I

UNITED STATES ATOMIC ENERGY COMMISSION

LA-1418

FLOATING DECIMAL CODING SYSTEM FOR CPC No. 4 Control Panels

By
Sidney L. Lida

May 1952

Los Alamos Scientific Laboratory

Technical Information Service, Oak Ridge, Tennessee

II

INSTRUMENTATION

In the interest of economy, this report has been reproduced direct from copy as submitted to the Technical Information Service.

PRINTED IN USA
PRICE 70 CENTS
Available from the
Office of Technical Services
Department of Commerce
Washington 25, D. C.

Work performed under
Contract No. W-7405-eng-36.

AEC, Oak Ridge, Tenn.-W25770

III

ABSTRACT

A new development in the field of floating decimal calculations for the IBM Card Programmed Electronic Calculator (Model II) is a three-factor two-operation control panel with automatic calculation of certain transcendental functions. Channel X, as the third factor is called, is capable of delivering the previous result of a calculation, or a number punched in the card, or the number stored in either one of two counter groups, to the electronic calculator. It is then possible to calculate any possible arithmetic combination of A, B, and X. Other operations include transcendental functions, manipulative operations, and other special functions. There are many ways for the machine to make decisions. The most important of these is the ability to read instructions from four different fields of the card under varying conditions. A system of checks and adjustments has been devised in case a calculation violates the range of the floating decimal notation. There are stops which tell an operator when he has inadvertently tried to divide by zero or take the square root of a negative number. This report is specifically designed to be read by an experienced coder as well as a relative beginner in the card programmed calculator.

IV

TABLE OF CONTENTS

IV

PAGE

PART II

PART III

VI

PAGE

4F SUPPLEMENT

LA-1418

INTRODUCTION

This paper will introduce a new theme in write-ups for CPC coding. Coding for the tabulator, electronic calculator, and summary punch will all be integrated into this one paper. The multi-fold purpose is:(1) to prepare a paper which is extremely brief and can be used by an experienced coder without consideration of tedious detail needed by a beginner, (2) to answer the need of an experienced but occasional coder who needs essential details, and (3) to give a relative beginner in CPC and floating decimal coding, examples of all types of operations along with extensive details. The aim throughout has been to make the write-up readable and understandable without regard to elegance of presentation.

With this purpose in mind the paper has been divided into three parts. Part I consists of a straight-forward explanation of essential details, Part II is a brief explanatory index of control punches, special addresses, electronic operations, and switch functions. Part III consists of detailed examples of all operations.

Thus, Part II will suffice for the expert, Part I and II for the occasional coder, and Parts I, II, and III for the beginner. It is suggested that the beginner who is unfamiliar with the idea of floating decimal calculations turn to Example One of Part III, and then return to this introduction.

In Part I superscripts will be used. A given superscript will refer to the number of the example in Part III which illustrates the particular point in question.

In general the philosophy of calculation differs from that of previous Los Alamos set-ups. There exists an additional channel called X. A full discussion of this channel will appear later. With this new channel it is then possible to operate on three numbers instead of two. There is a first attempt to partially store instructions and also be able to skip particular cards in a deck under specified conditions. A logical floating decimal zero is employed along with automatic checks and adjustments if the range of the floating decimal notation is violated.

At the end of this write-up is a"4F" supplement dealing with a faster combination operation control panel (the 4F panel) designed by Dura Sweeney. With one switch it is possible to use this new panel with almost identical coding described herein. After mastering the general #4 control panels and reading the 4F supplement, it will be an extremely simple matter to use the 4F control panel. It has been found that a typical problem can be completed in about two-thirds normal computing time. For this reason it is believed that this combination of control panels will become the standard production panels.

PART I

ESSENTIAL DETAILS OF CODING

INSTRUCTION CARD LAYOUT:

Each instruction card is divided into fields as shown in Table 1.

Table 1

Columns	Assignment
1-9	Instruction field #1 (normal)
10	Thousands digit of card number
11-20	Normal A read-in (address of 90)
21-30	Normal B read-in (address of 90)
31-39	Instruction field #2
40	Hundredths digit of card number
41-49	Instruction field #3
50	Tens digit of card number
51-60	Alternate A read-in (address of 99)
61-70	Alternate B read-in (address of 99) and X read-in (address of Y in 3)
71-79	Instruction field #4
80	Units digit of card number

Each of the 9 columns in an instruction field are given the assignments indicated in Table 2.

Table 2

Columns	Assignment
1 and 2	Channel A address
3 and 4	Operation to be performed
5 and 6	Channel B address
7 and 8	Channel C address
9	Special control punches

Note: X and Y punches in the 9 columns of the instruction field are considered to be an integral part of the fields and are used, as is all of column 9, for special control punches. It may be noted that channel X addresses consist of special control punches.

ADDRESSES:

a. Storage Registers and Counter Groups[2]

In general, CPC operations utilize three channels over which numbers may pass. Channels A and B are channels over which numbers are delivered to the electronic calculator in preparation for an operation. Channel C is a path over which the result of this operation may be sent to some storage register.

Channel A, B, and C addresses each consists of 2 digits designating the contents of some storage. There are two types of storage, (1) consisting of registers available in the auxiliary storage units (type 941) that may be attached to the CPC, and (2) counter groups which are always available. The addresses of these storages follow:

11 through 18 21 through 28	16 storages (type 941) available if one or more 941 units are attached.
31 through 38 41 through 48	16 additional storages (type 941) available if two or more 941 units are attached.
51 through 58 61 through 68	another 16 additional storages (type 941) available if three 941 units are attached.
91 through 98	8 storages (counter groups) always available.

Thus an address of 15 in columns 1 and 2 (channel A address) means that storage register 15 will have its contents delivered to channel A. An address of 97 in columns 5 and 6 (channel B address) means that the number in counter group 97 will be delivered to channel B. A channel C address of 23 tells the machine to reset this register to zero and then accept the result of this calculation, C. If the C address were 91, the same thing would happen except that counter group 91 would be affected instead of register 23. A C address is needed only when one wants to send a result to storage; otherwise the C address columns are left blank.

Calling on a register or counter group to deliver its contents to channels A or B or both does not destroy the number in that storage. But when sending a new number to storage the storage first resets to zero before accepting the new number.

Timing problems must be taken into consideration. If a particular storage register is addressed on channel C of a given card, then at least <u>two cards must intervene</u> before that particular storage register can be addressed on channel A or B. In the case of counter groups, however, <u>one</u> intervening card is all that is necessary.

One may call out numbers on channel X from counter groups 91 and 92 only. An X5 punch (read as an X punch in column 5 of the instruction field being used) tells counter group 91 to read onto channel X. An X7 punch tells counter group 92 to read onto channel X. Timing considerations are the same as if we had called on these counter groups from either channel A or B.

Problems in timing also dictate that the same storage may <u>not</u> be addressed on channel C on successive cards. Specifically, if a given storage register (type 941) is addressed on channel C, at least two cards must intervene before again addressing this storage register on channel C. Only one intervening card is necessary for counter groups.

b. Reading into Channels from the Card:[3]

Numbers punched in certain fields of an instruction card may enter channels A or B for immediate use in a calculation. An A (or B) address of 90 tells the machine to take the number punched in columns 11-20 (or 21-30) of the card and deliver it to channel A (or channel B). These are the normal card read-in fields for A and B. There is provision for reading from alternate fields on A and B. An A (or B) address of 99 tells the machine to take the number punched in columns 51-60 (or 61-70) of the card and deliver it to channel A (or channel B). A Y3 punch tells the machine to read onto channel X from columns 61-70 of the card. Since we have assigned columns 61-70 as <u>either</u> channel X card read-in or channel B alternate card read-in we <u>can</u>, if we wish, read a number punched in this field into both channels at once if we have both a 99 B address and a Y3 punch. These addresses (like <u>all</u> addresses) may be used in <u>any</u> instruction field. An X over the right most digit of a field indicates the number is negative.

c. Using Result of Previous Card[4]:

The result of the calculation of the previous card may enter channels A or B for immediate use in calculating. An A (or B) address of 9X tells the machine to take the result of the previous card and deliver it to channel A (or channel B). If channel X has no specific address the number available on that channel for calculation is automatically the result of the previous card. The result of the previous card may be sent to any storage without interfering with the above operations.

d. Sign Control and Absolute Value[5]:

The sign of a number entered on channel A or B may be changed. A Y punch over the right hand digit of an A or B address will change the sign of the number delivered to the channel by that address. The positive absolute value of a number may be delivered to channel A or B by punching a Y over the left hand digit of the address. The negative absolute value may be delivered by punching a Y over both digits of the address. The above modifications are operative for <u>any</u> A or B address, including the 9X addresses.

An X punch over the right hand digit of the C address will change the sign of the result before it enters storage. This X8 punch <u>will</u> <u>also</u> change the sign of the result <u>before</u> an automatic transfer to channel X. Thus, even if we do not have a C address, the X8 punch may be used to change the sign of the previous result before entering on channel X. <u>This X8 punch will never, in any</u>

case, change the sign of the result before entering channels A or B by means of 9X addresses. In fact, the X8 punch affects nothing except as noted above.

No matter what punches are used to alter the sign of the channels, the altered sign is the one which actually lists. To help the coder check a listing there are * (asterisk) or CR symbols which print whenever the sign has been changed by any of the above punches. The location of these symbols on the listing may be seen by reference to the section on instruction card listing.

An * or CR symbol is used to indicate a negative number on channels C or X. A 9 or N following the number listed on channels A or B indicates a negative number on those channels. The listing always prints the absolute value of the number with sign. There should never be a nines-complement number listed in any case.

e. Multiple C Addresses[6]:

It is occasionally convenient to be able to send the result of a calculation to two storages for future use. It is possible to do this by double digit punches in the right hand digit of the channel C address. Thus a C address which has a 3 as the left hand digit and both a 2 and 7 as the right hand digit, stores the result of the calculation in storage registers 32 and 37. It should be noted, then, that we are limited to just double digit punches in the right hand C address. The right hand limitation means, for example, that we cannot send a result to both 36 and 46 at the same time. Only storage registers (not counter groups) may be addressed in this way.

It is also permissible, with limitations, to send a result to a counter group and a storage register at the same time. A number may be sent to a given counter group and any storage register provided the right hand digit of the storage register address is the same as the right hand digit of the counter group address. Thus we may have a 93 and 53 multiple C address since this conforms to the above rules. A 93 and 56 would not conform to the rules above. A multiple C address of 23 and 33 would violate the rule of the preceding paragraph. It is never possible to send a result to two different counter groups simultaneously.

f. Non-Clear Counter Group Before Read-In[7]:

A counter group, when addressed on channel C, normally resets to zero before accepting the new number. This clearing of the counter group may be eliminated by a Y7 punch. The counter group is told by this Y7 punch to add the absolute value of the number it was previously holding to the absolute value of the result of this calculation and to give the sum the sign corresponding to the sign of the new number being entered. This Y7 punch is ignored by the machine if the C address is not for a counter group.

This non-clearing feature was requested by people making tables with the CPC and ordinarily will not be used.

It should be noted that the sum of the absolute values is not done in floating decimal arithmetic. The counter group does not recognize a floating decimal number and consequently can add only in fixed decimal arithmetic. That is, it adds the two numbers digit for digit with proper carries to the next higher order digit. Also, since the counter group's capacity is only 10 digits, any carry resulting from addition of the high order digits of the two numbers is lost.

SPREAD-READ-IN[8]:

Spread-read-in is a sequence of two operations designed to place eight ten-digit numbers with sign into the eight counter groups. The two operations, each instructed by a separate card, may be considered one spread-read-in operation. The first card is called the spread-read-in control card and must always be followed by one and only one spread-read-in card. Since these two cards are thought of as being one operation we cannot have either of the cards without the other. The first card of the sequence, the spread-read-in control card, must have a 90 as the C address. This 90 does not function as a normal C address because no storage in the CPC has an address of 90. The 90 is there simply to tell the machine to prepare itself to accept the next card as a spread-read-in card instead of as a regular instruction card. The spread-read-in control card must not have normal instructions such as addresses and an operation. It may, however, have control punches which affect listing and double spacing (the controls for these two functions are to be found under PRINTING CONTROLS). This card also may contain Y punches in columns 10, 20, 30, 40, 50, 60, 70, or 80. One of these punches will prevent a selected counter group from resetting to zero before accepting the new number punched in the next card. A Y punch in column 20 concerns counter group 92, a Y punch in column 50 concerns counter group 95, etc. If a counter group is told not to clear by one of these Y punches, the number in the counter group after the spread-read-in card will be the sum, in fixed-decimal arithmetic, of the absolute value of the number in the counter group before the spread-read-in card and the absolute value of the number entering the counter group on the spread-read-in card. This sum will be given a positive sign if both numbers were positive and a negative sign if either or both of the numbers were negative. Thus if we prevent clearing before spread-read-in, and spread-read-in a positive zero, we will preserve the number that was in the counter group before the spread-read-in operation. Any number of counter groups may be given this instruction.

Because of timing conditions, the card preceding the spread-read-in control card must not send a number to any counter group.

The second card of the sequence, the spread-read-in card itself must contain the numbers to be entered into the counter groups. Table 3 correlates (1) card columns in which a number is punched, (2) the counter group the number will enter, and (3) the type bars used to print the number. Any digit punched anywhere in the card will be regarded as part of a number to be entered into a counter group and not as an instruction

Table 3

Counter Group Entered	Cols. of SRI Card	Type Bars Used for Printed Record	
		Digits	Sign
91	1-10*	α2-α11	α12
92	11-20*	α13-α22	α23
93	21-30*	α24-α33	α34
94	31-40*	α35-α44	n1
95	41-50*	n2- n11	n12
96	51-60*	n13- n22	n23
97	61-70*	n24- n33	n34
98	71-80*	n35- n44	n45

* An X punch in the right most column indicates the number is negative.

α type bars constitute the left 44 type bars.
n type bars constitute the right 45 type bars.

No X or Y punches may be put into the spread-read-in card other than the X-punches used to designate negative numbers. If one wants to list this card to make a record of the numbers entering the counter groups, he must do so by control punches in the spread-read-in control card.

The card immediately following a spread-read-in card may have A and B addresses for counter groups. If this is the case, the numbers entering the channels will be the new numbers which have just been put into the counter groups.

CAUTION: Since all 80 columns of a spread-read-in card are used, this card must not have a card number punched in it.

SPREAD-READ-OUT[9]:

Spread-read-out is a sequence of operations designed to create a permanent record of the numbers stored in the eight counter groups.

An X punch in column 4 of the instruction field will initiate a spread-read-out. With this operation the tabulator will stop feeding cards, the summary punch will punch a card, the tabulator will print a line showing the numbers in the counter groups, and finally the tabulator will resume normal operations and start feeding cards

Thus the spread-read-out operation simply records numbers stored in the counter groups. The numbers in the counter groups are not destroyed and operations may be continued as if they had never been interrupted by this recording process. Table 4 correlates counter groups, card punching, and printing.

An * (asterisk) will appear to the left of the printed line, for identification purposes, whenever we instruct a spread-read-out.

We may ignore the spread-read-out instruction entirely by placing Switch #3 in A position. If Switch #2 is in A position we eliminate the punched record but retain the printed record.

Table 4

Counter Group to be Recorded	Cols. used for Punched Card Record	Type Bars used for Printed Record	
	Digits	Digits	* or CR symbol if number is negative
91	1-10*	α2-α11	α12
92	11-20*	α13-α22	α23
93	21-30*	α24-α33	α34
94	31-40*	α35-α44	n1
95	41-50*	n2- n11	n12
96	51-60*	n13- n22	n23
97	61-70*	n24- n33	n34
98	71-80*	n35- n44	n45

* An X punch over the right most digit indicates the number is negative.

α type bars constitute the left 44 type bars.

n type bars constitute the right 45 type bars.

The card ordering the spread-read-out may have normal instructions. The only restriction is that <u>none</u> of the addresses may be for counter groups. The <u>card following</u> the card ordering the spread-read-out <u>must be completely blank</u>, except possibly for card number. This card will <u>never</u> list.

One should note that the card punched by the summary punch is in the proper form to be used as a spread-read-in card later on in the problem. In fact this is a common use of the summary punched card, to provide external storage.

INSTRUCTION CARD PRINTING AND PRINTING CONTROLS:

a. Printing:

When an instruction card is listed, information indicated in Table 5 will print in the places shown.

Every card in an instruction deck may be made to list by turning "set-up change" Switch #1 to the "on" position.

An X1 punch in a given card will unconditionally cause the card to list. An X3 punch in a given card will unconditionally cause the following card to list. These punches are called immediate list and delay list respectively. They are both termed short list instructions because only cards affected by these punches will list if "set-up change" Switch #1 is not on. A Y8 punch will cause an immediate list only if Switch #5 is in the A position. This punch may be used to create what is called an intermediate list for spot checking a calculation if trouble is suspected.

Table 5

Information	Type Bars	
	Digits	Signs
Number delivered on channel A	α2-α11	α12
Number delivered on channel B	α13-α22	α23
Number delivered to channel C for the preceding card**	α24-α33	α34
Number delivered on channel X	α35-α44	n1
* (asterisk) if |A| is instructed	n3	
Channel A address	n4-n5	
CR symbol if -A is instructed	n6	
Operation code	n8-n13	
* (asterisk) if |B| is instructed	n15	
Channel B address	n16-n17	
CR symbol if -B is instructed	n18	
CR symbol if non-clear counter group is instructed	n20	
Channel C address	n21-n22	
* (asterisk) if one orders a change of sign of a result before storing or entering on channel X	n23	
* (asterisk) if there is a short list or intermediate list instruction	n25	
Field from which instructions were read	n26	
Card number	n28-n31	
Numbers punched in columns 71 through 80 (useful for additional identification besides card number)	n35-n44	

** This information, as noted, is for the preceding card. All other information is for the card which is being listed.

b. 80-80 List:

If Switch #1 is in A position we ignore instructions and list all 80 columns of the card. This listing will take place at 150 cards per minute and, since instructions are ignored, no calculations will take place. X punches in columns 10, 20, 30, 40, 50, 60, 70, and 80 will list as negative signs for numbers punched in the same column and preceding nine columns.

c. Double Spacing:

A 7 punch in column 9 of the instruction field of a card will cause the tabulator to double space before listing the next card. This will happen, of course, only if the next card has been told to list by some listing instruction. If the next card is not told to list, the machine will remember the double space instruction until some following card <u>has</u> been told to list. Thus, a double space instruction will be remembered until the tabulator is told to print a line.

d. Restore Paper:

The tabulator prints on a continuous roll of paper. This continuous roll, however, is broken periodically by perforations across the sheet. These perforations divide the roll into equal sections called sheets.

A 9 punch in column 9 of the instruction field of a card tells the machine to list the <u>next</u> card on the 1st line of the next sheet.

Manual control of the restore paper operation may be accomplished by manipulation of buttons located to the upper right of the tabulator carriage.

STOPPING THE MACHINE:

The CPC may be stopped manually by depressing the stop key, but there are other occasions when it stops itself automatically or by coded instructions.

A 1 punch in column 9 of the instruction field of a card tells the machine to stop itself if the result of the calculation performed on that card is negative.[10]

The machine will automatically stop if one has inadvertently tried to take the square root of a negative number or divide by zero.[11] It will also stop if the result of a calculations is beyond the capacity of the floating decimal notation in the upward direction.[12]

An X punch in column 25 of a card (note that column 25 is not associated with any instruction field) instructs the machine to compare the number punched in columns 71 to 80 of the card with the result of the preceding card if Switch #4 is on. If the numbers do not compare the machine stops.[13] It is possible to use this feature in coding a problem to automatically check intermediate results of a calculation designed to check-out the coding. Its primary function, however, is for use in checking the accuracy of the machine via test decks.

The machine may be restarted in any of the above cases by depressing the START and FINAL TOTAL buttons simultaneously. They must be held down until the machine feeds two cards.

STORED INSTRUCTIONS[14]:

Although the CPC is not designed to be compatible with the idea of stored instructions, we have been able to store A and B addresses in any storage of the machine.

An X punch in column 15 of a card will cause part of the number called out on channel A of the card to be used as the A and B addresses for the next card. One may calculate with this number called out on channel A in any way desired. This calculation will not affect the use of channel A as instructions for the next card. The timing and use of this feature is best explained by the example designated by the superscript above.

SELECTED CARD SUPPRESSION[15]:

It is now possible to have the machine test the result of a calculation and determine its sign. At any time later in the deck it is possible to selectively ignore any desired card or cards if the result of the previous test was positive or negative. Any number of these sign tests may be made, but everytime a test is made the result of the previous test is destroyed.

An X punch in column 11 of a card orders a sign test to be made on the result of the card. As early as the next card, or on any succeeding card, we may have an X punch in column 12. This punch instructs the CPC to ignore all instruction fields in the next card (the card immediately succeeding the X12 card) if the result of the previous sign test was positive. If the result of the sign test was negative, the next card would be treated as an ordinary instruction card. Note that only the instruction fields are ignored. This, by definition, includes all addresses, operations, and control punches in any instruction field. Thus, for example, the machine will not ignore an X punch in column 11 because this punch is not in any instruction field.

An X punch in column 13 of a card calls for ignoring the next card if the result of a previous sign test was negative.

Timing conditions for use of this feature are explained completely in the example in Part III.

CHANGING THE INSTRUCTION FIELD[16]:

As was mentioned before, there are four separate instruction fields from which the machine may be told to take its instructions. The previous section dealing with selected card suppression told of a way to make the machine decide whether to skip a certain card or not. This feature, coupled with the conditional changes of instruction fields, are means by which a coder may have the machine automatically decide what course to take under different problem conditions.

To change instruction fields, we must first tell the machine under what condition we want it to make a change. Then we must tell it which field to shift to, if it actually does change due to the specified condition. The machine is capable of changing from any instruction field to any other.

Table 6 shows the coding necessary for the different types of conditional shifts.

Table 6

Punch in Col. 9 of Instruction Field Being Used	Condition Under Which a Change Will Be Made
X	if the result of this calculation is negative.
5	if the result of this calculation is positive or zero.
6	if the result of this calculation is zero.
8	an unconditional order to change field.

If the machine decides to change fields it will automatically change to instruction field #1 if not instructed to go to any other field. It can be told to go to field #2 by a 2 punch in column 9 of the instruction field being used. A 3 punch tells it to go to field #3 and a 4 punch to field #4. This punch is put in the same card as the punch specifying the condition for the shift.

Thus, assuming the machine is taking its instructions from field #3 and we put an X punch and a 2 punch in column 49 (column 9 of the instruction field being used), the machine will start taking its instructions from field #2 if and only if the result of the calculation is negative.

Timing considerations dictate that if we order a shift on card "n", and if the shift actually occurs, then the shift will not be effective until card "n+2". In other words, it takes two cards for the machine to change instruction fields. Instructions for card "n+1" will be read from the old instruction field, and can be used for calculations if desired.

One may order a change of instruction fields if the number called out on channel A is odd. Determining this condition of being odd or even is done in the electronic calculator and, hence, must be coded for as an operation. The field to which the machine goes for its instructions, if the shift actually occurs, is specified in the usual way by punches in column 9. The operation is called for by a 1 punch in column 3 and a double punch of a 5 and 8 in column 4. There must also be an "exponent" of 57 entered on channel B. If one does the preceding coding, the procedure is: (1) the calculator examines the <u>integral part</u> of the number, rounded according to the decimal part*, called in on channel A; (2) if this integral part is odd the machine will start taking its instructions from the newly designated field; (3) the delay before the new field becomes effective is the same here as in all the previous field shifts.

Once a change is made, the machine continues to use the new field of instructions until, by the coding, it is ordered to again change fields.

However, when one is first starting to use the machine, he can be assured that the machine initially will take its instructions from field

---

*If the number called in on channel A is 12.6 then the integral part of this number rounded according to the decimal part is an odd number, namely, 13.

#1. The design of the control panels is such that the machine automatically returns to field #1 when all of the cards are run out of the machine and into the hopper. This is the reason for calling it the normal field of instructions.

OPERATIONS:

a. General[1]:

As mentioned earlier the calculator is capable of operating on three factors, A, B, and X. Channel X is operative only for those operations which specifically call for its use.

This section is meant to be a somewhat general discussion of the capabilities of the calculating unit. Thus, operations are divided into five classes; arithmetic, special, manipulative, discriminatory and transcendental functions. All calculations are performed in floating decimal arithmetic. To fully comprehend all operations it is necessary to study the operations section of Part II along with this section.

A systematic and concise treatment of the operations, including the operation coding needed for specific operations, is given in Part II. However, superscripts referring to explanatory examples in Part III will be found both here and in Part II.

b. Zero Representation:

Zero representation is a peculiar problem in floating decimal calculation. In all other floating decimal systems at Los Alamos an artificial zero has been created. That is, if a number is subtracted from itself, the significant digits of the result are all ones. The "exponent" is the "exponent" of the original number decreased by 9. This artificial result is then called the machine zero. The disadvantages of such a number are evident. The present design has been to produce a "logical" zero. This means that in the example above, the significant part of the result would be, as expected, all zeros. The "exponent" is also made to be zero. Furthermore, the machine will recognize this type of zero and calculate with it in a logical manner. For example, any number multiplied by a "logical" zero will result in a "logical" zero. If one attempts to divide by zero the machine stops.[11] When any decisions are to be made, it must be remembered that the machine considers the "logical" zero to be a positive number.

c. Automatic Checks and Adjustments:

The machine stops if one has inadvertently tried to take the square root of a negative number.[11] Also, since the "exponent" can not be larger than 99 nor less than zero, it is possible for the result of a calculation to go out of the range of this notation in either direction. If the result goes out of range in the upward direction the machine stops, since this will dictate another coding of the problem or a correction of an error.[12] On the other hand, if the range is violated in the downward direction the machine simply converts the result to a "logical" zero and continues to calculate. This last automatic adjustment is based on the assumption that when a number becomes so small the coder is willing to consider it zero.[17]

d. Arithmetic Functions[1]:

It is possible to calculate any possible arithmetic combination of two factors, A and B, or of three factors A, B, and X.

Part II shows all of these combinations as basic functions. For example, one such basic function is A+B+X. By use of modifying control punches described above in the section on "ADDRESSES - Sign Control and Absolute Value", we find that the calculation of $|A| - |B| + X$ is just as feasible as the basic form, A+B+X.

In general, the combination operations which involve three factors, will take the time of two cards although sometimes the time of one card will suffice. The 4F control panel is so designed that almost all of these double operations will be done in one card time. NOTE: It should be stated that cards can pass through the machine at a maximum rate of 150 per minute. Thus, one card time is defined as $\frac{1}{150}$ of a minute or .4 seconds.

To transfer a number from channel A to channel C (this is usually done to load storages before a problem begins) we simply add zero to the number called out on channel A.

e. Special Functions:

One operation allows the machine to calculate the square root of the number entered on channel A.[18]

A new special function permits the calculation $A^n \cdot B$ when n is an integer and $1 \leq n < 10^8$.[19]

f. Manipulative Functions[20,21,22,23,24]:

This class of functions includes non-arithmetic operations. With two of these functions one may isolate the integral and fractional parts of a number. The remaining operations allow calculations with the "exponents" of numbers. These latter types are essential to efficiently reduce an arbitrary argument in preparation for evaluating a transcendental function.

g. Discriminatory Functions:

These are operations which make decisions. We have previously discussed methods for making decisions, and only one of these required a distinct operation. This operation is the one which conditions a change of instruction field if the number entered on channel A is odd (see CHANGING THE INSTRUCTION FIELD).[16]

Another operation in this category is one which examines the numbers entered on channels A and B and selects algebraically, the smaller of the two. A parallel operation may be used to select the larger of the two.[25]

h. Transcendental Functions:[26,27,28,29]

In one card the machine will calculate e^x, sin x, cos x, sinh x, cosh x, $\frac{1}{2}\log\frac{1+x}{1-x}$, or arctan x. The argument, x, however, has limitations. For example, in calculating sin x it is essential that x be in radians and $0 \leq x < 1$. For an arbitrary argument, then, this implies a card programmed reduction routine.

The above functions are evaluated by well known series formulas.

The coder decides how many terms of the series he wishes to take as explained in the examples in Part III. If he wishes 8 significant figure accuracy he must take a certain number of terms in the series. However, if he needs only 4 significant figures he may tell the machine to take fewer terms of the series and consequently save computing time.

The example for each function in Part III contains the card programming (coding) necessary for reduction of an arbitrary argument and the operation necessary to evaluate the series. The coder may use this programming when he wishes to evaluate a function in his own coding.

PART II

BRIEF EXPLANATORY INDEX OF OPERATIONS AND CONTROL

INSTRUCTION FIELD CONTROL PUNCHES

a. X Punches in the Instruction Field Columns

X1. . . . . immediate list

X3. . . . . delay list

X4. . . . . spread read-out (summary punch)

X5. . . . . channel X address for counter group 91

X7. . . . . channel X address for counter group 92

X8. . . . . change sign of C before entering storage or C ⟶ X transfer

X9. . . . . change instruction field if C is negative

b. Y Punches in the Instruction Field Columns

Y1. . . . . $|A|$ (absolute value of A)

Y2. . . . . -A (change sign of A)

Y3. . . . . read channel X from columns 61-70 of card

Y5. . . . . $|B|$ (absolute value of B)

Y6. . . . . -B (change sign of B)

Y7. . . . . non-clear counter group before read-in

Y8. . . . . intermediate list (immediate list if Switch #5 is in A position)

c. Digit Punches in Column 9 of Instruction Field

1. . . . . stop machine if C is negative

2. . . . . change to instruction field 2

3. . . . . change to instruction field 3

4. . . . . change to instruction field 4

5. . . . . change instruction field if C is positive or zero

6. . . . . change instruction field if C is zero

7. . . . . double space before printing the next line

8. . . . . change instruction field unconditionally

9. . . . . restore paper to next sheet

OPERATIONS

a. Arithmetic Functions

Coding			Coding		
Col 3	Col 4	Basic Function	Col 3	Col 4	Basic Function
1	-	A + B*	3,8	4	$\frac{A}{X \cdot B}$
3	-	$A \cdot B$	4	1	$\frac{A}{B} + X$
4	-	A/B	4,8	1	$\frac{X}{B} + A$
1	1	A + B + X	4,9	1	$\frac{A}{X} + B$
3	3	$A \cdot B \cdot X$	1	4	$\frac{X}{(A+B)}$
4	4	$\frac{B \cdot X}{A}$	1,8	4	$\frac{A}{(X+B)}$
4,8	4	$\frac{A \cdot B}{X}$	1	2,4	$\frac{(A+B)}{X}$
1	3	(A+B) X	1,8	2,4	$\frac{(X+B)}{A}$
1,8	3	(X+B) A	3	1	$(A \cdot B)+X$
3	4	$\frac{X}{A \cdot B}$	3,8	1	$(X \cdot B)+A$

* This addition operation may be used to transfer a number from channel A to channel C simply by making B equal to zero. It is not necessary to have a separate operation for transfer.

Since the above functions are simple arithmetic calculations there are no examples to explain them in Part III.

The coding for the above functions follow definite rules. Hence, if one knows the rules he need never refer to the table except for convenience.

In a combination operation we think of a first operation followed by the second operation. A 1, 3, and 4 punch mean respectively add, multiply, and divide. Coding for the first operation is found in column 3 while that for the second operation is in column 4. Let us assume that the first operation is performed on factors A and B to produce an intermediate result, I, while the second operation is performed on the factors I and X. Thus a 1 in column 3 and a 3 in column 4 gives us the function (A+B)X as indicated in the table. The 1 in column 3 tells the machine to add A and B to produce I. The 3 in column 4 instructs I times X thus producing (A+B)X.

An 8 or 9 punch in column 3 means to interchange A and X or B and X respectively.

Since division is not commutative we run into difficulties when applying the above rules. The exact meaning of division however, is brought out in the concise summary of rules that follow.

Let us assume the usual factors of A, B, and X along with I, the intermediate result, defined above.

Summary of Arithmetic Rules

Coding		Function
Col 3	Col 4	
1		A + B
	1	X + I
3		A · B
	3	X · I
4		A/B
	4	X/I
	2,4	I/X (division inversion)
8		A and X interchange
9		B and X interchange

With these rules one may verify the coding given in the table. For example, let us verify the function given by the coding of a 3 and 8 in column 3 and a 4 in column 4. The 3 gives us $A \cdot B = I$. The 4 gives us $\frac{X}{I} = \frac{X}{A \cdot B}$. The 8 instructs the interchange of A and X which finally gives the function $\frac{A}{X \cdot B}$ as indicated.

With these rules we may create arithmetic functions which are not listed. For example let us assume a 3 and 9 in column 3 with a 2 and 4 in column 4. The 3 gives us $A \cdot B = I$. The 2 and 4 gives us $\frac{I}{X} = \frac{A \cdot B}{X}$. The B and X interchange instructed by the 9 finally result in $\frac{A \cdot X}{B}$.

This last function we have just created is considered, however, to be equivalent to the function $\frac{B \cdot X}{A}$ which is found in the table of basic functions. The only difference between the two is the interchange of A and B. Since anything instructed by the coder to be read onto channel A can also be read onto channel B and vice versa, it is never necessary to have both of the above functions. There may be occasions where it is convenient to use a "created" function but it will never be necessary. All of the necessary functions are to be found in the table.

b. Special Functions

Coding		Number of Explanatory Example in Part III	Function	Special Requirements
Col 3	Col 4			
5		18	$\sqrt{A}$	B = A
3	6	19	$A^n \cdot B$	n is an integer with $1 \leq n < 10^8$ n is read into the calculator over channel X but not as a floating decimal number. n is the number in the low order positions of the significant digits of channel X. (See explanatory example).

c. Manipulative Functions

Coding		Explanatory Example in Part III	Operation
Col 3	Col 4		
1	8	20	If $A = x_A \cdot 10^{E_A}$ and $B = x_B \cdot 10^{E_B}$ and $E_B \geq E_A$ then the result will be x_A shifted to the right $(E_B - E_A)$ places and rounded according to the last digit shifted off. The true exponent of the result will be E_B.
1	9	21	If $A = x_A \cdot 10^{E_A}$ and $B = x_B \cdot 10^{E_B}$, the result will be x_A with the $(E_B - E_A)$ least significant digits zero. If $(E_B - E_A) \geq 8$ the result is a logical zero.
7	.	22	The result of this manipulation is the "exponent" of A attached to the significant digits of B.
7	8	23	The result is the "exponent" of A minus the "exponent" of B in floating decimal notation.
1	6,8	24	The integral part of A (rounded according to the fractional part) added to the "exponent" of X appears as the "exponent" of the result. An "exponent" of 57 must be entered on channel B.

d. Discriminatory Functions

Coding		Explanatory Example in Part III	Operation
Col 3	Col 4		
1	5,8	16	This operation is discussed in Part I under "CHANGING THE INSTRUCTION FIELD". An "exponent" of 57 must be entered on channel B.
2	9	25	The result of this operation is the smaller, algebraically, of A and B.
2	7,9	25	The result of this operation is the larger, algebraically, of A and B.

e. Transcendental Functions

Coding		Explanatory Example in Part III	Function	Special Requirements	
Col 3	Col 4			n	x
6	7,8	26	sin x	even	$0 \leq x < 1$; x in radians
6	7,8	26	cos x	odd	$0 \leq x < 1$; x in radians
6	-	27	e^x	arbitrary	$x^2 < 1$
6	8,9	28	$\frac{1}{2}\log\frac{1+x}{1-x}$	even	$x^2 < 1$
6	7,8,9	29	arctan x	even	$0 \leq x < 1$
6	8	26	sinh x	even	$x^2 < 1$
6	8	26	cosh x	odd	$x^2 < 1$

Note: x is entered over channel A. One must enter 10000000yy on channel B where yy is the value of n above. The value of n determines how many terms will be taken in the series expansion of the function.

SWITCH FUNCTIONS

a. Set-Up Change Switches

#1. . . . list every card

#2. . . . double space before every printing

#3. . . . MUST ALWAYS BE ON FOR PROPER OPERATION

b. Double Pole Double Throw Switches

#1. . . . 80-80 list

#2. . . . eliminate punched record on spread-read-out

#3. . . . ignore all spread-read-out instructions

#4. . . . must be on for X25 control punch to be effective

#5. . . . must be on for intermediate list cards to print

#6. . . . must be on if and only if the 4F control panel is being used.

#7. . . . not used

#8. . . . not used

#9. . . . not used

NOTE: All switches are normally off or in B position except "set-up change" Switch #3 which is always on. The switches function as described above only if they are in the on or A position.

SPECIAL ADDRESSES:

a. Channel A

Address	Number Delivered to Channel A
9X	Result of previous calculation
90	Number punched in columns 11-20
99	Number punched in columns 51-60

b. Channel B

Address	Number Delivered to Channel B
9X	Result of previous calculation
90	Number punched in columns 21-30
99	Number punched in columns 61-70

c. Channel X

Address	Number Delivered to Channel X
None	Result of previous calculation
Y3	Number punched in columns 61-70
X5	Number contained in counter group 91
X7	Number contained in counter group 92

d. Channel C

Address	Operation
90	Treat the next card as a spread-read-in card

PART III

EXAMPLES

INTRODUCTION:

Examples are numbered in sequence. The number in the upper right hand corner of each page shows what examples are to be found on each page. For instance, if the number is (3-6) it means that examples 3, 4, 5, and 6 are to be found on that page. The purpose of this is for easy reference when reading either Part I or Part II.

Each example follows the same pattern. First there is a descriptive heading. Second, a reference is made to some illustration. Third, there is a card by card analysis of the programming found in the illustration which brings out essential coding points of the subject matter being described. These parts are sometimes followed by a general discussion which emphasizes particular points in the coding. Many times it will be wise to read this last discussion along with the card by card analysis.

The illustrations are in a section at the end of Part III. They are numbered in sequence for easy reference. Each illustration sheet is basically an accurate reproduction of the coding sheet that will be used by the coder. Several examples may appear on one illustration sheet. Thus a reference to illustration 4b will indicate the second example on illustration sheet 4.

It is assumed throughout that one knows the coding for A+B, A·B, and A/B. This coding is a 1, 3, and 4 punch, respectively, in column 3 of the instruction field. For purposes of giving examples of channel X addresses we must also know the function A+B+X. The associated coding is a 1 both in column 3 and column 4 of the instruction field.

Examples of transcendental functions show the complete programming necessary for calculating the function of an arbitrary argument. This programming then reduces the argument to the necessary range before actually using the series operation. The card by card description is done only for the particular card on which the series operation is performed. Anyone wishing to know the reduction formulas used in this programming may consult the author.

The terms in the series expansion used for all transcendental functions include in the numerator an x^n factor. When it is stated that n terms are used in the series expansion for a given function it simply means that the series is calculated out to and including the term containing x^n or x^{n-1}, whichever is possible.

The complete programming for sin x, cos x, sinh x, and cosh x, are not shown for reasons given in their examples (see Example 26). Example 26 should be read before other examples on transcendental functions.

EXAMPLES

1. Floating Decimal Notation

In our floating decimal notation we represent a given eight digit number by a ten digit number. This ten digit number is then used by the machine to perform its calculations. Some examples of the representation follow:

Original Number	Intermediate Step	Floating Decimal Representation
765.34872 ⟶	7.6534872 x 10^{2} ⟶	7653487252
-.0081342974 ⟶	-8.1342974 x 10^{-3} ⟶	-8134297447
1.0000000 ⟶	1.0000000 x 10^{0} ⟶	1000000050
10.000000 ⟶	1.0000000 x 10^{1} ⟶	1000000051

The first eight digits of the floating decimal number are called the significant part of the representation since they are identical to the significant digits of the original number. The last two digits of the representation are seen to be the power of 10 in the intermediate step plus 50 and are called the "exponent". This two-digit "exponent" can then have a range of 00 to 99 inclusive. An "exponent" of 00 represents a true exponent of -50 while 99 would represent a true exponent of 49.

Discussions of all operations are in separate examples, but it is hoped that the concrete examples of the arithmetic operations which follow will make the notation clear.

Addition

True Number	582.70156 ↓	+ .61305767 ↓	= 583.21462 ↓
Floating Decimal Representation	5827015652	+ 6130576749	= 5833146252

Multiplication

True Number	9.2425636	x	77.415549	=	715.51814
Floating Decimal Representation	9242563650	x	7741554951	=	7155181452

Division

True Number	-.065258427	÷	.28546309	=	-.22860548
Floating Decimal Representation	-6525842748	÷	2854630949	=	-2286054849

Another important item of machine operation must be understood. The result of any operation (except operations specially designed to do the opposite) is always in floating decimal form, i.e., the first significant digit is non-zero. Thus, if we add a number to zero which is not in floating decimal form then the number <u>will</u> be in floating decimal form before it is stored or used on the next calculation. For example, if the number 0100000050 (in floating decimal representation) is added to zero, the result (in floating decimal representation) is 1000000049. Both representations have the value .1. Only the form of representation is different. The latter representation is the <u>true</u> floating decimal representation of one tenth.

2. Addresses - Storage Registers and Counter Groups

REFERENCE: Illustration 1a

Card 1 - Takes the number stored in counter group 93, adds it to itself and stores the result back into the same counter group thus destroying the number that was previously in 93.

Card 2 - Adds the numbers in storage registers 22 and 24 and puts the result in 22.

Card 3 - Counter group 93 is called in on channel A. This is permissible because we have allowed one intervening card since storing a number in 93. This number is added to the number in 24 and the result put in counter group 91.

Card 4 - This card simply shows that it is possible to have the same storage register addressed on A, B, and C at the same time. The same thing was done in the case of 93 in Card 1.

Card 5 - The calculation is A+B+X. The A, B, and C addresses mean the same as they have previously. The X5 punch tells channel X to deliver the number contained in counter group 91 to the calculator. Note that one card has intervened since sending a result to 91 while two cards have intervened since sending a result to 22.

Card 6 - The X7 punch tells channel X to deliver the number stored in counter group 92 to the calculator.

3. **Addresses** - Reading onto Channels from Card

REFERENCE: Illustration 1b

Card 1 - Channel A is read from columns 11-20 of the card and channel B from columns 21-30. These are the normal card read-in fields for the channels. The operation adds -1 (X punch in column 20 indicates the factor is negative) to 0. Thus a -1 is transferred from the card to storage register 11. This adding of zero to a number and then storing, is the usual way of transferring a number from one location to another.

Card 2 - Channel A is read from columns 51-60 of the card and channel B from columns 61-70. These are the alternate card read-in fields. The calculation is 2+3 = 5.

Card 3 - Shows the ability to read from an alternate and normal field on the same card. The Y3 punch tells channel X to read from columns 61-70 of the card. Thus channels X and B read from the same columns of the card and the calculation is 0-4-4 = -8.

Card 4 - The other combination of alternate and normal fields is shown. The calculation is 5+1+6 = 12.

4. **Addresses** - Using Result of Previous Card

REFERENCE: Illustration 2a

Assume that a number y is stored in register 12.

Card 1 - y times y equals y^2.

Card 2 - The 9X addresses deliver the result of the previous card, y^2, to both channels A and B. The result, y^4, is sent to counter group 95.

Card 3 - By the 9X address, y^4 is delivered to channel A. Result of calculation is y^4+1.

Card 4 - Contents of counter group 95, y^4, is on channel A. y^4+1, result of previous card, is on channel B. y^4+1, result of previous card, is on channel X, since this C⟶X transfer is automatic in lieu of other channel X addresses. The calculation A+B+X yields $(y^4) + (y^4+1) + (y^4+1) = 3y^4+2$.

Note that cards 1 and 2 illustrate a case in which no C address is needed.

5. Addresses - Sign Control and Absolute Value

REFERENCE: Illustration 2b

Card 1 - Calculation: -A+B = -1+2 = +1. The Y in column 2 changed the +1 on the channel A to -1. The X8 punch changes the sign of the result before entering storage register 23. Thus register 23 now contains -1.

Card 2 - Calculation: $|A|$ -B -X = 1+2-1 = +2.

Y1 punch gives $|A|$; Y6 changes sign of B. The previous result enters on channel X with its sign changed because of the X8 punch in the previous card. Counter group 92 now contains +2.

Card 3 - Calculation: A- $|B|$ +X = -2-1+2 = -1.

Y punches in columns 5 and 6 give - $|B|$. The other addresses are normal except we change the sign of the result before entering counter group 96. Thus 96 contains +1.

Card 4 - Calculation: $|A|$ + $|B|$ +X = 1+1+2 = +4.

The number in 23 is still a -1 even though we called the absolute value of this number out on channel A. The 9X B address calls out -1 (the previous result). The 9X address is not affected by the fact there was an X8 punch in the previous card. The Y5 punch gives us $|B|$ which is +1. The X7 punch tells the contents of counter group 92 to enter onto channel X. Thus the +2 stored in 92 is delivered to channel X. It should be noted that the X8 punch in the previous card did not affect this X address. It will never affect the sign of X except on the automatic C⟶X transfer. The X8 punch in this card, however, will give -X for the next card, since, as will be seen in the next paragraph, we depend on the C⟶X transfer. There is no channel C address for this card so the X8 punch affects nothing else.

Card 5 - Calculation: $-|A| + |B| - X = -4+1-4 = -7$.
Y1 and Y2 punches call for $-|A|$. The Y5 punch gives $|B|$. Channel X delivers the previous result with its sign changed. The result of this card is sent to register 23 without changing its sign.

One must always remember that the X8 punch affects nothing except what has been mentioned in this example. Thus, if we should tell the machine to do a particular thing because of the sign of the result, the machine examines this sign before an X8 punch has changed it.

6. Addresses - Multiple C addresses

REFERENCE: Illustration 3a

Card 1 - The numbers in 11 and 15 are added together and the result sent to both storage registers 11 and 15. Note that when sending to two storage registers simultaneously the double punch occurs only in column 8.

Card 2 - The result of the calculation is 4+0 = 4. The 4 is sent to both storage registers 42 and 48.

Card 3 - A 5 is sent to both counter group 92 and storage register 32. Note that in this counter group - storage register combination the double punch occurs only in column 7.

Card 4 - The contents of counter group 94 are added to the contents of storage register 64 and the result is sent to both 94 and 64 with a change in sign.

The C addresses above are independent of any A and B addresses. Every address and control punch behaves in a normal way except that the result is sent to two storages instead of one.

7. Addresses - Non-clear Counter Group Before Read-In.

REFERENCE: Illustration 3b

Assume that the number -1234567890 is stored in counter group 91 and +9876543210 is stored in counter group 97.

Card 1 - The Y7 punch tells counter group 91 not to reset before accepting the result of this calculation. The addition in this counter group then takes place as follows:

Original number in 91 ⟶	-1234567890
Number being entered into 91 ⟶	+1065432110
Resulting number in 91 ⟶	+2300000000

Note that the absolute values of the numbers were added and the sum given the sign of the new number entered.

Card 2 - Notice the X8 punch which changes the sign of the result before entering counter group 97. Thus, by applying the rules above we have

Original number in 97 →	+9876543210
Number being entered into 97 →	-7123526795
Resulting number in 97 →	-7000070005

Notice that the carry of one which results from addition of the two high order digits is completely lost.

Card 3 -

Number now in 91 →	+2300000000
Number being entered into 91 →	+1000000000
Resulting number in 91 →	+3300000000

Card 4 -

Number now in 97 →	-7000070005
Number being entered into 97 →	-1000010001
Resulting number in 97 →	-8000080006

8. Spread Read-In

REFERENCE: Illustration 4a

Card 1 - Counter group 97 is loaded with the number 1010000050.

Card 2 - Counter group 93 is loaded with the number -14.

Card 3 - This card does not <u>have</u> to be blank. The only reason it is left blank here is to fulfill the requirement that the card preceding the spread-read-in control card must not send a result of a calculation to a counter group.

Card 4 - This is the spread-read-in control card and therefore has a 90 C address. With this card the machine now knows that it should treat the next card as a spread-read-in card. The Y punches in columns 30 and 70 tell the machine <u>not</u> to reset counter groups 93 and 97 before reading in the new numbers punched on the next card.

Card 5 - This is the spread-read-in card itself. The numbers contained in counter groups 91, 92, 94, 95, 96, and 98 <u>after</u> this card are 1, -2, 4, 5, 6, and 8 respectively. Counter groups 93 and 97 are affected by the Y punches in the previous card. For these counter groups we have

Original number in 93 →	-1400000051
Number entered by spread-read-in →	+0000000000
Resulting number in 93 →	-1400000051

Original number in 97 →	+1010000050
Number entered by spread-read-in →	-7000000050
Resulting number in 97 →	-8010000100

Note that the absolute values of the numbers are added and that the sign of the sum is minus if <u>either</u> <u>or</u> <u>both</u> of the numbers are negative. It will be given a positive sign only if <u>both</u> numbers are positive.

It should be noted that the example using counter group 93 shows the way by which the original number in 93 is preserved.

Anything that can be done with one counter group can be done with another. The operations are interchangeable with respect to the counter groups.

Card 6 - Calculation: A+B+X = 4+8-2 = 10.

This card simply shows that the numbers which have just been entered into the counter groups may be used on the card immediately following the spread-read-in card. In this case A comes from 94, B from 98, and X from 92.

CAUTION: All of the preceding cards may have card numbers except the spread-read-in card itself.

9. Spread-Read-Out

REFERENCE: Illustration 4b

Card 1 - This card orders a spread-read-out by virtue of the X4 punch. Note that none of the channel addresses are for counter groups. After this card prints, the machine will punch a card which records the contents of all counter groups. The tabulator will next print a line showing the contents of all counter groups. The machine will ignore the X4 punch if Switch #3 is in A position. Only the printed record will occur if Switch #2 is in A position and Switch #3 is not in A position. This card ordering the spread-read-out may be left completely blank except for the X4 punch. Spread-read-out is, in fact, usually coded for in this way.

Card 2 - This card is completely blank. It could possibly contain a card number but that is all. Under no condition, however, will this card print.

Card 3 - This card is put in simply to show that all operations may be resumed with no restrictions after the blank card.

10. Stopping the Machine - Coded Instruction

REFERENCE: Illustration 5a

Card 1 - Because of the 1 punch in column 9 the machine will stop itself after printing the result of this card because the result of this card is negative. As pointed out previously the X8 punch changes the -1 to a +1 before entering register 12, but does not affect any decisions the machine might make with regard to stopping.

11. Stopping the Machine - Square Root and Division by Zero

REFERENCE: Illustration 5b

Card 1 - A 5 operation is $\sqrt{A}$. For this operation we must have B=A. The machine will automatically stop on this card because we are trying to take the square root of a negative number.

Card 2 - The operation is $\frac{A}{B} = \frac{1}{0}$. This attempt to divide by zero will automatically result in a machine stop.

12. Stopping the Machine - Range Violation

REFERENCE: Illustration 5c

Card 1 - The calculation is $A \cdot B = -10^{30} \cdot 10^{35} = -10^{65}$. It is not possible to represent 10^{65} as a floating decimal number since this would require an "exponent" of 115 and we have allowed room for only a two-digit "exponent" in the notation. Consequently, the machine automatically stops since an error in coding is indicated.

13. Stopping the Machine - Comparison

REFERENCE: Illustration 5d

Card 1 - The result of this calculation is 9258147036.

Card 2 - The X punch in column 25 tells the machine to compare the number punched in columns 71-80 of this card with the result of the previous card. If Switch #4 is in A position and these numbers do not compare exactly the machine stops. The number punched in columns 71-80 of this card is -9257147036.

This number does not compare with the previous result in two respects. First, the signs of the numbers are different. Second, the fourth highest order digits differ by one. The machine would have stopped if only one of these conditions had been present. There obviously can be no card number for this card since we are using column 80 for another purpose. Neither can columns 71-80 be used in the role of an instruction field at this time.

14. Stored Instructions

REFERENCE: Illustration 6a

Let us assume that the number stored in counter group 95 is 2100310050.

Card 1 - The calculation is A+B. In floating decimal arithmetic the addition is as follows.

Number in counter group 95 → 2100310050
Number read in from card → 1000100049
Result of floating decimal addition → 2200320050

The X15 punch tells the machine to use part of the number called in over channel A of this card as the A and B addresses for the next card. Let us examine the number called in on channel A of this card. It is 2100310050. The underlined section of this number is the part that will be used as A and B addresses on the next card. The fact that we have calculated with this number does not affect the function of the X15 punch. Note that we have stored the result of this calculation, 2200320050, back into counter group 95.

Card 2 - Calculation: | A | -B. The numbers stored in storage registers 21 and 31, with the modifications instructed by the Y1 and Y6 punches, are added together and the result put in register 43. The 21 and 31 addresses will be used irregardless of any digit punches in the A and B address columns.

Card 3 - The result of this calculation is 2300330050 and is stored in counter group 95. The number called in on channel A is 2200320050, the contents of counter group 95. The X15 punch then tells the machine to use the 22 as the A address for the next card and 32 as the B address.

Card 4 - The calculation is $-|A|+B$ where A and B are the contents of storage registers 22 and 32 respectively.

The function of counter group 95 in the cards above is not confined to counter groups. In fact, the X15 punch will function for <u>any</u> number called in over channel A no matter where it was stored. The number we originally assumed to be stored in counter group 95 could have been a computed number. Thus, one is able to compute A and B addresses in a way dependent on the problem itself.

15. Selected Card Suppression

REFERENCE: Illustration 7a

Card 1 - The X11 punch instructs the machine to examine the sign of the result of this calculation, a plus 1, and remember if it is positive or negative. This sign test can be made in conjunction with any calculation whatsoever.

Card 2 - The only unusual thing about this card is the X12 punch which tells the machine to ignore all punches in every instruction field of the next card if the result of the previous sign test was positive.

Card 3 - Absolutely nothing happens on this card. The instruction field is completely ignored because the sign test on card 1 was, in fact, positive as required by the X12 punch on the previous card.

Card 4 - A new sign test (X11 punch) is made on the result of this card. The X12 punch, however, still refers to the previous test and, consequently, the following card's instruction fields will be ignored even though the result of the sign test made on this card is negative.

Card 5 - Everything is ignored in this card except the X12 punch which is not in any instruction field. This X12 punch will not affect the following card, however, since the result of the sign test on card 4 was negative. The X12 punch is effectively only on a positive sign test.

Card 6 - Normal calculations take place. The X13 punch tells the machine to ignore instruction fields on the next card if the result of the previous sign test was negative.

Card 7 - Instructions are ignored by virtue of the X13 punch in card 6 and the fact that the result of card 4 was negative. The X13 punch in this card, however, is not ignored since it is not in any instruction field.

Card 8 - The 90 C address, signifying that this is a spread-read-in control card, is ignored due to the X13 punch on card 7 and the negative sign test on card 4. Consequently, the next card, which would normally be treated as a spread-read-in card, will be treated as a regular instruction card. However, the X13 punch in this card tells the machine to ignore all instructions in the next card because of the negative sign test on card 4.

Card 9 - Would have been treated as an ordinary spread-read-in card if the sign test on card 4 had been positive. In this case, however, the card is completely ignored. These last two cards show the means of selectively skipping spread-read-in cards. It is important that both cards in the spread-read-in operation be suppressed.

Any X12 or X13 punches on any succeeding cards will always refer back to the sign test made on card 4 unless a new sign test intervenes. When a new sign test is made it destroys the results of the previous test.

All instructions were ignored for card 3. As was pointed out before, these instructions would also have been ignored if they had been punched in any other instruction field besides #1.

16. Changing the Instruction Field

REFERENCE: Illustration 8a

Card 1 - Calculation: A-B = 1-2 = -1.

The double punch of X and 3 in column 9 tells the machine to start taking its instructions from instruction field 3 if the result of this calculation is negative. It is true that the X8 punch changes this result to a +1 before storing in register 24. However, the test for changing fields is done on the true result which, in this case, is negative. Therefore the machine will change to field 3. Because of the necessary delay, however, the change will not be effective until card 3.

Card 2 - This card adds the contents of storage register 13 to itself and stores the result back in 13. Note that the machine is still taking its instructions from field 1 and has not yet changed due to the instruction on card 1.

Card 3 - Calculation: A+B = 2+1 = +3.

The change of instruction fields has become effective with this card and the machine is taking its instructions from field 3. The 5 punch in column 9 of the instruction field tells the machine to change fields if the result of this calculation is positive. Since the result is positive the machine will, in fact, shift to field 2 becuase of the 2 punch in the same column. It is to be noted that zero is considered to be positive. Thus, the machine would have shifted fields if the result of this calculation had been zero.

Card 4 - The number in register 14 is added to itself and the result is sent to counter group 93. The machine is still taking its instructions from field 3 since the order of card 3 to change fields has not yet become effective.

Card 5 - The change to field 2 has now become effective. The 6 and 4 punch in column 39 (9th column of instruction field being used) tells the machine to shift to field 4 (4 punch) if the result of this calculation is zero (6 punch). The result of the calculation is zero so a change of fields will be made.

Card 6 - The contents of counter group 93 are added to the number in register 15 and the result sent to 93. Note that we have let one card intervene since a number was sent to 93. This illustrates the necessity of strictly adhering to timing considerations even when in different fields.

Card 7 - The shift ordered on card 5 has become effective. Here the operation tells the CPC to examine the integral part of the number entered on channel A, rounded according to the decimal, and, if it is odd, change instruction fields. The number entered on channel A in this case is 8.5. The machine does its rounding by always adding a 5 in the position being dropped. Thus the 8.5 becomes a 9 after rounding. The 9, being an odd number, means the machine will change fields. It will automatically shift to field 1 since there is no specific designating punch in column 79. Note that a 57 "exponent" is read onto channel B as per requirements for this operation.

Card 8 - The 8 in column 79 says to shift field unconditionally. Thus, the machine will shift fields without making any test whatsoever. The 4 punch tells it to shift to field 4.

Card 9 - We are now in field 1 due to the shift instructions on card 7. The double punch of 8 and 2 in column 9 order a shift to field 2 unconditionally.

Card 10 - Instructions are now being taken from field 4 as the result of the unconditional shift instructed on card 8.

Card 11 - The shift ordered on card 9 has been effected and we are now in field 2.

The machine will continue to take its instructions from field 2 until some other shift instruction is coded, or all cards are run out of the machine and into the hopper. The machine automatically reverts to field 1 when the cards are run out of the machine.

The last few cards show the ability to read instructions from a new field on every card if it is desired. It should be noted, however, that the same delay per individual shift is still there.

For purposes of simplicity the only instructions shown on the illustration sheet are the ones actually used by the machine due to the way instruction fields actually changed. However, for example, we could have put instructions down in field 1 for every card through card 9. Then suppose that the result of the calculation on card 1 had been positive. We would not have shifted fields at all and would have used the instructions in field 1 down to the unconditional shift of card 9. So if the results of all calculations where tests were made had been variable, the problem would have taken varying courses of calculation depending upon the set of conditions present at the time of the tests.

17. Automatic Checks and Adjustments - Range Violation

REFERENCE: Illustration 9a

Card 1 - Calculation $A \cdot B = 10^{-30} \cdot 10^{-25} = 10^{-55}$.

The result of this calculation is too small to be represented in our floating decimal notation. The "exponent" for such a number would have to be -05 (the "exponent" is obtained by adding 50 to the true exponent). We have not provided room in our floating decimal set-up to admit negative exponents. The sign of a floating decimal number always pertains to the significant digits and not to the "exponent". The above calculation is said, then, to violate the range of the notation in the downward direction. The machine in such cases converts the result to a "logical" zero and continues to calculate. The number that enters storage register 15 is 0000000000, a "logical" zero.

18. Special Functions - $\sqrt{A}$

REFERENCE: Illustration 9b

Card 1 - This card calculates the square root of the contents of storage register 13. Note that we have met the requirement B=A by reading the same number onto channels A and B.

Card 2 - The calculation is the square root of the previous result. Thus we calculate the fourth root of the contents of register 13 and store this root back in 13.

19. Special Functions - $A^n \cdot B$

REFERENCE: Illustration 9c

Card 1 - Calculation: $A^n \cdot B = (.9)^{13}(2)$.

Note that the 13 is read from channel X as a fixed decimal number in the low order positions of the significant digits. The "exponent" of channel X means nothing. It was arbitrarily made zero in this case.

Card 2 - The number in register 54 is raised to the 4th power and multiplied by itself to give the 5th power. The 4 is read in over channel X the same as the 13 on the previous card.

If a number is to be raised to the 2756th power, the number read in over channel X must be 000275600. The preceding case is not very probable and is brought out here simply to show the continuing pattern of reading in exponents.

This exponent, n, may even be a calculated number. The procedure for doing this is shown in the next example which illustrates the 1, 8 operation, a manipulative function.

20. Manipulative Functions - "1, 8" Operation

REFERENCE: Illustration 9d

Card 1 - The difference in "exponents" of the numbers entered over channels A and B is 55-49 = 6. The result of the calculations is 0000001755. Notice that the significant digits of A have been shifted to the right 6 digits (difference of exponents is 6) and rounded according to the 7, which is the last digit shifted off. The exponent of B is now the exponent of A.

Card 2 - Let us assume that the contents of storage register 17 is 8000000050. Assume also that we want to perform the calculation A^n where n is the positive integer stored in register 17 and A is the number stored in register 54. By reading a 57 "exponent" in over channel B and doing the 1, 8 operation we find that the result of the calculation is 0000000857. Note that this is in perfect form to be used as an "n" in the calculation $A^n \cdot B$.

Card 3 - With no specific X address the number on channel X is automatically the result of the previous calculation, 0000000857. We now do the operation $A^n \cdot B$ where A is the contents of register 54 and B is one. "n", as read over channel X, is eight. The problem of A^n where n is the positive integer stored in 17 is complete.

Note the use of the 57 "exponent" on channel B. If the number in register 17 had been 2700000051 the result of the calculation would have been 0000002757 which also is in proper form to be read onto channel X for the calculation A^{27}.

21. Manipulative Functions - "1, 9" Operation

REFERENCE: Illustration 10a

Card 1 - The number on channel A is 45698.721. The result of the calculation is 4569800054 which is the integral part of A. However, we obtain the integral part because we were careful to read a 57 "exponent" over channel B. The "exponent" of B minus the "exponent" of A is 57-54 = 3. Thus the three least significant digits of A were made zero as indicated above. Other "exponents" could have been used on channel B. A 57 was used for <u>this</u> example to show how to obtain the <u>integral</u> part of a number. Note especially that <u>no</u> rounding was done.

Card 2 - Calculation: A-B

Number on channel A →	4569872154
Number on channel B →	-4569800054
Result of subtraction →	7210000049

This card simply shows the logical way of obtaining the decimal part of the number after the integral part has been found.

22. Manipulative Functions - "7" Operation

REFERENCE: Illustration 10b

Card 1 - The result of the calculation is the exponent of A attached to the significant digits of B. The result, then, is 1000000057.

23. Manipulative Functions - "7, 8" Operation

REFERENCE: Illustration 10c

Card 1 - The "exponent" of A minus the "exponent" of B is 52-36 = 16. The result of the manipulation is this difference in floating decimal form, namely, 1600000051.

24. Manipulative Functions - "1, 6, 8" Operation

REFERENCE: Illustration 10d

Card 1 - The required 57 "exponent" is read onto channel B. The integral part of A, rounded according to the decimal part, is 5. This number added to the "exponent" of X (50) gives 55. Thus the "exponent" of the result is 55. One may now use the "7" operation to attach this calculated "exponent" to the significant digits of some number. This operation is used in calculating the logarithm of a number. (see example for log x).

25. Discriminatory Functions - Comparison

REFERENCE: Illustration 11a

Card 1 - The result of this "2, 9" operation is 1462769950 which is the smaller, algebraically, of the numbers entered over channels A and B.

Card 2 - The result is a -2 which is the lesser of A and B.

Card 3 - The result is the greater of A and B, namely a +1.

Card 4 - Here we call in the absolute values of the numbers on A and B. The greater of the numbers is +2 which appears on channel C as the result.

26. Transcendental Functions - Sin x, Cos x, Sinh x, Cosh x

a. Sin x, Cos x

Series Expansion Used for Sine x and Cosine x:

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots$$

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots$$

REFERENCE: Illustration 15a

Card 1 - Calculation of the sine of .75 radians. Channel A contains the argument (.75). Channel B has the required 1000000012. The 12 (this is n in the coding) tells the machine how many terms to calculate in the series expansion. Note that 12 is an even number as per coding requirements. Calculation out to the term including x^{11} is sufficient for eight figure accuracy. If the coder does not need that much accuracy he may reduce the number of terms with a resultant saving of time.

Card 2 - Calculation of the cosine of .75 radians. The same discussion applies here as applied to the calculation of sine .75 radians. Note however, that n is odd as required, and large enough for eight figure accuracy.

This illustration assumes that x is in radians and $0 \leq x < 1$. Most other examples of series coding will include a reduction routine to insure that the argument is within the limits necessary before actually entering the series calculations. The reduction for an arbitrary argument in the case of sine and cosine, however, is relatively bulky. It was thought best to let the coder program his own reduction. Thus if the coder knows his argument will always lie between 0 and $\pi/2$ the reduction is quite simple and the programming will be much shorter than for a completely general case. However, the general reduction is available from the author if anyone wants it.

b. Sinh x, Cosh x

Series Expansion Used for these Functions;

$$\sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \ldots$$

$$\cosh x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \ldots$$

REFERENCE: Illustration 15b

Card 1 - Calculation of sinh of .5. n = 12 is sufficient for 8 figure accuracy.

Card 2 - Calculation of cosh of .5. n = 11 is sufficient for 8 figure accuracy

There is no easy direct reduction for sinh x and cosh x. General rules for series coding apply here just as for sin x and cos x.

27. Transcendental Functions — e^x

Series Expansion Used for e^x

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$$

REFERENCE: Illustration 12a

The entire programming in this illustration is that which is required to calculate e^x for an arbitrary argument. It is assumed that x is stored in register 11 and we wish to store e^x back into 11 at the end of the programming. No discussion is given of the type of argument reduction employed since we are not dealing with mathematical methods in this manual. If one uses the programming outlined, he must observe that counter groups 91 and 92 are used for temporary storage.

Card 5 - At this point we have an argument reduced so that $x_1^2 < 1$. We enter the argument on channel A. The operation orders the calculation of e^{x_1}. The 14 "exponent" on channel B tells the machine to take 14 terms of the exponential series which is sufficient for eight figure accuracy. x_1 is the reduced argument.

28. Transcendental Functions — $\frac{1}{2}\log\frac{1+x}{1-x}$

Series Expansion Used for $\frac{1}{2}\log\frac{1+x}{1-x}$:

$$\frac{1}{2}\log\frac{1+x}{1-x} = x + \frac{x^3}{3} + \frac{x^5}{5} + \ldots$$

REFERENCE: Illustration 13a

The programming assumes the arbitrary argument is in register 11. The calculation performed is log $|x|$ which is put back into 11. Counter groups 91 and 92 are used as temporary storages during the routine.

Card 5 - This card calculates $\frac{1}{2}\log \frac{1+x}{1-x}$ using 30 terms (the B "exponent") of the series expansion. It is assumed that $x^2 < 1$ where x is the argument entered on channel A.

29. Transcendental Functions - arctan x

Series Expansion Used for arctan x:

$$\arctan x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \ldots$$

REFERENCE: Illustration 14a

This programming calculates arctan x for an arbitrary x. Register 11 is assumed to contain x and after the calculation we store arctan x back into 11. Counter groups 91 and 92 are used for temporary storage. Even though use is made of instruction field 2 we return to field 1 after calculation of the function.

Card 6 - The reduced argument entered on channel A is used in calculating arctan x. The B "exponent" tells us that 24 terms will be used.

ILLUSTRATION SHEET 1

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer X-Read In	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F				A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11——20	21——30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51——60	61——70	1	2	3	4	5	6	7	8	9	10
										Illustration 1a																																	
9	3	1		9	3	9	3																																				
2	2	1		2	4	2	2																																				
9	3	1		2	4	9	1																																				
1	1	1		1	1	1	1																																				
9	3	1	1	X 2	2	2	4																																				
9	3	1	1	9	8	X 2	2																																				
										Illustration 1b																																	
9	0	1		9	0	1	1			X 1000000050																																	
9	9	1		9	9	9	1																									2000000050	3000000050										
9	0	Y 1	1	9	9	4	3																										X 4000000050										
9	9	Y 1	1	9	0	9	2				1000000050																					5000000050	6000000050										

ILLUSTRATION SHEET 2

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11——20	21——30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51——60	61——70	1	2	3	4	5	6	7	8	9	10
										Illustration 2a																																	
1	2	3		1	2																																						
9	X	3		9	X	9	5																																				
9	X	1		9	0						1000000050																																
9	5	1	1	9	X	1	2																																				
										Illustration 2b																																	
9	Y 0	1		9	0	2	X 3			1000000050	2000000050																																
Y 9	0	1	1	9	Y 0	9	2			X 1000000050	X 2000000050																																
9	0	1	1	Y 9	Y 0	9	X 6			X 2000000050	X 1000000050																																
Y 2	3	1	1	Y 9	X	X	X																																				
Y 9	Y X	1	1	Y 9	6	2	3																																				

ILLUSTRATION SHEET 3

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4	Ctr Group 5	Ctr Group 6	Ctr Group 7	Ctr Group 8
Instruction Field 1										A Normal	B Normal	Instruction Field 2	Instruction Field 3	A Transfer	B Transfer X-Read In	Instruction Field 4
A		O		B		C		F				A O B C F	A O B C F			A O B C F
1	2	3	4	5	6	7	8	9	10	11——20	21——30	1 2 3 4 5 6 7 8 9 10	1 2 3 4 5 6 7 8 9 10	51——60	61——70	1 2 3 4 5 6 7 8 9 10
										Illustration 3a						
1	1	1		1	5	1	1 5									
9	0	1		9	0	4	2 8			4000000050						
9	0	1		9	0	9 3	2			5000000050						
9	4	1		6	4	9 6	X 4									
										Illustration 3b						
9	0	1		9	0	Y 9	1			1065432110						
9	0	1		9	0	Y 9	X 7			7123526795						
9	0	1		9	0	Y 9	1			1000000000						
9	0	1		9	0	Y 9	7			X 1000010001						

ILLUSTRATION SHEET 4

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11—20	21—30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51—60	61—70	1	2	3	4	5	6	7	8	9	10
										Illustration 4a																																	
9	0	1		9	0	9	7			1010000050																																	
9	0	1		9	0	9	3			X 1400000051																																	
		BLANK																																									
						9	0				Y																						Y										
1	0	0	0	0	0	0	0	5	0	X 2000000050		4	0	0	0	0	0	0	0	5	0	5	0	0	0	0	0	0	0	5	0	6000000050	X 7000000050	8	0	0	0	0	0	0	0	5	0
9	4	1	1	9	8	X 2	2																																				
										Illustration 4b																																	
2	3	1	X	4	5	3	6																																				
			BLANK																																								
9	1	1		4	4	4	5																																				

ILLUSTRATION SHEET 5

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11—20	21—30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51—60	61—70	1	2	3	4	5	6	7	8	9	10
										Illustration 5a																																	
9	0	1		9	0	1	X 2	1		X 1000000050																																	
										Illustration 5b																																	
9	0	5		9	0					X 1000000050	X 1000000050																																
9	0	4		9	0					1000000050																																	
										Illustration 5c																																	
9	0	3		9	0					X 1000000080	1000000085																																
										Illustration 5d																																	
9	0	1		9	0					9258147036																																	
1	4	1		9	0	9	2				X 4913611849																							9	2	5	7	1	4	7	0	3	X 6

ILLUSTRATION SHEET 6

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer X-Read In	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F				A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11——20	21——30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51——60	61——70	1	2	3	4	5	6	7	8	9	10
										Illustration 6a																																	
9	5	1		9	0	9	5			0000X00000	1000100049																																
Y		1			Y	4	3																																				
9	5	1		9	0	9	5			0000X00000	1000100049																																
Y	Y	1				4	4																																				

-78-

ILLUSTRATION SHEET 7

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11——20	21——30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51——60	61——70	1	2	3	4	5	6	7	8	9	10
										Illustration 7a																																	
9	0	1		9	0					X 1000000050																																	
1	1	1		1	2	1	3			X 0000000000																																	
5	Y 7	1	1	X 9	X	1	X 4																																				
9	0	1		9	0					XX X 1000000050																																	
1	3	1		1	2	1	3			X 0000000000																																	
2	1	1		2	2	9	3			X 0000000000																																	
4	7	1		2	4	6	7			X 0000000000																																	
						9	0			X 0000000000																																	
1	1	1	1	1	1	1	1	5	0	2222222250	X 3333333350	4	4	4	4	4	4	4	4	5	0	5	5	5	5	5	5	5	5	5	0	6666666650	7777777750	8	8	8	8	8	8	8	8	5	X 0

ILLUSTRATION SHEET 8

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11—20	21—30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51—60	61—70	1	2	3	4	5	6	7	8	9	10
										Illustration 8a																																	
9	0	1		9	Y 0	2	X 4	X 3		1000000050	2000000050																																
1	3	1		1	3	1	3																																				
											1000000050											9	9	1		9	0	2	5	5 2		2000000050											
																						1	4	1		1	4	9	3														
												9	9	1		9	9			6 4																						
												9	3	1		1	5	9	3																							
										8500000050																						57	9	0	1	5 8	9	9				
										1000000050																							9	0	1		9	0	2	6	8 4	
9	0	1		9	0			8 2		X 1000000050																																
																																	9	3	1		9	3	9	5		
												1	4	1		1	5	9	3																							

ILLUSTRATION SHEET 9

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11—20	21—30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51—60	61—70	1	2	3	4	5	6	7	8	9	10
										Illustration 9a																																	
9	0	3		9	0	1	5			1000000020	1000000025																																
										Illustration 9b																																	
1	3	5		1	3																																						
9	X	5		9	X	1	3																																				
										Illustration 9c																																	
9	0	Y 3	6	9	0	9	6			9000000049	2000000050																						0000001300										
5	4	Y 3	6	5	4	5	4																										0000000400										
										Illustration 9d																																	
9	0	1	8	9	0					1670481149	1000000055																																
1	7	1	8	9	0						57																																
5	4	3	6	9	0	5	4				1000000050																																

-81-

ILLUSTRATION SHEET 10

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer X-Read In	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F				A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11—20	21—30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51—60	61—70	1	2	3	4	5	6	7	8	9	10
										Illustration 10a																																	
9	0	1	9	9	0	1	1			4569872154	57																																
9	0	1		9	Y X	1	2			4569872154																																	
										Illustration 10b																																	
9	0	7		9	0					8716632157	1000000050																																
										Illustration 10c																																	
9	0	7	8	9	0					1 52	1 36																																
										Illustration 10d																																	
9	0	Y 1	6 8	9	0					4600000050	57																						100000050										

ILLUSTRATION SHEET 11

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11——20	21——30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51——60	61——70	1	2	3	4	5	6	7	8	9	10
										Illustration 11a																																	
9	0	2	9	9	0					1462813450	1462769950																																
										X	X																																
9	0	2	9	9	0					2000000050	1000000050																																
			7							X																																	
9	0	2	9	9	0					2000000050	1000000050																																
Y			7	Y						X																																	
9	0	2	9	9	0					2000000050	1000000050																																

ILLUSTRATION SHEET 12

Ctr Group 1										Ctr Group 2	Ctr Group 3
Instruction Field 1										A Normal	B Normal
A		O		B		C		F			
1	2	3	4	5	6	7	8	9	10	11——20	21——30
										Illustration 12a	
1	1	4		9	0	9	1				2302585150
9	X	1	9	9	0	9	2				57
9	1	Y 1	3	9	Y X						
9	X	4		9	0						3000000050
9	X	6		9	0						1000000014
9	X	3	3	9	X	9	1				
9	2	1	6 8	9	0						57
9	X	7		9	1	1	1				

Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 2										Instruction Field 3										A Transfer	B Transfer X-Read In	Instruction Field 4									
A		O		B		C		F		A		O		B		C		F				A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51——60	61——70	1	2	3	4	5	6	7	8	9	10

ILLUSTRATION SHEET 13

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F			X-Read In	A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11—20	21—30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51—60	61—70	1	2	3	4	5	6	7	8	9	10
										Illustration 13a																																	
9	0	7		Y 1	1	9	1			50																																	
Y 1	1	7	8	9	Y X	9	2																																				
9	1	1		9	Y 0						3000000050																																
9	1	1	4	9	0						3000000050																																
9	X	6	8 9	9	0						1000000030																																
9	0	3 9	1	9	0					2000000050	1098612350																																
9	2	3	1	9	0	1	1				2302585150																																

ILLUSTRATION SHEET 14

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11——20	21——30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51——60	61——70	1	2	3	4	5	6	7	8	9	10
										Illustration 14a																																	
9	0	1		Y 1	Y 1			X 2		1000000050																																	
1	1	4		Y 1	1	9	1																																				
										1000000050		9	0	4		Y 1	1	9	2																								
Y 1	1	Y 3	1	9	0						4140000049	9	X	Y 3	1	9	0																1000000050										
Y 1	1	1	2 4	9	0						X 4140000049	9	2	1	2 4	9	0																										
9	X	6	7 8 9	9	0						1000000024	9	X	6	7 8 9	9	0			8																							
9	X	1		9	0						3925167849	9	Y X	Y 1	1	9	Y 0																1570796350										
9	X	3		9	1	1	1																																				

ILLUSTRATION SHEET 15

Ctr Group 1										Ctr Group 2	Ctr Group 3	Ctr Group 4										Ctr Group 5										Ctr Group 6	Ctr Group 7	Ctr Group 8									
Instruction Field 1										A Normal	B Normal	Instruction Field 2										Instruction Field 3										A Transfer	B Transfer	Instruction Field 4									
A		O		B		C		F				A		O		B		C		F		A		O		B		C		F			X-Read In	A		O		B		C		F	
1	2	3	4	5	6	7	8	9	10	11—20	21—30	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	10	51—60	61—70	1	2	3	4	5	6	7	8	9	10
										Illustration 15a																																	
9	0	6	7 8	9	0					7500000049	1000000012																																
9	0	6	7 8	9	0					7500000049	1000000011																																
										Illustration 15b																																	
9	0	6	8	9	0					5000000049	1000000012																																
9	0	6	8	9	0					5000000049	1000000011																																

-87-

4F SUPPLEMENT

INTRODUCTION:

This section assumes that one has in mind all essential coding details which have been described in the first three parts of this manual.

The 4F control panel is a panel designed by Dura Sweeney to go into the electronic calculator. By simply putting Switch #6 in the A position, the regular #4 control panels will be compatible with the new 4F panel in the electronic calculator.

Differences in coding occur only with regard to operations. Channel addresses, sign control, changing instruction fields, and all other functions remain the same.

As pointed out earlier this panel is designed in such a way that almost all combination operations will take place in one card time (.4 seconds). Consequently, we have approximately 50% more effective computing time for a typical problem employing a modest amount of combination operations. In extreme cases this figure can approach 100%.

To get the speed described above it was necessary to eliminate some operations found on the general #4 board. A list of these restrictions appear in the next section to enable one to better judge if he can use the 4F control panel to advantage. In general, it will be found that the 4F panel will be well worth the using even though some of the conveniences of the general #4 panels are lost.

RESTRICTIONS

Operations and functions not available when using the 4F panel.

1. A few combination operations are lost (see table of functions).

2. No special function of $A^n \cdot B$.

3. No A ⇄ X interchange for operations.

4. No built-in transcendental functions.

   NOTE: It is extremely easy to card program these functions. The procedure and advantages are brought out in a following section.

5. No discriminatory functions. This includes the operation which allows a change of instruction field if A is odd.

6. The "exponent" of the logical zero is one. The significant digits are always zero. Calculations with this zero are, however, exactly the same as with the logical zero on the #4 panels.

7. Automatic checks and adjustments.

   a. No stop if division by zero is attempted

   b. No stop if square root of negative number is attempted

   c. No stop if a range violation in upward direction

   d. No automatic conversion to a logical zero if range is violated in the downward direction.

OPERATIONS:

a. Arithmetic Functions

Coding		Function	Coding		Function
Col 3	Col 4		Col 3	Col 4	
1	-	A+B	3,9	1	(A·X)+B
3	-	A·B	3	3	A·B·X
4	-	A/B	3	4	$\frac{A \cdot B}{X}$
1	3	(A+B)X	4	1	$\frac{A}{B}$ + X
1,9	3	(A+X)B	4,9	1	$\frac{A}{X}$ + B
1	4	$\frac{A+B}{X}$	4	3	$\frac{A \cdot X}{B}$
1,9	4	$\frac{A+X}{B}$	4	4	$\frac{A}{B \cdot X}$
3	1	(A·B)+X			

The rules for forming these functions are the same as for the #4 panel except that a 4 in column 4 is $\frac{I}{X}$. Note that this is equivalent to the 2 and 4 in column 4 for the operations discussed in Part II. The 9 in column 3 means, as usual, to interchange B and X before calculating. An 8 in column 3 is not an A and X interchange but a manipulative operation to be described later. No "created" functions are possible with this control panel.

b. Special Function

A 5 in column 3 instructs $\sqrt{A}$. A requirement is that B=A for this operation.

c. Manipulative Operations

Coding		Operation
Col 3	Col 4	
2*		No rounding is done on a single arithmetic operation. Rounding is done for the first operation of a combination operation but not on the second.
	2*	Results usually come out of the machine <u>true</u> floating decimal representation, i.e., the highest order digit of the significant digits in a representation is not zero. This operation when used in conjunction with arithmetic operations allows the leading digits to be zero as a result of a calculation. Example: A = 1234567850 B = 1234445450 A-B = 0000122450 This happens only if there was a 2 in column 4. If there was not a 2 in column 4 the result would have been 1224000046 which is the <u>true</u> floating decimal representation.
6		The result is the "exponent" of A in floating decimal representation.
7		The result is the "exponent" of <u>B</u> attached to the significant digits of <u>A</u>.
8		The two high order significant digits of B, minus 50, are added to the "exponent" of A. The result is A with its altered "exponent". Example: A = 1234567849 B = 5400000051 C = 1234567853

* A 1 and 2 in column 3 is equivalent to the "1, 9" operation of the #4 panels. A 1 in column 3 and a 2 in column 4 is equivalent to the "1, 8" operation of the #4 panels.

d. Transcendental Functions

Transcendental functions may be evaluated only by card programming with operations described above. Because this control panel can do a combination operation in the same time as a single arithmetic operation, it is possible to card program the common functions with not more than a one or two second loss per function as compared to other control panels.

If there are not too many of these functions in a problem it will be found wise to live with this small loss of time in computing the functions since the time will probably be more than made up on combination operations. The card programming has been worked out for the common functions and is available to anyone wanting them.

GLOSSARY

GLOSSARY

ADDRESS - The number designating the location of a quantity to be used in machine computation.

CHANNEL - A path within the machine over which a number may pass.

CHANNEL READ-IN FIELD - A group of card columns from which a number is read onto a channel.

CLEAR - The process of resetting a storage to all zeros.

CODER - One who translates a mathematical problem into machine language in preparation for machine computation.

CODING - Machine language.

COMBINATION OPERATION - An operation which performs two arithmetic computations on three factors.

CONTROL PUNCHES - Individual punches in a card controlling special machine operations.

COUNTER GROUP - A type of storage in the CPC.

CPC - Card Programmed (Electronic) Calculator.

CR SYMBOL - A character that prints from the type bar which is equivalent to the asterisk.

DECK - A stack of punched cards used to instruct the machine in a calculation.

DOUBLE POLE DOUBLE THROW SWITCHES - The group of switches labeled 1 through 9 on the left side of the tabulator.

DOUBLE PUNCH - Two digit punches in the same column of a card. Sometimes this phrase is used to mean two or more punches in a column.

ELECTRONIC CALCULATOR (type 605) - The component of the CPC which performs the calculation.

"EXPONENT" - The last two digits in the floating decimal representation of a number.

HOPPER - A receptacle on the tabulator located below the card feed which receives the cards after they have been completely run through the machine.

ICE BOX - A term referring to one of the auxiliary storage units (type 941) which may be attached to the C.P.C.

IDENTIFICATION - Usually used to mean punches in the card associated with a particular card or problem. The card number is one form of identification.

INSTRUCTION CARD - A card in the deck which tells the machine where to get factors A and B, what calculation to perform on them, and where to store the result. It also contains any special instructions such as double spacing or field shifting. A spread-read-in card is not an instruction card because there are no addresses or operations on this card.

INSTRUCTION FIELD - A group of card columns containing A, B, and C addresses, operations, and special control punches.

LEAST SIGNIFICANT DIGIT - The right most (lowest order) digit in the significant digits of a number.

LISTING - The rows of numbers recorded on the printed sheet.

LONG LISTING - A listing which records pertinent data for all cards of the deck.

MOST SIGNIFICANT DIGIT - The left most (highest order) digit in the significant digits of a number.

MULTIPLE PUNCH - Two or more punches in a card column.

NON-CLEAR - Not resetting a storage to all zeros.

PROGRAMMING - A set of machine instructions (coding) which solves a problem or part of a problem.

READ - Machine sensing of a number which is punched in the card or contained in a storage.

READ-INTO - The transfer of a number into some location of the machine.

READ-OUT OF - The transmittal of a number from some location in the machine.

SET-UP CHANGE SWITCHES - The group of switches labeled 1 through 3 on the left side of the tabulator.

SHIFT FIELDS - The process of changing from one instruction field to another.

SHORT LISTING - A listing which records pertinent data for selected cards in a deck.

SPREAD-READ-IN - The method of loading numbers into all eight counter groups in one operation.

SPREAD-READ-OUT - A process of recording the contents of all counter groups in printed form and on punched cards.

SRI - Abbreviation for spread-read-in.

SRO - Abbreviation for spread-read-out.

STORAGE - A location in the machine where a coder may keep constants or intermediate results. This is a general word referring to both counter groups and storage registers.

STORAGE REGISTER - A type of storage in the CPC.

STORAGE UNIT (type 941) - A component of the CPC which contains 16 storage registers.

SUMMARY PUNCH UNIT - The component of the CPC which punches the cards on a spread-read-out operation.

TABULATOR - The controlling unit of the CPC. This component reads instructions punched in the cards and sets up the necessary controls to carry them out.