary digits, it is not difficult to programme for 40 digits or even greater accuracy.

The scheme adopted for commands is of the "two address" type, in which each operation is considered as a transfer of the content of one register to another, the former being specified as a "source" and the latter as a "destination." An arithmetical function is specified as a quality of the particular transfer demanded by a command. A command is divided into three groups of digits. Two adjacent groups of 5 digits define the source and destination, and a further group of 10 digits specifies a numerical component normally used to indicate which store position is called for when either of the other two addresses involve the store. If the store is not called, the third address may be used to store special information.

These digit groupings are in the order in which digits are transmitted from a register: destination, 1–5; source, 6–10; numerical, 11–20. Commands are held in and are normally accepted from the store in serial order.

Organization. The computer is of the serial type, and all transfers take place along a single "digit trunk." All registers are connected to this trunk via "function gates" which are under the control of the central sequencing unit and the command decoding devices. The digit trunk consists of two parts, an "output trunk" and an "input trunk," and transfers are made between these conductors during single minor cycle periods, determined by a time selector which operates upon the detection of equality between digits 11–14 of the current command and the current minor-cycle number.

The selection of a single command from the store, and its performance, requires four transfers. The registers involved in controlling these actions are:

(A i) The "sequence register": a 10-digit register which keeps a tally of the progress of the programme, and instructs whence the next command is to be withdrawn. Its contents are normally increased by unity following the selection of a command.

(A ii) The "store control register," of 10-digit capacity, which is connected in the upper 6-digit positions, i.e. (15–20), to a decoding selector

used to specify which store channel is needed, and in its lower 4 digit positions (11–14) to the time selector.

(A iii) The "interpreter register," of 20-digit capacity, which is connected to the "source and destination selectors" each possessing 32 outputs.

Some of these registers possess special function gates and may be called to transfer or receive by a command, as follows:

(B i) The sequence register can be called to "count in," i.e., to increase its content by a number equal to the number of unit digits entering; to "add in" on the digit positions 11–20; and to "substitute in" on the same positions, i.e., reset and add in. Also it can be called to transmit in the digit group 11–20.

(B ii) The interpreter register can be called to transfer out its numerical content on digit positions 11–20.

The four transfers involved in satisfying a single command are of invariable form. They constitute what is called the computer routine and are as follows:

(C i) The content of the sequence register is transferred to the store control register whose content it replaces.

(C ii) Under the control of the content of the store control register, the store is allowed to transmit. The time selector finally allows the desired command to enter the interpreter register via the "input trunk."

(C iii) The command may call for the store to transmit or receive, hence the numerical part of the command is transferred from the interpreter to the store control register, replacing any previous content.

(C iv) Under the control of the time selector and the action of the source and destination selectors, a selected pair of function gates is actuated, and the desired transfer is performed.

The function gates and their actions are listed in Table I.

The Arithmetical Unit. The arithmetical unit consists of a group of 5 ultrasonic delay-line registers of which A, B, C are of 20-digit capacity, whilst H is of 10 digits only. The fifth register D can store sixteen 20-digit numbers. Registers A, C, and D can add, subtract, and replace, whilst H can read in and replace in either of the digit groups 1–10 or 11–20. Register B is a non-adding register.

Register A is the main accumulator and possesses a number of special gates for performing the functions of shifting right and left by one place, as well as reading out the most significant digit and the lowest digit. It is also capable of certain logical functions. Registers C and D are capable of reading out the sign digit of their content. This applies to any single number in D. Registers A, C, and D can read out their contents, and also may do so with simultaneous reset.

Special functions associated with multiplication have been introduced which also may be used for other purposes. For automatic multiplication, registers A, B, and C are coupled together. Register C holds the multiplicand whilst B receives the multiplier. The registers A and B are connected into series circulation together with an extra digit period, for a period of 41 minor cycles during which the multiplier digits are investigated successively and removed from circulation. The total product is built up by successive addition and shifting, until actually it occupies registers A and

TABLE I. Function gates

Command addresses: n; S, D

Sources (S)	Destinations (D)
Read Out:	*Read In:*
(1) High speed store position "n"	(1) High speed store position "n"
(2) low speed store position "n"	(2) low speed store position "n"
(3) current command positions 11–20	(3) shift to binary/decimal reading
(4) read current card column and shift to next column	(4) to output "print"
(5) position of next command	(5) to output and "punch"
(6) content of register A	(6) sequence register and "count"
(7) content of A and clear	(7) sequence register and "substitute"
(8) content of $A \times 2$	(8) sequence register and "add"
(9) content of $A \times \frac{1}{2}$	(9) and reset into A
(10) sign digit of content of A	(10) and add into A
(11) lowest digit of content of A	(11) and subtract into A
(12) lowest unit of content of $A \neq 0$	(12) and substitute A by conjunction of content and entry
(13) content of B	(13) and substitute A by disjunction of content and entry
(14) content of B shifted one place to right	(14) to shift total product left one position if unit digit received
(15) sign digit of content of B shifted one place to left	(15) and store in B
(16) content of C	(16) and multiply in B
(17) sign digit of content of C	(17) and reset into C
(18) content of D position "n"	(18) and add into C
(19) sign digit of content D position "n"	(19) and subtract into C
(20) content of H on positions 1–10	(20) and reset into D position "n"
(21) content of H on positions 11–20	(21) and add into D position "n"
(22) one lowest digit	(22) and subtract into D position "n"
(23) content of hand-set register number 1	(23) H and store group 1–10
(24) content of hand-set register number 2	(24) H and store group 11–20
	(25) and stop if unit digits received

B, the multiplier being lost. It is arranged that the binary point of the product lies between the two most significant digits of the more significant component which lies in A. Corrections are automatically made to the product at the outset if either or both the multiplier and multiplicand are negative.

For storage of double length products, B can read out one digit early, leaving its most significant digit free for the insertion of a sign digit if needed. Rounding off to 20 digits is performed by reading out the most significant digit of B with a delay of one digit period, i.e., as a digit in the last significant place, and adding it to A. Before rounding off it may be desired to retain as many digits as possible in one 20-digit word, so a "left-shift" function is available. This shifts the entire product one place to the left and may be called by a digit impulse supplied to its function gate. No right shift is available.

The H register can also read out in either of the digit groups 1–10 or 11–20 and so may be used for shifting groups of 10 digits to left or right by 10 positions.

Two special "constant registers" are provided. These are hand set to any desired set of digits for a calculation. They are particularly useful in maintenance tests and in testing programmes.

A special constant gives a single digit output in the least significant

place (i.e., position 1), and a further destination is provided to stop the sequence. This function is performed if a non-zero digit passes this gate. A list of the sources and destinations is given in Table I.

Use of the Functions. Whenever a command does not involve the store or register D, the digits in positions 11–20 may be used for other purposes, and special 20-digit constants may be compiled by transfer via the H register. Frequent use is found for special digit groupings in the positions 11–20.

Absolute transfer of control can be made by using the read-out faculty of the interpreter register and the read-in property of the sequence register, whilst relative transfers can be made by adding into the latter register. Conditional transfers are made by using the sign-selecting property of the arithmetical registers. A sign digit is counted into the sequence register, and an absolute or relative transfer may follow or not according to the sign. Multiconditional transfers can be made in a single count.

For instance, in terms of the code numbers of Table I, the commands

$$n;\ 3,\ 7,$$
$$n;\ 3,\ 8,$$

respectively, replace the content of the sequence register by "n" and add to it; whilst

$$0;\ 10,\ 6$$

adds a unit to the sequence register if the content of register A is negative.

The use of standard sub-routines as aids to building complete programmes is adopted. Most sub-routines can, with the aid of the properties of the sequence register and the multiplicity of adding registers in the arithmetical unit, be made independent of their actual position in the store. This simplifies programming. Transfers to and from sub-routines to the master programme are performed simply by use of the last word position of the D register, numbered 15. The content of the sequence register is stored there immediately before the transfer to a sub-routine headed, say, at position m. One of the commands of the sub-routine adds 1 to this address, and the last command transfers the content of D_{15} to the sequence register, which in turn calls for a transfer of control to the next command in the master programme. This method has the advantage of not involving the main arithmetical registers A, B, and C during transfers. The commands needed are shown in Table II.

Input and Output. At present data are inserted via punched cards of the IBM type. These are read in a columnar fashion, ten binary digits being

TABLE II

Command Position	Command	Significance
$n-2$	15; 5, 20	Stores $n-1$ in D_{15}
$n-1$	m; 3, 7	Transfers to "m"
m	—	Head of sub-routine
—	—	—
$m+r-1$	15; 22, 21	Adds unit to D_{15}
$m+r$	15; 18, 7	Last command of sub-routine; transfers to "n"
n	—	Continues programme

entered from each column. A primary routine of twenty commands is entered, prior to the feeding of cards, through a group of permanently wired stepping switches. This routine supplies sufficient information to the computer to enable it to assemble 10-digit data into 20-digit words.

The card reader is also capable of reading decimal data, punched in the usual decimal manner, and changes from binary to decimal reading and back are by the computer.

Most programme data are supplied in binary form being mostly of subroutine type, and little extra effort is required to punch the cards of the master programme in binary form.

The X and Y punch positions are used for reading control; in particular a Y punch is used to adjust commands as they enter the computer in order to allow for data which vary with the problem and so on.

Output is obtained in two ways: by standard teletype page printing and by decimal card punching, again in columnar fashion. Punching is so arranged that results may be re-inserted later into the computer as problem data if required.

T. PEARCEY

Division of Radiophysics
Commonwealth Scientific and Industrial
Research Organization
Australia

DISCUSSIONS

Minimum Access Programming

The following discussion outlines programming techniques which may be applied to reduce computation time on certain large scale general purpose computers. The computers to which these techniques can be applied most readily have memory systems of large capacity, storing commands and operands in an alterable medium. These techniques have been applied to great advantage in the ERA 1101, a computer using a magnetic drum storage system. On the 1101 the techniques have given an advantage of 20 to 1 in computing speed over ordinary induction loop techniques for the average program, with an advantage of 100 to 1 over short stretches.

The internal memory of a typical computer under discussion is divided into compartments called memory boxes, each memory box having an address. The computer performs operations such as addition and subtraction on operands contained in memory boxes. The content of a memory box is called a "word." The word may either be an operand or a command.[1] If the word is a command, it will consist of a command code and one to four execution addresses.

The addresses of the memory boxes have a cyclic order. Access to a memory box is made possible when a coincidence is detected between a locating register and a storage address register. The storage address register contains the address of the memory box to which access is desired. As each memory box is scanned its address is held in the locating register. The addresses of all the memory boxes pass through the locating register in their cyclic order every memory cycle.

Let an address be sent to the storage address register. The interval between the time that the coincidence detector is turned on and the time

that a coincidence is detected depends on the distance between the memory box containing the instruction and the memory box containing the operand.

For each command of a computer there is a minimum interval between the location of the command and its operand or operands, such that if a different interval is programmed the effect would be a loss of operating time. Attention to these minimum (coding) intervals in construction of a program is called minimum access programming.

In order to examine minimum access techniques more closely, those used on a specific machine, the ERA 1101, will be discussed.

The ERA 1101 is a high-speed electronic, general purpose digital computer, having a large internal memory. The computer operates on 24 binary digits (bits) and uses the one address system of logic. The internal memory consists of a magnetic drum[2] of 16,384 memory boxes. A portion of the drum in which a digit is stored is called a memory cell and a portion of the drum in which a word is stored is called a memory box. For each memory box there are 24 memory cells. All of the digits of a word are read from or written in a memory box in parallel. Arithmetic operations treat numbers up to $\pm 2^{47} - 1$ in a double length accumulator.

The drum is subdivided into eight groups along its length, so that for each angular position of the drum, any one of eight memory boxes is available for access. Each group, by itself, contains 2,048 memory boxes.

Let a memory box on the drum be designated by the address $a_{13}a_{12} \cdots a_2a_1a_0$ held by locating registers, where the a's are binary digits. Then the digits $a_{11}a_{10}a_9$ in the locating registers denote the group containing the memory box, the digits $a_{13}a_{12}$ denote the quadrant containing the memory box, and the remaining digits denote the angular position of the memory box within the quadrant.

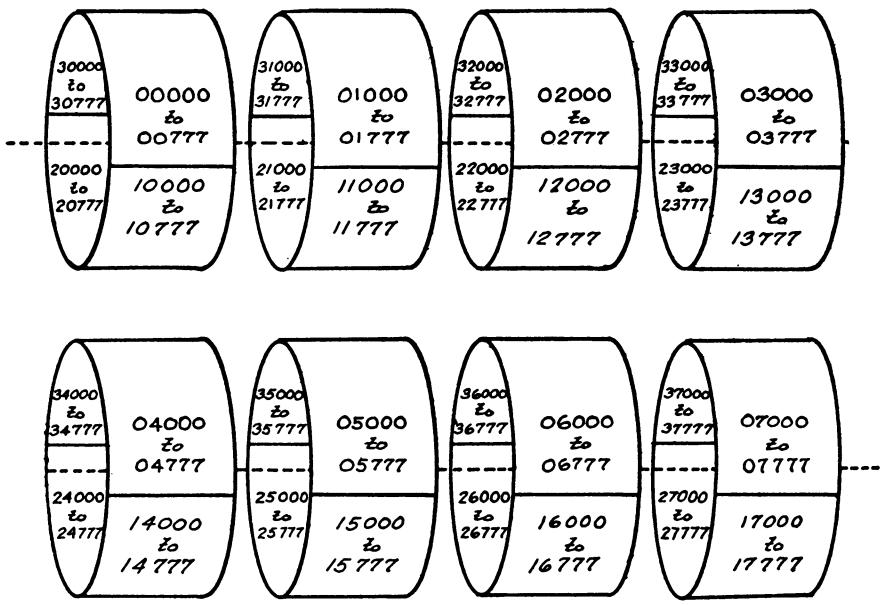


FIG. 1. Address mapping on drum for 1 interlace.

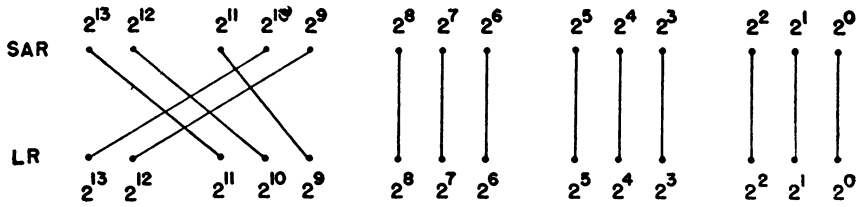


FIG. 2. Patching connections for 1B interlace.

In order to select memory locations around the drum so that they are associated with consecutive memory boxes, the storage address register (SAR) is connected to the locating register (LR) through an interchangeable plugboard called an interlace chassis.

If the digits of the SAR are carried into their corresponding positions in the LR, then consecutive addresses are located in consecutive memory boxes. This is called a one interlace.

For adjacent memory boxes, with any interlace, the difference between their addresses is called the increment. The one interlace has an increment of one.

One may see that there are 14! ways to interlace the drum with the storage address register, although only a few of these have been useful. Those used most frequently are the 1, 2, 4, 8, 16, 32, 64, 128 and 1B interlace. The special interlace hook-up called the 1B interlace has been found useful for loading the drum. The interlace may be dialed.

With a 64 interlace, memory boxes having consecutive addresses are 64 cell periods apart. One cell period is the interval between two adjacent memory boxes on the drum and is equal to 8 microseconds. Also for a 64 interlace we have an increment of 32.

Starting with address zero, adjacent memory boxes have the following addresses for 64 interlace. Octal notation is used.

X is the group index:

$0X000$
$0X040$
$0X100$
$0X140$

$0X740$
$1X000$
$1X040$
$1X100$

$1X700$
$1X740$
$2X000$
$2X040$
$2X100$
---
etc.

Each command consists of a six bit command code, $a_{23}, \cdots, a_{18}$, four skip bits $a_{17}, \cdots, a_{14}$ and a fourteen bit execution address $a_{13}, \cdots, a_0$.

A 1 appearing in any one of the skip bit positions of a command can be used to advance the program address counter by pulsing one of its first ten stages. Four selector switches determine which stages are to be pulsed by the skip bits should they appear.

Delays inserted in the skip bit lines allow multiple skip bits to be used with the same command.

One skip bit setting used by the author carries

$$a_{15} \text{ into } 2^0, \qquad a_{16} \text{ into } 2^1, \qquad \text{and} \qquad a_{17} \text{ into } 2^2$$

with the selector switch for a_{14} turned off.

Then if a command appears in box y with 1's in a_{15}, a_{16} and a_{17}, the next command will appear in address $y + 2^0 + 2^1 + 2^2 + 1$.

If an interlace of one were used, with no skips, at least one drum revolution would be required for each instruction. By choosing the proper interlace and skips, however, operands and instructions may be so placed that several instructions are executed in one drum revolution.

In order to coordinate the philosophy of minimum access programming with techniques used in the ERA 1101, a discussion of the 1101 commands is in order.

The list given in Table I gives the commands and their corresponding minimum allowable access times. In the table interpret "y" as the address of the memory box containing the word "(y)." The symbol "$\rightarrow$" is interpreted as "goes to." AL is the left half or higher order accumulator and AR is the right half or lower order accumulator. A is a 48-bit electronic register and Q is a 24-bit electronic register. Negative numbers are expressed by complements.

The intervals A, B, C and D specified in Table I are minimum values. Should a shorter interval be programmed, the only effect is a loss of operating time. The interval is then automatically lengthened by the duration of a complete drum revolution.

The intervals of commands 17, 27, 35 and 36 are valid for a *single* writing operation. The minimum allowable interval between *two* successive writing operations in the same group is 256 cell periods. In different groups, the minimum interval is 4 cell periods. Note that a reading operation will normally occur between successive writing operations.

The minimum interval for either C or D of commands 25 and 26 is approximately $6 + 0.33k$ cell periods where k is the number of places shifted.

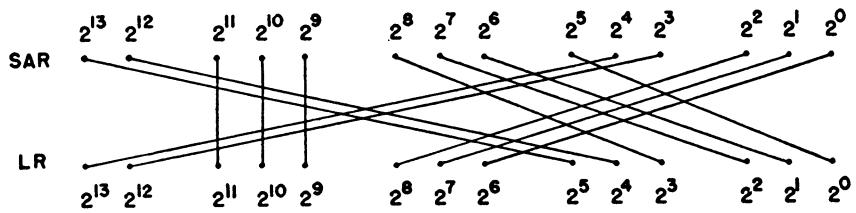
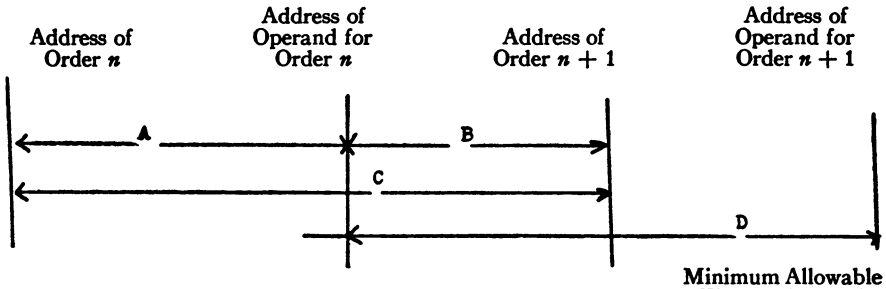


FIG. 3. Patching connections for a 64 interlace.

TABLE I. List of commands and minimum time intervals for coding



Symbol	Command Code	Command in Order	Minimum Allowable Time in Interval in Cell Periods			
			A	*B*	*C*	*D*
$y \rightarrow cA^+$	11	*Clear Add:* Insert (y) in A.	4	4	—	—
$y \rightarrow hA^+$	12	*Hold Add:* Add (y) to (A).	4	4	—	—
$y \rightarrow cA^-$	13	*Clear Subtract:* Insert the complement of (y) in A.	4	4	—	—
$y \rightarrow hA^-$	14	*Hold Subtract:* Add the complement of (y) to (A).	4	4	—	—
$y \rightarrow Q$	16	*Load Q:* Insert (y) in Q.	4	4	—	—
$Q \wedge A \rightarrow y$	17	*Substitute Digits:* Insert the lower half of (A) into y where there are ones in Q and block any change in y where there are zeros in Q.	4	8	—	—
$y \rightarrow mcA^+$	21	*Absolute Clear Add:* Insert the absolute value of (y) to A.	4	4	—	—
$y \rightarrow mhA^+$	22	*Absolute Hold Add:* Add the absolute value of (y) to (A).	4	4	—	—
$y \rightarrow mcA^-$	23	*Absolute Clear Subtract:* Insert the complement of the absolute value of (y) in A.	4	4	—	—
$y \rightarrow mhA^-$	24	*Absolute Hold Subtract:* Add the complement of the absolute value of (y) to (A).	4	4	—	—
ALk	25	*Shift A Left:* Shift A circularly, the number of places designated by the number in the execution address part of the instruction.	—	—	—	—
QLk	26	*Shift Q Left:* Shift Q circularly, the number of places designated by the number in the execution address part of the instruction.	—	—	—	—
$Ap \rightarrow y$	27	*Substitute Execution Address:* Insert (AR) into y blocking any change in the upper six binary digits of (y).	4	8	—	—
$y \rightarrow scA^+$	31	*Split Clear Add:* Insert (y) in A multiple precision.	4	4	—	—
$y \rightarrow shA^+$	32	*Split Hold Add:* Add (y) to A multiple precision.	4	4	—	—
$y \rightarrow scA^-$	33	*Split Clear Subtract:* Insert the complement of (y) in A multiple precision.	4	4	—	—
$y \rightarrow shA^-$	34	*Split Hold Subtract:* Add the complement of (y) to (A) multiple precision.	4	4	—	—

TABLE I—*Continued*

Symbol	Command Code	Command in Order	Minimum Allowable Time in Interval in Cell Periods			
$A \rightarrow y$	35	*Store A:* Insert (AR) into y.	4	8	—	—
$Q \rightarrow y$	36	*Store Q:* Insert (Q) into y.	4	8	—	—
zero $\rightarrow AR$	37	*Clear AR:* Clear the lower order accumulator.	—	—	5	—
$Q \rightarrow cA$	41	*Clear Add from Q:* Insert (Q) in A.	—	—	5	—
$Q \rightarrow hA$	42	*Hold Add from Q:* Add (Q) to (A).	—	—	5	—
$A \rightarrow cQ$	43	*Transmit A to Q:* Insert (AR) in Q.	—	—	5	—
QJ	44	*Q Jump:* If (Q) is negative, insert (y) in the SAR. In either case shift Q circularly one place to the left.	—	—	5	—
J	45	*Jump:* Insert y in the SAR.	—	—	5	—
cJ	46	*Sign-Conditional Jump:* If (A) is negative, insert y in the SAR.	—	—	5	—
zero J	47	*Zero-Conditional Jump:* If (A) is zero insert y in the SAR. In either case subtract one from A.	—	—	5	—
$Qy \rightarrow cA$	51	*Clear Logical Multiply:* Insert (y) in AR where there are ones in Q, blocking a change in AR where there are zeros in Q.	4	4	—	—
$Qy \rightarrow hA$	52	*Hold Logical Multiply:* Add (y) to (AR) where there are ones in Q, blocking a change in AR where there are zeros in Q.	4	4	—	—
$y \rightarrow P$	53	*Print Only:* Print the six bit character whose address is y.	4	4	—	—
$y \rightarrow P/P$	54	*Print and Punch:* Print and punch the six bit character whose address is y.	4	4	—	—
IS	55	*Intermediate Stop:* Stop the computer and turn on the intermediate stop light. Proceed only after the operator pushes the start button.	—	—	—	—
OS	56	*Optional Stop:* Stop the computer if the optional stop button has been pushed, and give suitable indication.	—	—	—	—
FS	57	*Final Stop:* Stop the computation and turn on the final stop light.	—	—	—	—
$Q \cdot y \rightarrow cA$	61	*Clear Multiply:* Obtain $(y) \cdot (Q)$ in A.	4	4	—	37
$Q \cdot y \rightarrow hA$	62	*Hold Multiply:* Add $(y) \cdot (Q)$ to (A).	4	13	—	45
$A/y \rightarrow cQ$	63	*Divide:* Obtain $(A)/(y)$. Find the quotient in Q and the non-negative remainder in A.	4	4	—	44
	65	*Optional Jump:* Same as optional stop except when the computer is on TEST. If on TEST, and the optional jump switch is on, then this command is the same as the jump command.	—	—	5	—
	66	*Do Nothing:* Performs no operation.	—	—	5	—
$y + 1 \rightarrow A$	71	*Clear Add Plus One:* Insert $(y)+1$ in A.	4	4	—	—

The intervals of commands 53 and 54 are for a single print operation. The minimum allowable interval between two successive print operations is approximately 19,000 cell periods or 150,000 microseconds. Should a shorter interval be programmed it will automatically be lengthened to this interval. Note, however, that while a print command is being executed, the computer can proceed with the program of instructions provided the following orders do not call for another print operation.

Although interval B of command 62 is indicated as being a minimum of 13 cell periods, a 12 cell period may be used for B when interval A is programmed at no more than 4 cell periods. Under this condition, interval C is therefore 16 cell periods.

The 66 command performs no operation but its skip bits may be used to augment those of the previous command.

In writing a minimum access program, the programmer sometimes takes advantage of induction loop programming methods. A case in point is the matrix multiplication of two 10×10 matrices. The solution of the problem gives $X \cdot Y = Z$, where X, Y and Z are 10×10 matrices.

A portion of the program of a matrix multiplication is shown in Table II. Skips are not used as the program was written before they were made available. In preparation for the execution of this program, elements of X are inserted row by row in memory boxes whose addresses are designated by the following table of addresses at 128 interlace in group ϕ. Octal notation is used.

401	402	403	404	405	406	407	410	411	412
421	422	423	424	425	426	427	430	431	432
441	442	443	444	445	446	447	450	451	452
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
621	622	623	624	625	626	627	630	631	632

Table of Addresses

The elements of Y transpose are inserted row by row in memory boxes whose addresses are designated by the table of addresses at 128 interlace, but in group 1.

Z is written in a similar block, where the elements of the first row are in consecutive addresses at 64 interlace.

The program of Table II obtains the products of the rows of X and a column of Y in 30 revolutions of the drum, and the entire matrix product has been clocked at 8.9 seconds where no overflow occurred during the multiplication. If a z_{ij} occurs larger than $2^{23} - 1$, then $2^{23} - 1$ is subtracted from z_{ij} and a conditional jump is executed.

The scaling routine which starts at address 30540 does not utilize minimum access techniques. This is an ordinary induction loop routine. A minimum access attack may be made on the scaling routine at the expense of memory space.

It is possible to manufacture portions of the minimum access program in the computer. This manufactured program may be punched out ready for use.

From the program of Table II and from other material presented above, it may be concluded that: Arranging operands and instructions so that they

TABLE II. Matrix multiplication on 64 interlace of two 10 × 10 matrices

Insert the following commands at 1 B interlace (Octal notation is used):

Memory Box	Command				Explanation
03120	11	03	10	31	3120 at 1 B interlace becomes 10031 at 64 interlace.
03121	11	03	10	71	
03122	11	03	11	31	
03123	11	03	11	71	
.	.	.	.	.	The operation starts with the command in memory box 10031.
.	.	.	.	.	
03131	11	03	14	71	
03220	27	03	00	32	3220 at 1 B interlace becomes 10032 at 64 interlace.
03221	27	03	00	72	
.	.	.	.	.	
.	.	.	.	.	
03231	27	03	04	72	
03320	45	01	00	71	3220 at 1 B interlace becomes 10033 at 64 interlace.
03321	45	01	01	31	
.	.	.	.	.	
.	.	.	.	.	
03330	45	01	04	71	
03331	45	03	05	00	

At 64 interlace:

30500	11	03	05	36	
30501	35	03	05	37	
30502	11	03	05	35	
30503	27	03	05	07	
30504	11	03	05	07	
30505	12	03	05	34	
30506	27	03	05	07	
30507	45	00	00	00	
30534	00	00	00	40	
30535	00	03	07	41	
30536	77	77	77	65	Minus ten counter.
30537	00	00	00	00	
31001	11	01	10	02	
31002	35	01	30	03	
31003	11	01	10	04	
31004	35	01	40	05	
.	.	.	.	.	
.	.	.	.	.	
31023	11	01	10	24	
31024	35	01	70	25	
31025	45	03	00	00	
31031	00	01	20	00	
31037	00	00	00	02	

Subroutines starting in memory boxes 31041, 31101, 31141, 31201, 31241, 31301, 31341, and 31401 have been omitted to conserve space. Their function is similar to the subroutines of 31001 to 31025 and 31441 to 31465.

31441	11	01	14	42
31442	35	01	30	03
31443	11	01	14	44
31444	35	01	40	05
.	.	.	.	.
.	.	.	.	.
31464	35	01	70	25
31465	45	03	00	00
31471	00	01	24	40

TABLE II—*Continued*

Memory Box	Command				Explanation
30000	11	01	00	01	zero A
30001	16	01	00	02	$x_{11} \rightarrow Q$
30002	62	01	30	03	$Q \cdot y_{1j} \rightarrow hA$
30003	16	01	00	04	$x_{12} \rightarrow Q$
30004	62	01	40	05	$Q \cdot y_{2j} \rightarrow hA$
.	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	
30023	16	01	00	24	$x_{1,10} \rightarrow Q$
30024	62	01	70	25	$Q \cdot y_{10j} \rightarrow hA$
30025	46	03	20	26	
30026	14	01	20	27	$37777777 \rightarrow hA^-$
30027	46	03	00	31	
30030	45	03	05	40	
30031	12	01	20	32	$37777777 \rightarrow hA^+$
30032	35	00	00	00	
30033	45	03	00	40	

The subroutines starting in memory boxes 30040, 30100, 30140, 30200, 30240, 30300, 30340, and 30400 are similar to the subroutines of 30000 to 30033 and 30440 to 30472.

30440	11	01	00	01	zero $\rightarrow cA^+$
30441	16	01	04	42	$x_{10,1} \rightarrow Q$
30442	62	01	30	03	$Q \cdot y_{1j} \rightarrow hA$
30443	16	01	04	44	$x_{10,2} \rightarrow Q$
30444	62	01	40	05	$Q \cdot y_{2j} \rightarrow hA$
.	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	
30463	16	01	04	64	$x_{10,10} \rightarrow Q$
30464	62	01	70	25	$Q \cdot y_{10j} \rightarrow hA$
30465	46	03	24	66	
30466	14	01	20	27	$37777777 \rightarrow hA$
30467	46	03	04	71	
30470	45	03	05	40	
30471	12	01	20	32	
30472	35	00	00	00	
30473	71	03	05	37	
30474	35	03	05	37	
30475	46	01	30	30	
30476	57	00	00	00	

At 1 B interlace:

12660	12	01	20	27	12660 at 1 B interlace is
12661	12	01	20	27	32026 at 64 interlace.
.	.	.	.	.	
.	.	.	.	.	
12671	12	01	20	27	
12760	46	03	06	00	12760 at 1 B interlace is
12761	46	03	06	00	32027 at 64 interlace.
.	.	.	.	.	
.	.	.	.	.	
12771	46	03	06	00	
13060	14	01	20	31	13060 at 1 B interlace is
13061	14	01	20	31	32030 at 64 interlace.
.	.	.	.	.	
.	.	.	.	.	
13071	14	01	20	31	
30600	25	00	00	31	
30601	43	00	00	00	
30602	11	03	05	75	
30603	35	03	05	74	
30604	44	03	06	06	

TABLE II—*Continued*

Memory Box	Command				Explanation
30605	45	03	05	50	
30606	71	03	05	74	
30607	35	03	05	74	
30610	45	03	06	04	
30636	00	01	10	02	
30637	00	00	00	30	
At 1 B interlace:					
17120	12	03	17	31	One → hA^+
17121	12	03	17	71	40 → hA^+
.	.	.	.	.	
.	.	.	.	.	17120 at 1 B interlace is
17131	12	03	17	71	13031 at 64 interlace.
17220	35	03	00	32	17220 at 1 B interlace is
17221	35	03	00	72	13032 at 64 interlace.
17222	35	03	01	32	
17223	35	03	01	72	
.	.	.	.	.	
.	.	.	.	.	
17231	35	03	04	72	
17320	45	01	30	71	17320 at 1 B interlace is
17321	45	01	31	31	13033 at 64 interlace.
17322	45	01	31	71	
17323	45	01	32	31	
.	.	.	.	.	
.	.	.	.	.	
17330	45	01	34	71	
17331	45	03	05	04	
At 64 interlace:					
31731	00	00	00	01	
31771	00	00	00	40	
13030	11	03	00	32	
13160	45	03	00	32	13160 at 1 B interlace is
13161	45	03	00	72	32031 at 64 interlace.
13162	45	03	01	32	
13163	45	03	01	72	
.	.	.	.	.	
.	.	.	.	.	
13170	45	03	04	32	
13171	45	03	04	72	
At 64 interlace:					
12027	37	77	77	77	
12030	37	77	77	77	
12031	37	77	77	77	
12032	37	77	77	77	
30540	25	00	00	31	This is the scaling routine.
30541	43	00	00	00	
30542	11	03	05	75	
30543	35	03	05	74	
30544	44	03	05	50	
30545	71	03	05	74	
30546	35	03	05	74	
30547	45	03	05	44	
30550	13	03	05	74	
30551	27	03	05	60	
30552	14	03	06	77	
30553	12	03	05	73	

TABLE II—*Continued*

Memory Box	Command				Explanation
30554	35	03	05	73	
30555	11	03	06	76	
30556	27	03	05	60	
30557	27	03	05	62	
30560	11	00	00	00	
30561	25	00	00	00	
30562	35	00	00	00	
30563	11	03	05	57	
30564	12	03	10	37	
30565	27	03	05	60	
30566	27	03	05	62	
30567	14	03	05	72	
30570	46	03	05	60	
30571	45	01	00	31	
30572	16	01	14	64	
30573	00	00	00	00	
30574	00	00	00	00	
30575	77	77	77	17	
30576	77	77	76	33	−ten[2]
30577	00	00	00	00	

occur as close as possible to the minimum allowable coding intervals shortens the computation time considerably. In the program of Table II the method used to obtain a product of a row by a column is faster than an ordinary induction loop program by a factor of approximately 50. A 10 × 10 matrix multiplication may be performed in approximately 8 seconds using minimum access techniques.

[1] In the UNIVAC a word contains two commands.
[2] A. A. COHEN, "Magnetic drum storage for digital information processing systems," *MTAC*, v. 4, p. 31–39.

Engineering Research Associates
Arlington, Virginia

DAVID P. PERRY

BIBLIOGRAPHY Z–XX

1. WILLIAM A. ALLEN & RICHARD H. STARK, "Ray tracing using the IBM Card Programmed Electronic Calculator," Optical Society of America, *Jn.*, v. 41, p. 636–640.

The IBM Card Programmed Electronic Calculator is described, and the manner in which the ray tracing equations are modified in order to adapt them especially for this calculator is outlined. Equations are given and procedure explained for tracing meridional and skew rays through a spherical lens system. Numerical examples are given for both a meridional and a skew ray trace, and the accuracy of the machine calculations is discussed.

ETHEL MARDEN
NBSCL

2. FRANZ L. ALT, "Evaluation of automatic computing machines," *Prod. Eng.*, v. 22, Nov. 1951, p. 146–152.

The author explains the why and wherefore of large scale automatic computing equipment and predicts the further growth of such equipment as engineering tools. The essential features of digital machines are described and contrasted with those of analogue machines and some advantages and

disadvantages of the two types are enumerated. A summary table of vital statistics is included on three computers in operation at present (ENIAC, SSEC, and SEAC), and three computers under test as of the early part of 1951 (Mark III, UNIVAC, and ERA). Despite the advances already achieved with the aid of such machines in the fields of engineering, mathematics, physics, industrial management and control, military science, etc., according to the author we have only begun to scratch the surface of the vast regions now capable of being investigated.

J. H. LEVIN

NBSCL

3. ANON., "Office robots," *Fortune*, v. 45, Jan. 1952, p. 82–87.

This article gives an overall account of the history of "electronic brains" from the development of ENIAC in 1946 to the present time. A few of the better known machines are discussed from those of the ninety organizations throughout the country working on some form of computer. A few diagrams are shown to illustrate the very rudiments of how a high-speed computer computes. Examples are given of the savings attained for some companies employing these machines. Several examples of special purpose machines are discussed, such as one developed by the Bell Telephone Laboratories, called the Automatic Message Accounting Machine, which automatically records and accounts toll and local calls and makes out customers' bills.

DONALD O. LARSON

NBSCL

4. A. A. AUERBACH, J. P. ECKERT, JR., R. F. SHAW, J. R. WEINER, & L. D. WILSON, "The BINAC," IRE, *Proc.*, v. 40, Jan. 1952, p. 12–29.

The logical design, instruction code, and some typical circuitry of the Binary Automatic Computer, BINAC, are all described here, though necessarily quite concisely. Such broad coverage in one article should appeal to a newcomer to the field, although it requires very thoughtful reading. The expert will certainly be interested because BINAC was the first of its genus of high-speed, serial, digital computers to be completed in this country, its pulse repetition rate, 4 mc/s, is still the highest in use, and BINAC can be a relatively perspicuous introduction to the much more complex UNIVAC's. The serious student of BINAC's and UNIVAC's circuitry should notice three basic changes made in the UNIVAC's:

1) The pulse repetition frequency was reduced from 4 to 2.25 mc/s.

2) In the pulse reforming circuit a flipflop was substituted for the Guillemin line.

3) In the mercury delay line memory simple impulse excitation was replaced by gating a carrier wave.

R. D. ELBOURN

NBSEC

5. F. P. COZZONE, "Organizing a computer center in the engineering department," *Prod. Eng.*, v. 23, Jan. 1952, p. 136–141.

The author stresses the importance of computing facilities in engineering analysis. A brief outline is given of some types of digital and analogue

computers as well as some varieties of auxiliary data recording and processing equipment. A table is included classifying five computers according to their adaptability to different types of aircraft design problems. Finally some suggestions are given on the organization of a sample engineering computing facility. This article is carelessly written and contains a number of errors of fact. For example, the IBM Defense Calculator is identified with the SSEC in Fig. 2 and with the Card Programmed Calculator in Table III.

J. H. LEVIN

NBSCL

6. J. W. DONNELL, "If computation costs too much," *Chemical Engineering*, v. 58, Dec. 1951, p. 138–141.

This article points to high-speed mechanical and electronic aids to calculation for savings in money and man power for computational problems arising in engineering. Ways in which engineering offices can make the most effective use of high-speed computing machinery are given. Several examples are described illustrating this point.

DONALD O. LARSON

NBSCL

7. DONALD P. FEDER, "Optical calculations with automatic computing machinery," Optical Society of America, *Jn.*, v. 41, Sept. 1951, p. 630–635.

This paper describes the procedures employed for tracing rays through an optical lens system, using automatic computing machinery, namely, the IBM Card Programmed Electronic Calculator and the SEAC. There are two main phases of the computation: finding the point of incidence of a ray with the following surface, and computing the direction cosines of the refracted ray. It is shown how the process may be modified to apply to aspheric surfaces. A process is outlined for use on the CPEC for the computations of first and third order imagery and for the extension of this calculation to aspheric surfaces.

ETHEL MARDEN

NBSCL

8. H. H. GOODE, "Simulation—its place in system design," IRE, *Proc.*, v. 39, Dec. 1951, p. 1501–1506.

The uses of both analogue and digital computers as aids in the task of designing large complex systems are discussed in broad general terms. Careful study of each problem rather than adherence to any universal rule is urged.

R. D. ELBOURN

NBSEC

9. H. J. GRAY, JR., "Logical description of some digital-computer adders and counters," IRE, *Proc.*, v. 40, Jan. 1952, p. 29–33.

A set of logical block diagram symbols and ordinary (non-Boolean) algebra are used to describe a few binary adder-subtractors and counters.

R. D. ELBOURN

NBSEC

10. B. KAZAN & M. KNOLL, "Fundamental processes in charge-controlled storage tubes," RCA, *Review*, v. 12, Dec. 1951, p. 702–753.

This paper is a review of a large, important, and delicate subject. The subject is large because it includes secondary emission, photo emission, photo conduction, and electron-induced conduction. The importance of these processes lies in the civilian and military uses of the tubes for which they are the heart. Some idea of the possible extent of these applications can be gained by realizing that the term covers not only all the many types of storage tubes for computer use, but also includes all television "pick-up" tubes (such as the Image-Orthicon), direct viewing tubes (such as the "Snooperscope") and all types of signal converter tubes. The subject is delicate in two ways: first, the processes themselves are very delicate, varying drastically upon very slight provocation; secondly, so few of the many attempts to produce useful storage tubes have been successful that many of the potential users of such tubes have become discouraged.

For inventors, the field of charge-controlled storage tubes is wide open, and the authors have written the primer on the subject. Part I tells how the basic processes behave under the action of light and electron bombardment; Part 2 gives definitions of terms used in the business; Part 3 tells about storage tube reading and writing; and Part 4 gives 97 bibliography references. A great many of these references, particularly those dealing with the basic processes, are written in German, but the authors have done us an extra service by giving a summary of each paper in English. The list is reasonably complete, although no attempt was made to catalogue the recent spate of papers which have been unraveling the theory of the Williams method of cathode ray tube storage. The authors of this paper have done a real service in bringing together so much of the information on storage tubes.

ARTHUR W. HOLT

NBSEC

11. A. J. LEPHAKIS, "An electrostatic-tube storage system," IRE, *Proc.*, v. 39, November 1951, p. 1413–1415.

A binary storage system is described which should prove useful in studying certain communication problems. The system comprises two channels, each using an MIT electrostatic storage tube, with provision for writing incoming pulses in one channel while recovering previously stored pulses from the other. The order of incoming pulses is preserved during storage, but the time rate of recovery may be faster or slower than the original writing rate. A block diagram and two typical circuits are shown. Performance details are discussed.

WILLIAM W. DAVIS

NBSEC

12. OFFICE OF NAVAL RESEARCH, *Digital Computer Newsletter*, v. 4, April 1952, 6 pages.

The contents are as follows:

1. Whirlwind I.
2. The ONR Relay Computer.
3. The ABC.
4. The SEAC.

5. The SWAC.
6. The Circle Computer.
7. Moore School Automatic Computer (MSAC).
8. The UNIVAC.
9. The Jacobs Instrument Company Computers (JAINCOMP).
10. The CADAC.
11. Consolidated Electronic Digital Computer Model 30–201.
12. The ACE Pilot Model.
13. Data-Handling Devices.

13. H. RUTISHAUSER, A. SPEISER, & E. STIEFEL, "Programmgesteuerte digitale Rechengerate," Inst. f. angew. Math., *Mitt.* No. 2, 1951, 102 pages.

This is a comprehensive exposition of the logical organization of automatic digital computers and of the principles of their physical realization. It is the first publication of its kind in German and far better than most similar writings in English. Concisely and precisely written, complete, and as unbiased as is possible in this controversial field, it is highly recommendable as an introduction to the subject.

The first chapter deals briefly with the why and how of automatic computation, its history, relations to formal logic and neurophysiology, and (perhaps too briefly) applications. A second chapter lists and describes, in only four pages, the major components of automatic computers. The third chapter explains in full detail the various methods for representing numbers (binary, coded decimal; fixed vs. floating point) and for carrying out the elementary arithmetic operations. There follows a chapter on coding and programming, including such topics as flow diagrams, branch points, single and multiple address codes, modification of instructions by arithmetic operations, and a completely carried out example (multiplication of two matrices). Methods of evaluating mathematical functions as well as checks, both built-in and programmed, are also discussed. The final chapter describes the basic circuits occurring in computers, such as "and" gates, "or" gates, adders, etc.; the several methods of number storage (acoustic, magnetic, electrostatic); and media for input and output. There is a summary table of computer developments as of December 1949 and a comprehensive bibliography.

FRANZ L. ALT

NBSCL

NEWS

Association for Computing Machinery.—The spring meeting of the Association was held on May 2 and 3, 1952, at the Mellon Institute, Pittsburgh, Pennsylvania. On May 2nd at 7:00 p.m. a banquet was held at which time a talk, "History of mechanical computing machinery," was presented by GEORGE C. CHASE, Monroe Calculating Machine Company, Orange, N. J. The program for the meeting was as follows:

May 2, 1952, 10:00 a.m.

General Session	FRANZ L. ALT, NBSCL and President of ACM, *Chairman*
Address of Welcome	E. R. WEIDLEIN, President, Mellon Institute
Evaluation of Automatic Computing	IRVEN TRAVIS, Burroughs Adding Machine Co.

Transistor Physics	W. SHOCKLEY, Bell Telephone Laboratories
Mark IV	H. H. AIKEN, Harvard Computation Laboratory

May 2, 1952, 2:30 to 5:00 p.m.

Computers and Components I	JAN. A. RAJCHMAN, RCA. *Chairman*
Special-Purpose Digital Data-Processing Computers	B. M. GORDON and R. N. NICOLA, Laboratory for Electronics, Inc.
The Remington-Rand Punch-Card Electronic Computer, Type 409–2	L. P. CROSMAN, Remington-Rand Corp.
The Elecom 100 General Purpose Computer	ALBERT AUERBACH, Electronic Computer Corporation
The Quadratic Arc Computer (QVAC)	M. J. MENDELSON, Northrup Aircraft, Inc.
A System for Counting and Recording Electrical Impulses in Printed Decimal Form	J. L. LINDESMITH, Clary Multiplier Corp.
The Logical Organization of the New IBM Scientific Calculator	N. ROCHESTER, IBM Corp.

May 2, 1952, 2:30 to 5:00 p.m.

Mathematical Applications I	MINA REES, ONR, *Chairman*
Engineering Problems Requiring Automatic Computing Facilities	E. L. HARDER, Westinghouse Electric Corp.
Digital Computer Methods for Problems which Involve Linear Inequalities	ALEX ORDEN, Hq. U.S.A.F.
Computational Problems of Linear Programming	A. CHARNES, Carnegie Institute of Technology
Small Problems on a Large Computer	C. W. ADAMS, MIT
Firing Table Computations on the ENIAC	H. L. REED, JR., Aberdeen Proving Ground
Small-Scale Research and Automatic Computing	E. C. BERKELEY, Edmund C. Berkeley & Associates

May 2, 1952, 2:30 to 5:00 p.m.

Theory I—Information and Control	J. H. CURTISS, NBS, *Chairman*
Use of Computing Machinery in Applications of Information Theory	W. G. TULLER, Melpar, Inc.
An Upper Bound on the Informational Capacity of a Synapse	W. S. MCCULLOUGH, Neuropsychiatric Institute, DONALD MACKAY, Kings College, London
Automatic Control of Machinery	P. L. NIES, G. P. TANQUARY, D. R. AUFDERHIEDE, D. W. BROWN, C. J. JACOBY, R. B. KELLER, Harvard Univ.
The Maze Solving Computer	R. A. WALLACE, L. and O. Research and Development Co.
A Chart Method for Simplifying Truth Functions	E. W. VEITCH, Burroughs Adding Machine Co.

May 3, 1952, 9:30 to 12:00 Noon

Computers and Components II	JOHN D. DILLON, Air Force Missile Test Center, Cocoa, Florida, *Chairman*
Transistor Circuits for Computers	J. H. FELKER, Bell Telephone Laboratories
Standardized Printed Circuit Units for Digital Computers	D. L. JOHNSON, Elliott Brothers, Ltd., Borehamwood, England
Non-Linear Switching Elements	B. MOFFAT, F. A. SCHWERTZ, Mellon Institute, B. O. MARSHALL, Air Force Cambridge Research Center

Optical Elements for Computers	B. O. MARSHALL, Air Force Cambridge Research Center, J. R. BOWMAN, F. A. SCHWERTZ, Mellon Institute
The Selenium Rectifier—A Non-Linear and Asymmetric Resistance Element	N. HARDY, International Resistance Co.

May 3, 1952, 9:30 to 12:00 Noon

Mathematical Applications II	ALSTON S. HOUSEHOLDER, Oak Ridge National Laboratory, *Chairman*
Discussion on the Use and Construction of Subroutines	JAMES ALEXANDER, Argonne National Laboratory, HERMAN H. GOLDSTINE, IAS, JOSEPH H. LEVIN, NBS, H. RUBENSTEIN and J. D. RUTLEDGE, Remington-Rand, Inc.

May 3, 1952, 9:30 to 12:00 Noon

Theory II—Mathematical	C. V. L. SMITH, ONR, *Chairman*
A Unified Approach to the Monte Carlo Method	J. H. CURTISS, NBS
The Solution of Boundary Value Problems by the Method of Kernel Functions	STEFAN BERGMAN, Stanford Univ.
Boundary Value Problems in Doubly Connected Domains	FRANZ L. ALT, NBS

May 3, 1952, 2:00 to 5:00 p.m.

Computers and Components III	L. P. CROSMAN, Remington-Rand Corporation, *Chairman*
Digital Storage Using Ferromagnetic Materials	P. D. ATKINSON, A. E DEBARR, R. MILLERSHIP, R. C. ROBBINS, Elliott Brothers, Ltd.
Some Recent Research on Ultrasonic Propagation in Solid Media	T. F. ROGERS, Air Force Cambridge Research Center
Static Magnetic Memory—Its Application to Computers and Controlling Systems	AN WANG, Wang Laboratories
Static Magnetic Memory for the ENIAC	ISAAC L. AUERBACH, Burroughs Adding Machine Co.
Magnetic Binaries in the Logical Design of Information Handling Machines	N. B. SAUNDERS, Transducer Corp.

May 3, 1952, 2:00 to 5:00 p.m.

Mathematical Applications IV	ALSTON S. HOUSEHOLDER, Oak Ridge National Laboratory, *Chairman*
Discussion on the Use and Construction of Subroutines	ROSELYN LIPKIS, NBS, DAVID J. WHEELER, University of Illinois and University of Cambridge, JOHN W. CARR, MIT, GRACE M. HOPPER, Remington-Rand, Inc.

May 3, 1952, 2:00 to 5:00 p.m.

Theory III—Logic and Circuit Synthesis	H. H. AIKEN, Harvard Computation Laboratory, *Chairman*
Formal Logic and Switching Circuits	THEODORE KALIN, Air Force Cambridge Research Center
Minimization Synthesis of Two-Valued Feed-Back Circuits	WILLIAM BURKHART, Monroe Calculating Machine Co.

Characteristic Numbers and Their Use in the Decomposition of Switching Functions	WARREN L. SEMON, Harvard Computation Laboratory
Rectifiers as Elements of Switching Circuits	PETER F. STRONG, Harvard Computation Laboratory
The Theory of Counting Techniques	THEODORE SINGER, Harvard Computation Laboratory
The Application of Counting Techniques	ROBERT L. ASHENHURST, Harvard Computation Laboratory

University of California at Los Angeles.—On April 30, May 1–2, 1952, at UCLA, a symposium on electronic computers was held under the sponsorship of the Los Angeles IRE Professional Group on Electronic Computers and the Department of Engineering of the University. The program was as follows:

Wednesday, April 30, 1952

Registration	
Opening Session	
Introduction	R. G. CANNING, Naval Air Missile Test Center, Point Mugu, Calif.
Welcome	L. M. K. BOELTER, Department of Engineering, UCLA, *Chairman*
Keynote—Engineering Tomorrow's Computers	H. D. HUSKEY, NBSINA
Session on Magnetic Devices	JOHN J. CONNOLLY, The RAND Corporation, *Chairman*
Design Features of a Magnetic Drum Memory for the National Bureau of Standards Western Automatic Computer (SWAC)	R. THORENSEN, NBSINA
Problems Involved in Magnetic Tape Recording	NORMAN E. GIBBS, Raytheon Manufacturing Company
Survey of Tape Drive Systems	HAROLD SARKISSIAN, Computer Research Corporation
Session on Analog Devices	W. L. MARTIN, Department of Engineering, UCLA, *Chairman*
An Electro-Mechanical Multiplier for Analog Computer Application	S. E. DORSEY, Naval Ordnance Test Station, Inyokern, Calif.
The Thermal Analyzer—A Special Purpose Analog Computer	W. L. MARTIN and R. BROMBERG, Department of Engineering, UCLA
Panel Discussion—Utilization of Germanium Diodes	L. L. KILPATRICK, North American Aviation, Inc., *Chairman* NORTON BELL, Consolidated Engineering Corporation L. S. PELFREY, Hughes Aircraft Co. W. SPEER, Computer Research Corp. A. S. ZUKIN, Hughes Aircraft Co.
Panel Discussion—Designing for Maximum Reliability	HARRY T. LARSON, Hughes Air-craft Company, *Chairman* JOHN J. CONNOLY, The Rand Corp. HARRY D. HUSKEY, NBSINA ROBERT LUSSER, Naval Air Missile Test Center, Point Mugu, Calif. ROBERT RAWLINS, Lockheed Aircraft Corporation WILLIAM WAGENSEIL, Hughes Aircraft Company

Thursday, May 1, 1952

Session on Programming and Coding	ROSELYN LIPKIS, NBSINA, *Chairman*
An Approach to the Use of the IBM Card-Programmed Electronic Calculator in the Solution of Engineering Calculations	MURRAY L. LESSER, Northrop Aircraft, Inc.
Some General Precepts for Programmers	E. C. YOWELL, NBSINA
Programming for On-Line Computations	H. LUXENBERG, Hughes Aircraft Co.
The Human Computer's Dreams of the Future	IDA RHODES, NBS (Read by R. R. REYNOLDS, NBSINA)
Automatic Program Control Utilizing a Variable Reference for Addressing	A. S. ZUKIN, Hughes Aircraft Co.
Programming for Finding Characteristic Values of Mathieu's Equation and the Spheroidal Wave Equation	GERTRUDE BLANCH, NBSINA
Session on Input-Output Equipment	H. DOELEMAN, Electronic Engineering Company of Calif., *Chairman*
The Benson-Lehner Photoformer	D. L. PITMAN, Benson-Lehner Corp.
Input-Output on the New IBM Scientific Computer	M. M. ASTRAHAN, IBM Corporation
An Accurate Digital-Analog Function Generator	W. A. FARRAND, North American Aviation, Inc.
Some Techniques of Analog to Digital Conversion	H. E. BURKE, Consolidated Engineering Corporation
The Teleplotter, A Digital Plotting Device	DONALD F. BELLOFF, Telecomputing Corp.
Summary Session	
Introduction	R. L. SISSON, Computer Research Corp.
Speakers	D. H. LEHMER, NBSINA LOUIS N. RIDENOUR, International Telemeter Corp.

Friday, May 2, 1952

Tours were conducted to the following facilities:

The Rand Corporation,
Telecomputing Corporation,
California Institute of Technology,
Northrop Aircraft, Inc.

OTHER AIDS TO COMPUTATION

Linear Algebraic Systems and the REAC

1. Introduction. A great variety of problems in both pure and applied mathematics involves, either directly or indirectly, the solution of systems of simultaneous linear algebraic equations. Although the solution of such a system can readily be indicated by determinants, it is found that the attainment of actual numerical answers frequently becomes laborious for systems of order greater than three. Further, it is found that a great deal of effort has been, and is being, expended in the development of numerical procedures and in the design and development of computers which can be applied to such linear systems.

The approach in this instance is somewhat different. Here we are interested in extending the utility of an existing computer, or to be more specific,