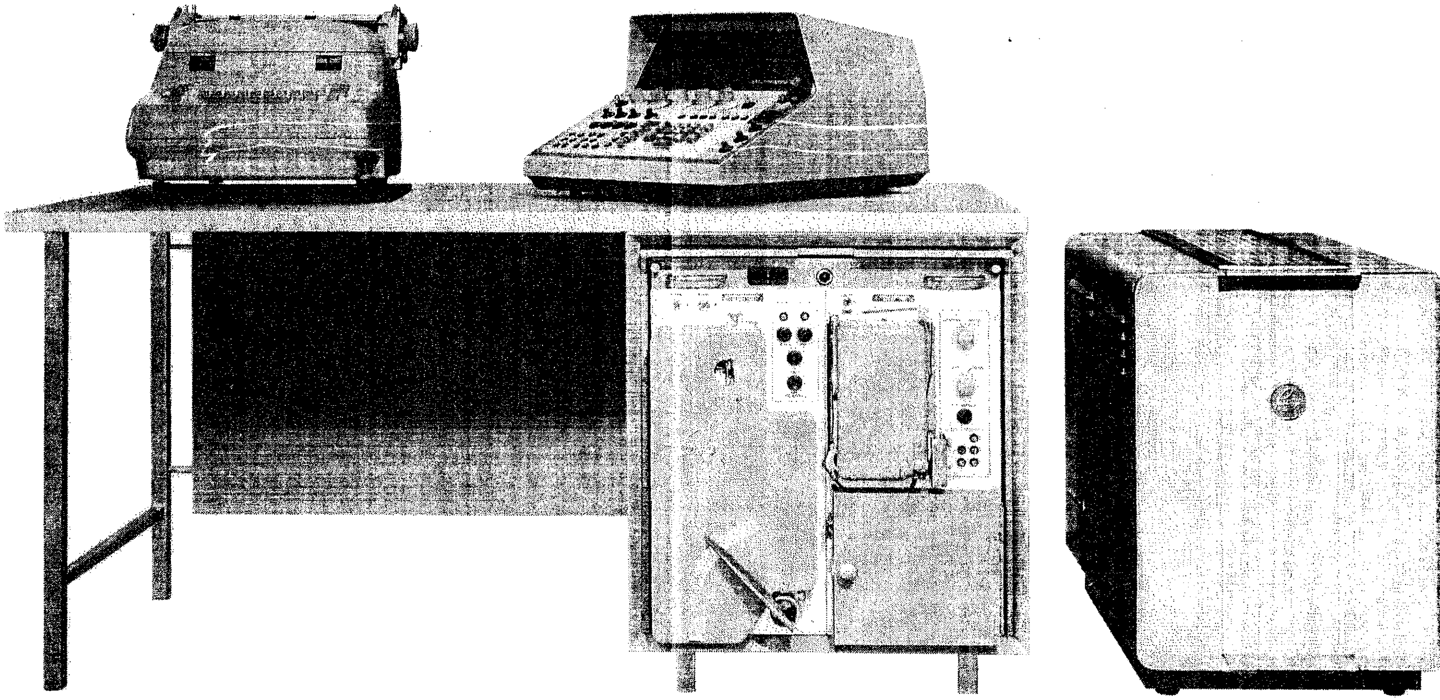
RECOMP II



OPERATING MANUAL

Autonetics Industrial Products
A DIVISION OF NORTH AMERICAN AVIATION, INC. 3584 WILSHIRE BOULEVARD, LOS ANGELES 5, CALIFORNIA

OPERATING MANUAL

FOR

RECOMP II

The information contained in this manual is generally applicable to all RECOMP II computers. However, minor changes in machine characteristics may be incorporated in individual machines.

Autonetics Industrial Products
A DIVISION OF NORTH AMERICAN AVIATION, INC. 3584 WILSHIRE BOULEVARD, LOS ANGELES 5, CALIFORNIA

Pub. 512-E -3
Rev. 5 -59

CONTENTS

Page

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

CONTENTS (Continued)

ILLUSTRATIONS

INTRODUCTION TO RECOMP II

RECOMP II is a fully transistorized, general-purpose, digital computer system featuring small-computer compactness and operating economy with the power and versatility of medium-scale systems. Designed and built by Autonetics, a division of North American Aviation, Inc., it is capable of handling the entire range of applications encountered in computation, data reduction, and control. System components are the computing, control, and memory units housed in a single cabinet, a photoelectric paper tape reader, paper tape punch, on-line electric typewriter, and control console with numerical entry keyboard. (See Fig. 1 - frontispiece and Fig. 2.)

The RECOMP II system has numerous features which increase its usefulness and make it easy to program and operate:

INPUT-OUTPUT AND STORAGE

A wide variety of input-output procedures gives the programer-operator extensive selection in problem-handling methods. Information may be put into the computer by the paper tape reader, typewriter, or console keyboard. Output may be recorded by the paper tape punch or typewriter, or presented on the console numerical display. Input-output may be in the following forms:

1. Instructions and storage locations in octal-binary format
2. Numeric data consisting of integers, fractions, and mixed numbers in decimal or octal format
3. Alphabetic or combined alphanumeric data in teletype code or printed format

Accurate reading and storage of information can be assured by a Verify operation which checks data in storage with that on the input tape. Correct output recording is ensured by an automatic echo check which verifies the data typed or punched against that transmitted from the computer. An error detected in either the Verify mode or echo check stops operations.

Capacity of the RECOMP II memory (4096 forty-bit words) enables permanent storage of many routines and constants in addition to providing ample working storage for large problems. This reduces the frequency of input-output operations, thereby simplifying programing and enabling more efficient use of computer time. Storage for 16 of the 4096 words consists of two 8-word, rapid-access loops. Use of these loops can materially shorten solution time, particularly in problems involving numerous short, repetitive operations.

LOGIC AND CONTROL

RECOMP II is an internally stored program computer. This design enables ready modification of computer programs and eliminates the necessity for making mechanical changes on the computer to accommodate new programs and applications. Single-address logic, with instructions stored two per word, simplifies programing. Programing and operation of RECOMP II can be readily learned by a nonprofessional programer or computer operator.

RECOMP II has both arithmetic and decision-making abilities and can operate on numeric, alphabetic, and combined alphanumeric information. This capability allows work on problems with varied data characteristics.

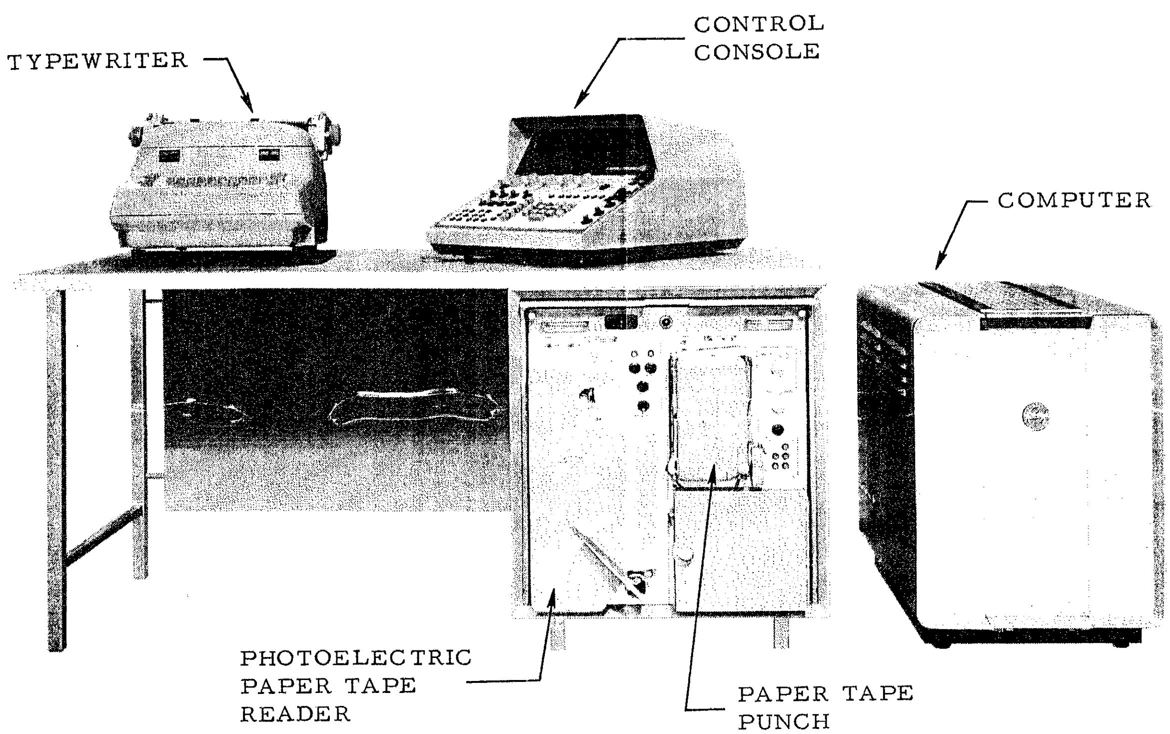


Fig. 2. RECOMP II computer components

A repertoire of 49 instructions, together with a trapping mode of operation which facilitates use of a great number of pseudo (routine-referral) commands, gives RECOMP II the power and problem-handling flexibility of many large-scale machines. The trapping mode also facilitates "debugging" of programs.

Numerous conditional and unconditional transfer instructions extend the decision-making abilities of the computer and increase the programer's choice in program course deviation. Single and continuous instruction execution modes give additional flexibility in problem-handling procedures. The single execution mode also facilitates program modification and evaluation of intermediate results. Two distinct operating states - Compute and Input - simplify timing considerations for the programer. An ability to interrupt computations manually or under program control facilitates program modification and insertion of higher-priority problems.

ARITHMETIC

The built-in floating point arithmetic of RECOMP II facilitates programing, especially on problems involving computations of a scientific nature. Scaling of numbers is simplified. Numbers may vary in magnitude without overflow occurring, and operations may be performed on much larger numbers than in fixed point arithmetic. The 40-bit word length permits solution of problems containing numbers of large magnitude. Some computer operations combine two of the arithmetic registers to form a 78-bit word for double precision.

ENVIRONMENTAL

RECOMP II can be easily moved for closer coordination on computer program development between programer and, for example, engineer. Weight of the computer itself is only 197 pounds. Normal warmup time is about 30 seconds. Special air conditioning is usually unnecessary because the computer will operate in an ambient temperature as high as 110 F. It can operate on standard, unregulated source current (115 volts, 50 to 60 cycles per second) at only 500 watts. Should source current vary to below 103 volts or above 125 volts, a light will indicate possibility of error.

Many additional features and advantages of RECOMP II will become apparent to the programer-operator as he reads this manual. It is suggested that he make special note of these to further their use at every opportunity.

BASIC ELEMENTS AND OPERATING PRINCIPLES

General-purpose digital computers can perform a large number of different operations and calculations by use of the basic arithmetic operations of addition and subtraction. The procedure for completing these operations is governed by a control unit, with the type of operation dependent upon a set of instructions or commands placed into the control unit. A change of instructions may be made without requiring any physical change to the computer.

Basically, operation of a digital computer is dependent upon five elements: a memory (storage) unit, an arithmetic unit, a control unit, an input device, and an output device. (See Fig. 3.) Although the operation and interconnections between these five elements are not the same for all general-purpose computers, the following description will apply to many computers, including RECOMP II. Usually one or more of these elements are not physically separate units, but functionally all are generally considered individual entities.

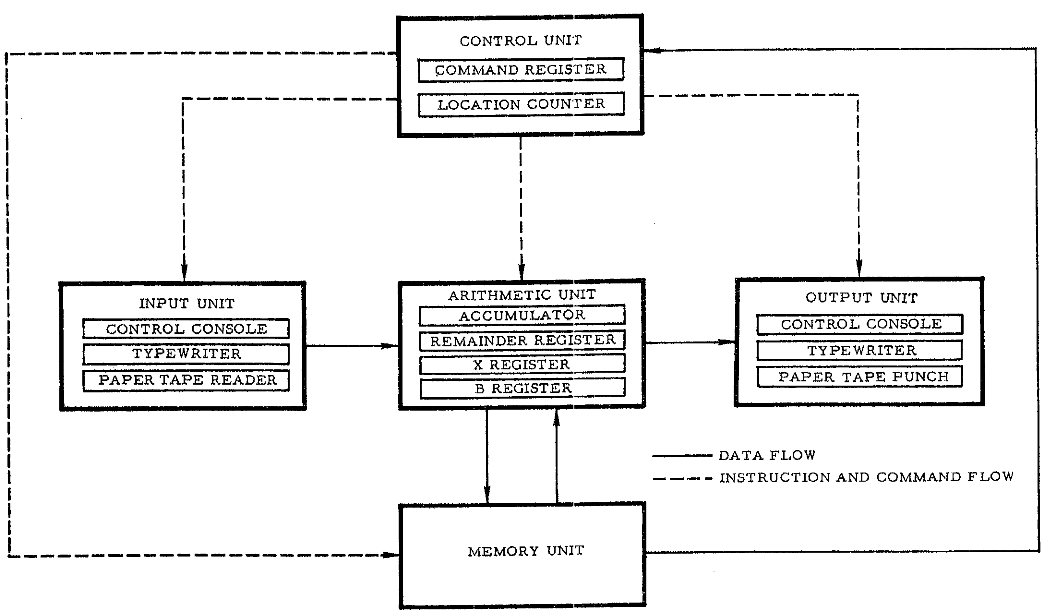


Fig. 3. Basic computer system block diagram

The procedure followed by RECOMP II in performing operations upon data is essentially as follows: Commands and numbers required for the solution of a problem are loaded into the memory, and the location counter is set to the address of the first instruction to be executed. When the start signal is received, the contents of the memory cell whose address is in the location counter are transferred into the command register. This initial command is then executed, and the setting of the location counter is increased by one digit in the half-word position to indicate the next command. After the two instructions in a word have been performed, the next instruction pair is transferred into the command register and executed. Processing and computation on data proceed, with commands being carried out in the numerical sequence of their addresses, except when a transfer of control instruction is encountered. During execution of a transfer of control command, the address portion of the instruction is placed in the location counter, thus establishing a new starting point for the following sequence of commands. In a conditional transfer of control command, the transfer occurs only if a specified condition is true. When computation is completed, output instructions read specified results from the memory and transmit them to an output device.

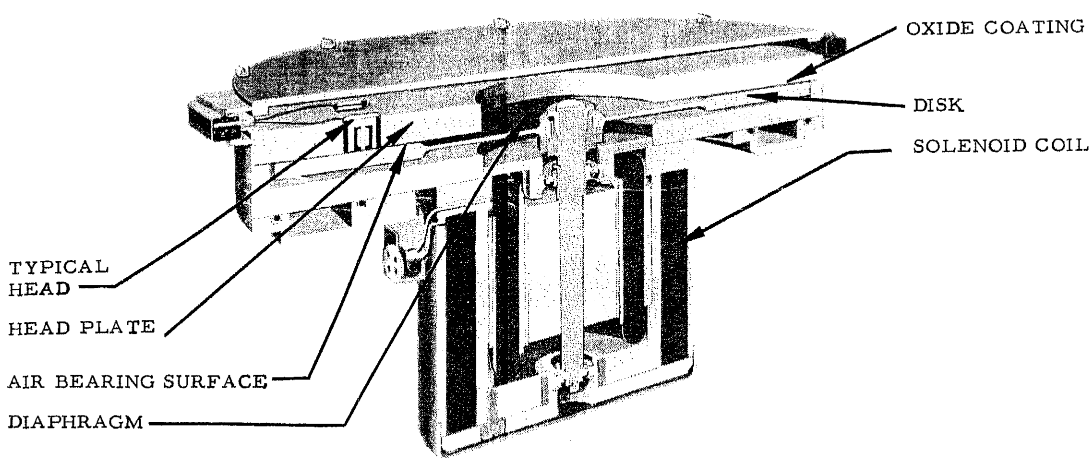


Fig. 4. RECOMP magnetic memory unit

Memory

The RECOMP II memory unit consists of a magnetic disk which rotates at 3450 rpm past stationary read and write heads. (See Fig. 4.) On the disk is a thin coating of ferrous oxide similar to that on conventional magnetic recording tape. As the disk rotates past each recording (write) head, a magnetic signal is recorded into the oxide coating and remains until replaced by new information. To extract information from the disk, a reproducing (read) head similar to the recording head is used. Extraction of information from the memory does not change its contents. The memory is divided into channels and sectors to facilitate locating information.

Arithmetic

The arithmetic unit consists of several 1-word recirculating registers - the accumulator, number register, remainder register, and X register, together with the appropriate switching and control elements for carrying out basic arithmetic and logical operations. (Registers are devices for retaining information.)

The most important of the temporary storage registers is the accumulator, often called the "A register." The accumulator is a register which holds the result of each arithmetic operation. Because the result of one operation may be used as an operand during the execution of the next instruction, the accumulator may also be considered to contain one of the operands involved in the execution of an instruction.

The number register, often referred to as the "B register," is a storage register into which words read from the main memory are transferred prior to the execution of an arithmetic instruction. This register is also utilized during input-output operations.

The remainder register, often termed the "lower accumulator" or "R register," is a temporary storage register used for holding the remainder in division or the least significant half of a product. It is also used to extend the range of the dividend in certain divide commands and permits the accumulation of a double-length product resulting from certain multiply commands.

The X register is used during the execution of certain commands such as floating point, input, and transfer of control.

Control

The control console contains the operating switches, indicator lights, visual readout display, and numerical input keyboard. (See Fig. 5.) Operating switches are used to direct computer operations and to obtain visual readout of memory and register contents. Main components of the control elements within the computer are the command register and location counter. The command register, referred to as the "C register," is a 1-word temporary storage register used for holding the command to be executed. The location counter is a storage register which holds the address of the command to be executed. For most commands, the address in the location counter is increased to indicate the next half-word location each time a command is executed. Commands are automatically executed in address sequence, unless interrupted by transfer of control commands.

There are two types of commands which interrupt sequential operation: unconditional and conditional transfer of control commands. An unconditional transfer of control command causes the control unit to receive the next command at a new location specified by the address portion of the transfer of control command. For a conditional transfer of control command, the next command is received from the new specified location only if a certain condition exists. For example, when using a TMI (Transfer on Minus) command, transfer of control occurs only if the contents of the accumulator are minus. If this condition does not exist, the control unit receives the next instruction in normal sequence. Conditional transfer control commands provide the computer with the ability to alter its sequence of operations as a consequence of an intermediate result.

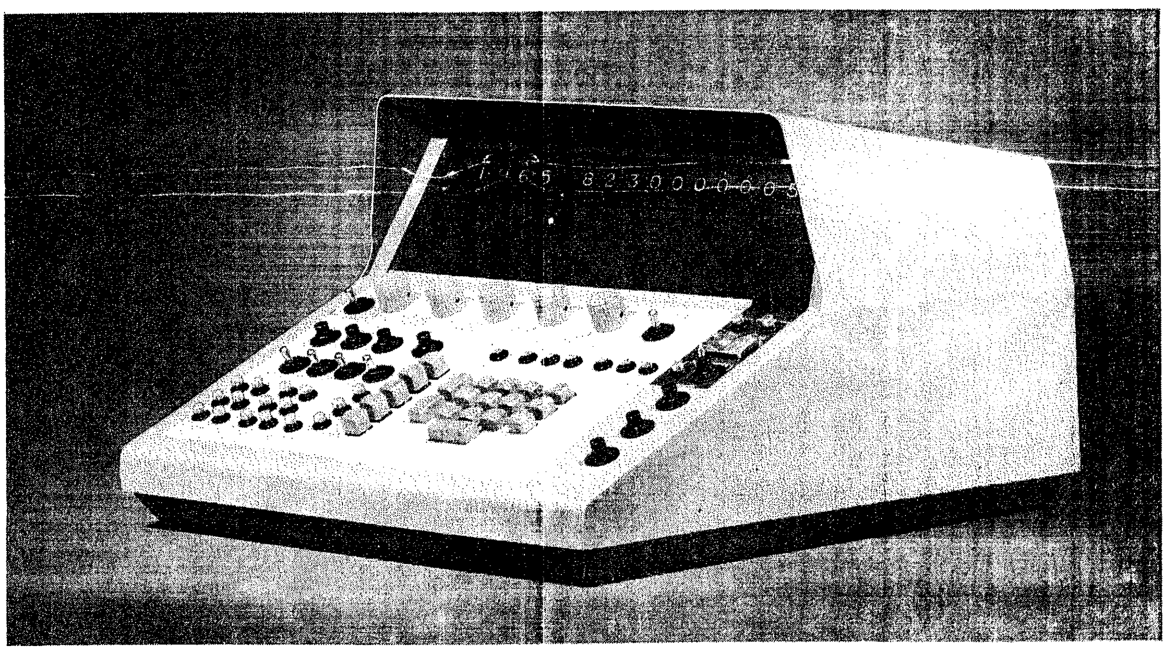


Fig. 5. Control console

Input

The input devices are the equipment used to fill the memory with commands and data, to set the location counter to the address of the initial command, and to provide starting and stopping signals. RECOMP has various input devices including a control console keyboard, a photoelectric paper tape reader, and typewriter. These input devices bring data and commands into the memory through the A register under the influence of the control unit.

Output

A visual display, paper tape punch, and typewriter are the standard output devices. In RECOMP II, processed data are removed through the B register under the influence of the control elements.

FORMAT OF STORAGE INFORMATION

Operation on and transfer of data are accomplished by interpreting binary information in the form of commands representing operations to be performed. Commands are stored in the main memory in a manner similar to the manner in which numerical quantities representing numeric, alphabetic, or combined alphanumeric information are stored.

Physically, so far as the computer is concerned, there is no difference, except in their arrangement (see Fig. 6 and 7), between numbers representing data and numbers representing commands. In RECOMP II, two commands can be placed in one word. A typical command consists of three parts: an algebraic sign, an operation code, and an address. The operation code portion of the command is a number which represents the operation to be performed, i.e., addition, subtraction, multiplication, etc. The address portion of a command specifies the location of the operand. Therefore, the memory contains not only the data pertaining to a particular problem, but also the commands required for producing the problem solution.

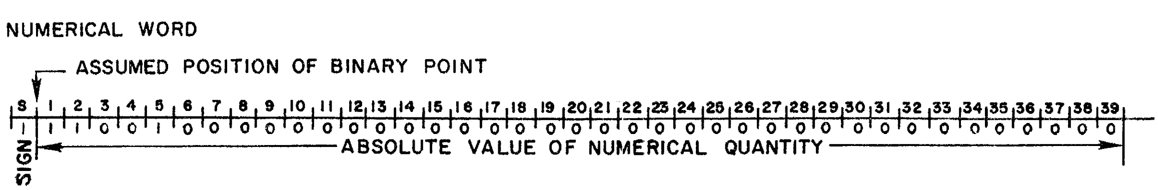


Fig. 6. Word format of typical number in memory

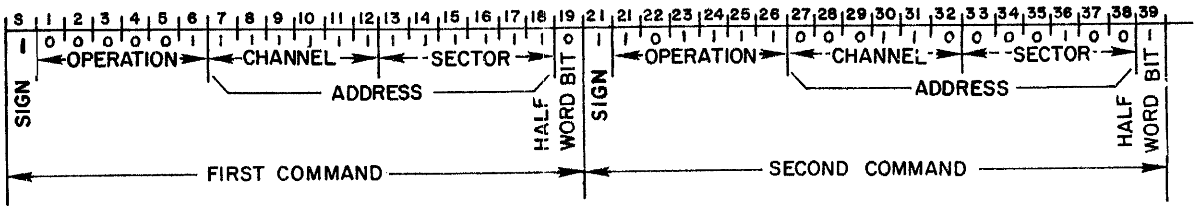


Fig. 7. Format of a typical instruction pair in memory

Numerical Format

A numerical quantity stored in the memory consists of 40 binary digits or "bits," as shown in Fig. 6. The absolute value of the numerical quantity occupies 39 of the 40 bit positions; the most significant digit is in position 1 and the least significant is in position 39. It is convenient for the programer to assume the binary point to precede bit position 1. The numerical value of the number shown is 0.620 octal or 0.78125 decimal. The "S" digit specifies the sign (1 for plus and 0 for minus) of the number. The sign of the number indicated in Fig. 6 is plus.

In RECOMP II, a word (sector) consists of 40 bits. A rotating disk divided into channels, with the channels divided into words, constitutes the memory. Sixty-four of the channels, each containing 64 words, comprise the information storage capacity. The last 16 words of storage in the 64th channel are used for the two high-speed, 8-word loops.

Each numerical quantity occupies a definite location in the memory. To control processing of a number, its location must be specified in the given command. Both the 64 memory channels and the words are numbered octally 00 through 77. A specific location is referred to, first by channel, and then by sector. For example, 0126 refers to channel 01 and sector 26, and 7342 refers to channel 73, sector 42.

Command Format

Commands are stored in the memory in the form of two commands per location. (See Fig. 7.) Each command consists of a sign, an operation code, and an address which tell the computer which operation to perform and the memory address or location of the number that is to be operated on.

The operation code consists of 6 binary digits or 2 octal digits; for example, 01 octal or 000001 binary represents the command "ADD" (Add).

The channel address consists of 6 binary digits or 2 octal digits; for example, 77 octal or 111111 binary represents channel 63.

The sector address also consists of 6 binary digits or 2 octal digits; for example, 77 octal or 111111 binary represents word sector 63.

Bit positions S and 20 specify the sign (1 for plus and 0 for minus) of the first and second commands, respectively. The sign is normally plus.

Bit positions 19 and 39 contain the command indication or "half-word" bit. This bit is significant only in the address of transfer of control, store address, word output, and display commands. It specifies which command (0 for first command and 1 for second command) of the location specified by the address is to be operated on. An example of a command utilizing the command indication bit is the second command shown in Fig. 7. It specifies an octal 57 or TRA (Transfer) operation, which tells the computer to transfer control of the program to the second command (1-digit) in memory location channel 06, sector 04 (0604). With word output and display a 0 bit specifies command format and a 1 bit indicates decimal format.

The planning required to determine the sequence of instructions needed for solving a problem and the storage positions for data is called "programing." A "routine" is the ordered set of instructions and numbers required for the solution of a problem.

NUMBER SYSTEMS

Number systems applicable to the RECOMP II computer are the decimal system, the octal system, and the binary system. Information going into or coming from the computer may be in decimal, octal, alphabetic, alphanumeric, or a combination of octal and binary. However, the computer operates strictly with binary numbers. The systems differ in that each has a different radix or base number. The radix of the decimal system is 10, of the octal system is 8, and of the binary system is 2. The largest possible digit of either system is the radix minus one: i. e., the radix 10 - 1 = 9 for the decimal system, the radix 8 - 1 = 7 for the octal system, and the radix 2 - 1 = 1 for the binary system. The maximum number of digits available in either system always equals the radix. For example, 0, 1, 2, . . . 7, 8, and 9 for the decimal system;

0, 1, 2, ... 5, 6, and 7 for the octal system; and only 0 and 1 for the binary. The arrangement of digits in the three systems is shown in Table 1. (A subscript may be used to distinguish the number system in use. For example 1101_{10} indicates decimal, 1101_8 indicates octal, and 1101_2 indicates binary.)

Table 1. Decimal, octal, and binary numbers

Decimal	Octal	Binary
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	10	1000
9	11	1001
10	12	1010
11	13	1011
12	14	1100
13	15	1101
14	16	1110
15	17	1111
16	20	10000
17	21	10001
etc	etc	etc

Decimal System

The decimal system (as well as the octal and binary systems) uses a positional notation scheme, where the value assigned to any digit depends on the position of that digit relative to the decimal point. For example, the digit 2 has considerably different meanings in 2000 and in 0.002.

In the decimal system, digits 0 through 9 are used. The digits increase by 1 unit up to 9, which is the largest digit. To create the next higher digit (10), a 1 is placed in the second position and a 0 in the first position. The number 10 may also be represented as $1 \times 10^1 + 0 \times 10^0$ where $10^1 = 10$ and $10^0 = 1$. Therefore, in an integer number the second position is considered the ten's position and the first, the unit's position. The number 28 would be represented as $2 \times 10^1 + 8 \times 10^0$. This manner of representing a decimal number continues for numbers after 99. For example, 123 would be represented as $1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0$, where the most significant digit (1) is in the hundred's (10^2) position. Fractional decimal numbers are represented similarly but with decreasing powers of 10. For example, 0.056 may be represented as $0 \times 10^{-1} + 5 \times 10^{-2} + 6 \times 10^{-3} = \frac{0}{10} + \frac{5}{100} + \frac{6}{1000}$. Therefore the positions to the right of the decimal point are tenths, hundredths, thousandths, etc. The decimal $36497.36497 = 3 \times 10^4 + 6 \times 10^3 + 4 \times 10^2 + 9 \times 10^1 + 7 \times 10^0 + 3 \times 10^{-1} + 6 \times 10^{-2} + 4 \times 10^{-3} + 9 \times 10^{-4} + 7 \times 10^{-5}$.

Octal System

The radix of the octal system is 8. Therefore, the largest digit is 7. The octal system follows the same pattern as the decimal system. Because 7 is the largest single digit, the next higher number (8) is formed by placing a 1 in the second position from the right and a 0 in the first position. The octal number 461 may be represented as $4 \times 8^2 + 6 \times 8^1 + 1 \times 8^0$ = decimal 305. Fractional octal numbers follow the same pattern as decimal fractions, the radix now being 8. For example, the octal number $0.357 = 3 \times 8^{-1} + 5 \times 8^{-2} + 7 \times 8^{-3}$ = decimal 0.466796875.

Binary System

The radix for the binary system is 2. (Refer to Appendix B for the Table of Powers of 2.) The largest single digit is 1. This system has the advantage that it uses only two digits, 0 and 1. Because the largest binary digit is 1, the next higher digit (equivalent to decimal 2) is formed by placing a 1 in the second position and a 0 in the first position. The second position then represents the number of 2's present. The binary number $101.1 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1}$ = decimal 5.5.

Conversion Between Systems

Octal or Binary to Decimal

To convert an octal or binary number to the decimal system, each digit, starting from the most significant, is first separately converted to its decimal equivalent. The converted digits are then added to produce the final decimal number. (Refer to Appendixes C and D for the conversion tables.) For example, to convert octal 175.005 to decimal:

$$1 \times 8^2 + 7 \times 8^1 + 5 \times 8^0 + 0 \times 8^{-1} + 0 \times 8^{-2} + 5 \times 8^{-3} =$$

$$64 + 56 + 5 + 0 + 0 + 0.009765625 =$$

$$125.009765625$$

To convert binary 110.0010 to decimal:

$$1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} =$$

$$4 + 2 + 0 + 0 + 0 + 0.125 + 0 =$$

$$6.125$$

Decimal Integers to Octal or Binary

To convert a decimal integer to either an octal or binary number proceed as follows:

1. Divide the number by either 8 for the octal system, or 2 for the binary system.
2. Note the remainder to one side.
3. Divide the new quotient by either 8 for the octal system, or 2 for the binary system.
4. Note the remainder to one side again.
5. Continue steps 1 through 4 until a zero quotient is obtained.
6. Write the remainders in reverse order of occurrence.

Example (octal): change decimal 122 to octal system.

	Remainder
8) 122	
8) 15	2
8) 1	7
8) 0	1

$$122_{10} = 172_8$$

Example (binary): Change decimal 121 to binary system.

	Remainder
2) 121	
2) 60	1
2) 30	0
2) 15	0
2) 7	1
2) 3	1
2) 1	1
2) 0	1

$$121_{10} = 1111001_2$$

Decimal Fraction to Octal or Binary

To convert a decimal fraction to either an octal or binary number, proceed as follows:

1. Multiply the decimal fraction successively either by 8 for the octal system or by 2 for the binary system; then place the decimal point in the product at the same position as in the multiplicand. The first digit to the left of the decimal point is the most significant digit of the octal or binary fraction.

2. Multiply all digits of the product, except the integer part, by either 8 for the octal system or 2 for the binary system. In the succeeding product, the digit to the left of the decimal point is the next most significant octal or binary digit.

3. Repeat step 2 until the desired accuracy is obtained.

Example (Octal): Change decimal fraction 0.121 to octal system.

```
 .121
   x8
0.968
   x8
7.744
   x8
5.952
   x8
7.616
```

0.121 decimal fraction = (0.0757. . .) octal.

Example (binary): Change decimal fraction .121 to binary system.

$$
\begin{array}{r}
.121 \\
\underline{x2} \\
0.242 \\
\underline{x2} \\
0.484 \\
\underline{x2} \\
0.968 \\
\underline{x2} \\
1.936 \\
\underline{x2} \\
1.872
\end{array}
$$

0.121 decimal fraction = (0.00011 . . . .) binary.

Octal to Binary

To convert octal numbers to binary numbers, simply replace each octal digit by a group of three binary digits.

For example: 4735.56_8 = 4 7 3 5 .5 6
100 111 011 101 .101 110
= 100111011101.101110_2

Binary to Octal

To convert a binary number to the octal system, proceed as follows:

1. Place binary digits in groups of three starting from the binary point.
2. Replace each binary group with its octal equivalent.

For example: 1101100111001.000101 = 1 101 100 111 001 . 000 101
= 1 5 4 7 1 . 0 5
= (15471.05_8)

Binary-Coded-Decimal System

Binary-coded-decimal electrical pulses from the computer are used to actuate the typewriter keys and the control panel readout indicators. This code is also used in the conversion of numbers entered from the control console keyboard. In this system, each decimal digit is replaced by a group of four binary digits.

For example: 121_{10} = 1 2 1
= 0001 0010 0001

ARITHMETIC OPERATIONS

To this point, the performance of arithmetic operations has been referred to in generalities. Arithmetic functions are covered in greater detail below.

All arithmetic operations are performed in the computer in units called registers. RECOMP II has five 1-word registers that can contain numerical commands or data.

The A register (accumulator) is the primary arithmetic register; the R register is an extension of the A register; the C register is the command register containing the operation code and address; and the X and B registers are used for intermediate storage.

In RECOMP II all information is stored on the disk memory in binary representation. This representation takes the form of magnetically charged spots on the disk. These spots are read by the computer as they pass beneath a read head, and are directed to an appropriate register.

On the memory disk all information appears as magnetic spots. There is no way the computer can determine if the information is a command word or a data word. For instance, the binary word

+ 000001101000010111010000111010000110000

could be grouped

+ 000 001 101 000 010 111 010 000 111 010 000 110 000

and considered as the octal data word

+ 0 1 5 0 2 7 2 0 7 2 0 6 0

or the command word

+000 001 101 000 010 111 0 1 000 011 101 000 011 000 0

which may be interpreted:

(0 1) (contents of) (0 3) (contents of)

or

+ ADD 5027.0 + SUB 5030.0

The C register, which directs the compute mode (during arithmetic operations), has the specific task of interpreting all words entering it as instructions.

The program is written onto the magnetic disk into sequential locations. A word entered into the C register is interpreted as a command. The computer obeys the command and repeats the cycle using the subsequent program step. The operational cycle for each command is as follows (RECOMP II stores two commands per computer word):

1. Command selection
2. First command interpretation (number selection)
3. First command execution and augment location counter
4. Second command interpretation (number selection)
5. Second command execution and augment location counter

When the program is read in, the counter is set to the address of the first step and, unless ordered otherwise by the operator or the program, increases by one in the half-word bit each time an instruction is executed. RECOMP reads the setting of the location counter and proceeds to that address for the next instruction.

This entire operation is completely automatic and is performed at very rapid speeds.

The only exception to the above operational cycle is when the algebraic sign position is negative; that is, it contains a binary 0. In this case the operation of the first command is ignored and the computer enters the trapping mode (refer to page 51).

Addition, subtraction, multiplication, and division of whole numbers (integers) are routine operations. However, when mixed numbers (integers and fractions) are manipulated, care must be exercised to keep the point between integer and fraction in the proper place. In fixed point machines this task of remembering where the point lies falls on the operator or programer. Floating point computers such as RECOMP II, with special circuitry and instructions, permit this tiresome task to be carried out by the computer.

Rules for Sign in Arithmetic Operations

The rules for the sign of the results of arithmetic operations are the same as those for algebra.

1. Addition. If the signs of the augend and the addend are the same, the sign of the sum is also the same. If the signs are different, the smaller in magnitude is subtracted from the larger, and the sign of the larger is applied to the difference.

2. Subtraction. The sign of the subtrahend is changed and the result is added to the minuend algebraically as described above.

3. Multiplication. If the signs of the multiplier and multiplicand are the same the product is positive; if they are different it is negative.

4. Division. If the sign of the divisor and dividend are the same the quotient is positive; if they are different it is negative.

When a zero value is obtained in the A register as the result of an addition or subtraction, the sign of zero will be the same as the initial sign of the A register.

Fixed Point Arithmetic Operations

In fixed point arithmetic operations, it may be convenient to scale numbers so that the point falls between the sign and the most significant digit of the word. Consequently, in a problem with numbers greater than, equal to, or less than one, the programer must consider the computer to be handling only numbers less than one in absolute value, and all quantities must be scaled accordingly.

In scaling, it is necessary to remember the original magnitude. This is done by the programer or through data input subroutines. The number entered is accompanied by a scale factor - the power of the radix which, when multiplied by the scaled value, gives the original value. In the computer it gives the true point at which the integer is to be separated from the fractional portion. For example, the number +427.8924564 would have a scaled value of +.4278924564 and a scale factor of +3 indicating the power of the radix (10 in this case) which after multiplication replaces the point in its original position.

The scale factor is generally not entered into the arithmetic registers during operations on the fraction which it modifies but is stored elsewhere in order that it may be used when numbers of different scales are to be added or subtracted. The scale factors of these numbers are then compared to determine the number of places which the numbers must be shifted to properly aline integer and fractional portions to allow the numbers to be added or subtracted. When numbers are scaled as described above to give a maximum number of significant digits, care must be excercised before the numbers are utilized in arithmetic operations. The scale factors must be such that the scale of the result will produce an answer which will be within the range of the arithmetic registers.

Overflow occurs when the result of an arithmetic operation becomes too large to be held by the accumulator. Any operation producing a digit to the left of the assumed binary point causes an overflow, and digits to the left of the point (the most significant digits) are lost. When this condition occurs, the computer halts and the OVERFLOW indicator flashes - unless the instruction following the one producing the overflow is a TOV (Transfer on Overflow) command. When the computer is halted, operations are resumed by depressing the RESET, then the START buttons. An overflow can occur in addition, subtraction, or division, but not in multiplication.

The equation shown below with the following values given to the constants and variables is an example of scaling using binary (radix of 2) scale factors.

$$F = 3.254$$
$$G = 2.9394$$
$$K = 0.1120$$
$$X = 10^5$$
$$Y = 3.802651515$$
$$Z = 0.3458106061$$
$$P = 80.3128$$

Find

$$C = \left\{-F + G(Y + Z) - K\left[(Y + Z)^2 - YZ\right]\right\}\frac{P}{X}$$

Z needs no scaling as it is less than 1 and Y could be scaled down by multiplying by 2^{-2} which yields

$$Y \cdot 2^{-2} = (3.802651515)2^{-2} = 0.950662878$$

Looking at the equation we see that we need the sum Y + Z. This means, from the law of exponents, that Z must be scaled the same as Y. Therefore:

$$Z \cdot 2^{-2} = (0.3458106061)2^{-2} = 0.0864526515$$

$$(Y + Z)2^{-2} = 1.0371155295$$

The number exceeds machine capacity. This shows that not only must the size of the number itself be considered, but also its use in the equation. In this case it is necessary to scale Y and Z by 2^{-3} so that the addition can be performed.

$$Y \cdot 2^{-3} = (3.802651515)2^{-3} = 0.475331439$$

$$Z \cdot 2^{-3} = (0.3458106061)2^{-3} = 0.0432263257$$

$$(Y + Z)2^{-3} = 0.5185577647$$

Squaring

$$(Y + Z)^2 . 2^{-6} = 0.2689021553$$

Multiplying by K

$$K \cdot 2^0 \left[(Y + Z)^2 \cdot 2^{-6}\right] = 0.0301170414$$

To add F we must scale it by 2^{-6}

$$F \cdot 2^{-6} = (3.254)2^{-6} = 0.05084375$$

Adding

$$K (Y + Z)^2 \cdot 2^{-6} + F \cdot 2^{-6} = 0.0809607914$$

Scaling the other constants and data

$$G \cdot 2^{-3} = (2.9394)2^{-3} = 0.367425$$

$$X \cdot 2^{-17} = (10^5)2^{-17} = 0.762939453$$

$$P \cdot 2^{-7} = (80.3128)2^{-7} = 0.62744375$$

and proceeding on with the solution of the equation.

$$(YZ) \cdot 2^{-6} = 0.020546832$$

$$(KYZ)2^{-6} = 0.0023012452$$

$$\left\{ KYZ - \left[K (YZ)^2 + F\right] \right\} \cdot 2^{-6} = -0.07865462$$

$$G(Y + Z)2^{-6} = 0.1905310867$$

$$\left\{ G(Y + Z) + KYZ - \left[K(Y + Z)^2 + F\right] \right\} \cdot 2^{-6} = 0.1118715405$$

$$\left\{ G(Y + Z) + KYZ - \left[K(Y + Z)^2 + F\right] \right\} P \cdot 2^{-13} = 0.0701930989$$

To make possible the division we must scale down the number 2^{-17}

$$\left\{G(Y+Z)+KYZ - \left[K(Y+Z)^2+F\right]\right\} P \cdot 2^{-17} = 0.004387068$$

$$\left\{G(Y+Z)+KYZ - \left[K(Y+Z)^2+F\right]\right\} \frac{P \cdot 2^{-17}}{X \cdot 2^{-17}} = 0.005750218 = C$$

The example solved above deals with only one set of data. Normally a problem will have many sets. Such a problem must be scaled to allow for maximum number size of all data. However, care must be taken not to scale down the numbers too much and lose accuracy.

Floating Point Number Operations

The use of floating point numbers greatly eases the task of programing, especially in scientific and mathematical problems. In floating point arithmetic, scaling is necessary only on input. Numbers may vary in magnitude without overflow occurring. Floating point arithmetic instructions are used in the same manner as their fixed point equivalents.

Each floating point number uses two words of memory for storage. (See Fig. 8.) One word contains the mantissa (fraction) and the other word contains the characteristic (exponent). The mantissa appears as a binary fraction less than one, which is the same form as the fixed point. The characteristic of the number appears at the right-hand portion of the following word as a binary integer at a binary scale of 2^{-39}.

	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
M	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Fig. 8. Floating point number

For example, the number +5.0 would appear in floating point form in two consecutive memory locations as follows: $+5.0 = (+.625)2^3$. The general notation for all floating point numbers is $N = M \cdot 2^c$.

TIMING CHARACTERISTICS

The two characteristic factors of computer operation which must be considered in programing are input-output time and computer operation time.

Input-Output Rates

Through the photoelectric paper tape reader, input of information may be accomplished at a rate of 400 characters/second. These characters may be alphabetic, numeric, or alphanumeric. Input via the decimal keyboard or typewriter is accomplished at rates up to 10 characters/second, dependent on operator capabilities. Output rates depend on the output medium selected. The electric typewriter prints 10 alpha or numeric characters/second, and the paper tape punch operates at 20 characters/second, or at the speed of the typewriter when punching and typing.

Computer Operation Time Factors

Two factors enter into the timing of computer operations: access time and command execution time. Access time is required: (1) to locate a command pair in memory and read it into the command register and (2) to locate the address specified in the command and read or write the data at that memory location. Command execution time is required for performing the action specified in the command.

Access Time

The access time for obtaining commands or operands from or putting operands into main memory varies from 0.54 ms to 17.6 ms. Average access time to main memory is 9.0 ms. Information stored in a high speed loop location can be retrieved in an average access time of 0.95 ms (0.54 ms to 1.35 ms) and placed in it in an average time of 1.49 ms (0.54 ms to 2.43 ms).

Access time must be considered each time a new command pair is brought into the command register and when the instruction calls for information to be located and brought into an arithmetic register or placed from an arithmetic register into a specific location in memory. Minimum access times are 0.54 ms for (1) locating the address containing the command pair and reading the command pair into the command register, and (2) locating the address of the operand and reading the operand from or writing it into memory. No locating time is needed for the second (right half) command of a pair because the second command is already in the command register (unless the second command of a pair is the first in a sequence).

Operation Time

The time utilized in performing the action specified in a command also varies, according to the nature of the command. Execution times of all commands are given in Appendix E. It will be noted that while most commands have a fixed execution time, the times of some vary according to the number of shifts, degree of normalization, or other conditions.

Thus, many factors must be considered in timing a program. Estimating program processing time prior to running a program is usually unnecessary. It is desirable though to prepare a program in such a way that the processing time will be as short as possible. This procedure is called optimization and is very important in routines and subroutines that are used repeatedly and when specific operations must be completed within a definite time. Optimization is possible on all operations except those involving the paper tape reader, paper tape punch, and typewriter.

Both optimization and estimation of processing time are much more easily accomplished by working with word times instead of milliseconds. One word time is equivalent to approximately 0.27 ms, the time required for the memory disk to rotate the distance occupied by one word (sector) of memory. Optimization based on word times rather than milliseconds has another advantage--any slight variation in disk rotation speed will not change optimization because the disk is the source of synchronization of all computer operations.

Access times for commands and operands as well as the execution times of commands can be expressed in word times. Because there are relatively few fixed times other than minimums, care is necessary to allow sufficient word times for each operation. Insufficient word times will not affect sequence of operation nor accuracy but will result in an unoptimized program.

Except on the execute command phase, the number of word times utilized by each major phase of a command cycle can vary from 2 to 65 (64 word times equal 1 disk revolution). The major phases, applicable to most commands, are (1) locate and read command pair, (2) locate address of operand and read or write operand from or to memory, and (3) execute command. Minimum access times are 2 word times for a pair of commands and 2 for each operand. To achieve this minimum requires proper location of commands and operands in memory with respect to one another. The minimum command execution time is one word time (see Appendix E). While many execution times are fixed, others are variable according to the number of shifts, whether transfer is made or not, differences in sizes of exponents, or other conditions. These variables can also be readily expressed in word times--either in exact quantity if the variable is known or by a maximum quantity if the variable is unknown. A maximum number of word times will neither destroy synchronization of operations nor materially reduce the degree of optimization. An insufficient number of word times, however, can markedly reduce the degree of optimization.

OPERATING CONTROLS

All major controls for RECOMP are conveniently located on the control console thus allowing the operator to exercise complete control over the system during operation. (See Fig. 5.)

The control console contains a readout display for visual output, a decimat keyboard for input, indicator lights, and operational switches.

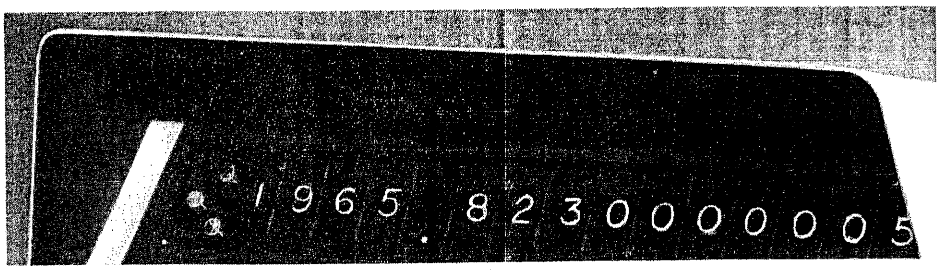


Fig. 9. Display register

READOUT DISPLAYS

The display register (Fig. 9) on the control console provides visual display of information in registers or storage and permits verification of data to be put into storage before entry is made. Rotary switches (Fig. 13) enable the operator to select the specific memory locations and registers from which it is desired to display information. The information is displayed on a bank of 15 Nixie (decade) tubes and a small panel indicating algebraic sign and format of the displayed information. Glow filaments in each Nixie tube permit display of any number from 0 through 9 in each tube. A decimal point, indicated by a small neon tube, may be displayed between any digits and preceding the first and following the last digits. Information may be displayed either manually when computation is stopped or under computer control by executing the Display command.

The format for manual display of information may be command, octal, or decimal (Fig. 10 through 12). Procedures for displaying information manually are given on pages 26 and 27, and by program control on page 28.

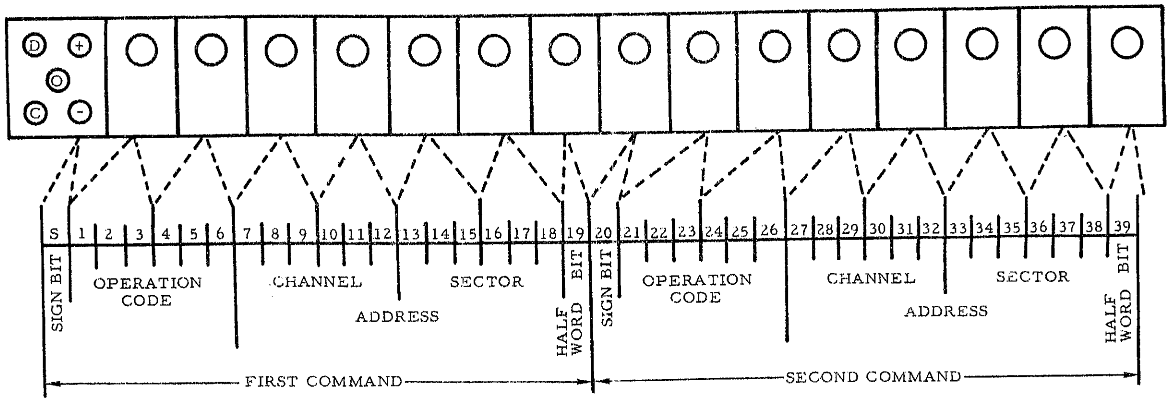


Fig. 10. Command word format

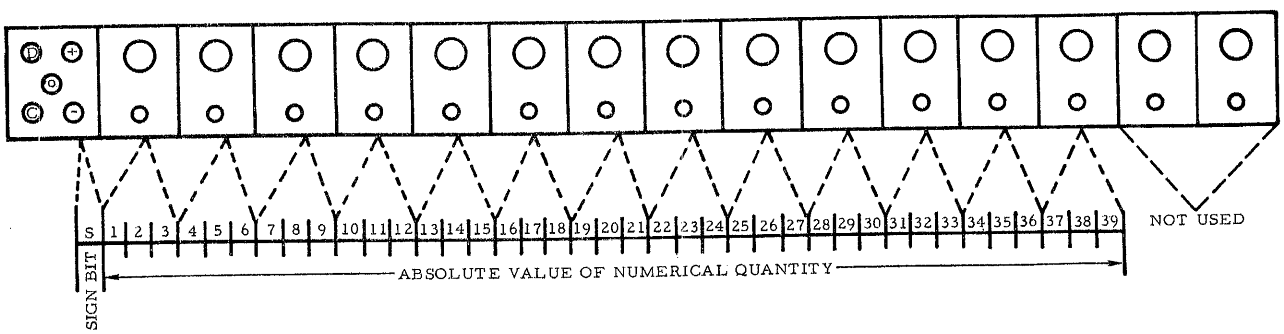


Fig. 11. Octal number format

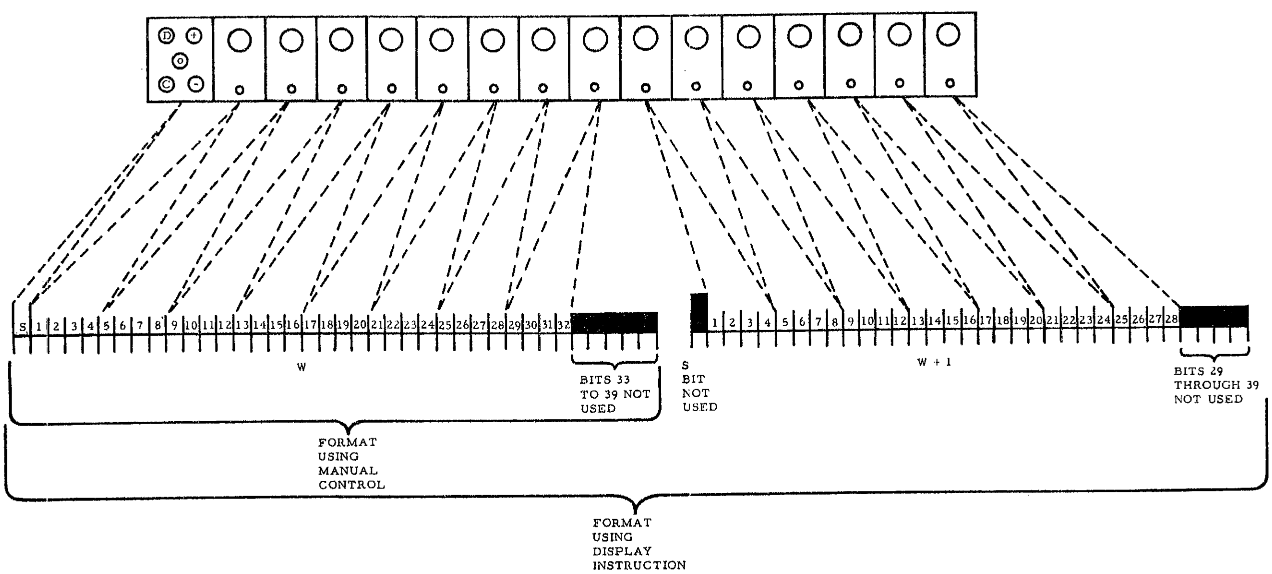


Fig. 12. Decimal format

Manual Input Display

Information, including sign and decimal point, is displayed on the console visual readout immediately upon depression of a key on the control console keyboard. The information remains displayed until either an ENTER or CLEAR key is depressed.

When the number fill mode is used, the digits of the number may be observed for accuracy before they are placed into storage. Should an error be noted before the DECIMAL POINT or ENTER key is depressed, the digits can be deleted without advancing the location counter by depressing the CLEAR key. This also clears the visual display. The correct digits may then be keyed.

Since depressing the DECIMAL POINT key will cause entry of an integral number preceding the decimal point, the number should be checked before depressing that key. Digits remain on display after the DECIMAL POINT key has been depressed until the ENTER key is depressed following entry of any fractional portion.

When the command entry mode is used, all information may be viewed before the instructions are entered by the ENTER key.

For other than commands, the visual readout need not be filled before information may be entered.

Following are the formats of decimal number and command displays when the information is entered manually into storage in the manner described on pages 36 to 41:

Decimal Number — Algebraic sign (plus or minus) is displayed on the panel at the left end of the display. Digits are displayed on the visual readout as the control console keys are depressed. When mixed numbers are entered the integral and fractional parts are displayed side by side. The decimal point is displayed below the level of the digits in the appropriate position.

NOTE: Depressing the DECIMAL POINT key terminates the entry of digits as integers, enters the information into the memory location (M) designated by the location counter, and advances the location counter to the next consecutive location (M + 1). It does not clear the visual readout; this is done by depressing the ENTER key. When the ENTER key is depressed immediately after the DECIMAL POINT key, as in a number containing only integers, the location counter is not advanced. Depressing the ENTER key a second time will advance the location counter. Neither is the location counter advanced when the DECIMAL POINT key is depressed immediately following the sign, as in a number containing only a fraction. If the number contains both an integer and fraction, with the fraction entering location M + 1 when the ENTER key is depressed, the counter is advanced by one to M + 2.

Command — Algebraic sign (plus or minus) of the first command is displayed on the panel at the left end of the display; the sign of the second command as a 0 or 1 (0 = minus, 1 = plus) in the eighth display position.

The digits representing the octal equivalent of the operation code are displayed in the same sequence as the keys are depressed on the control console keyboard. These digits are shown in the first and second display positions for the first command and the ninth and tenth positions for the second command.

The 4 octal digits and 1 binary digit of the address are also displayed in the same sequence as that in which the keys are depressed. These digits are displayed in the third through seventh and eleventh through fifteenth positions for the first and second commands, respectively.

NOTE: The display is cleared when the ENTER key is depressed to enter the instructions into the computer. The visual display must be filled before entry is attempted.

Manual Output Display

Information is displayed immediately upon depressing either of the three readout buttons - command, octal, or decimal. Following are the formats of each of the three types of information displays:

Command — Algebraic sign (plus or minus) of the first command is displayed on the panel at the left end of the display, and of the second command as a 0 or 1 (0 = minus, 1 = plus) in the eighth number display position.

The digits representing the octal equivalent of the operation code are shown in the first and second display positions for the first command and ninth and tenth positions for the second command.

The 4 octal digits and 1 binary digit of the address are displayed in the third through seventh and eleventh through fifteenth positions for the first and second commands, respectively.

Octal — Algebraic sign is displayed as a plus or minus on the panel at the left end of the display.

The sign is followed by up to 13 octal digits representing the octal equivalent of the binary word. The fourteenth and fifteenth display positions are not used and remain blank.

Decimal — Algebraic sign is displayed as a plus or minus on the panel at the left end of the display.

The sign is followed by eight decimal digits in the first through eighth display positions. The ninth through fifteenth display positions are not utilized and remain blank. Display characteristics will be the same for a number containing only integers, only fractions, and both integers and fractions, except that for both integers and fractions the display is obtained from two consecutive word locations.

NOTE: Only the sign and 32 most significant bits from the word are utilized in decimal format, the bits in positions 33 through 39 having no effect on the readout. Data to be displayed in decimal format are first converted by programing into binary coded decimal form of four binary digits for each decimal digit, decimal point, or blank. A blank or decimal point may be substituted for any decimal digit. The sign is taken directly from the sign position in memory. Conversion is made before the readout button is depressed; when a number of words are to be displayed, the words are stored in consecutive memory locations.

Programed Display

Information is displayed under program control by execution of the Display command. (Refer to page 81.) The information will remain on display until a new Display command is executed or until computation is stopped and new information is entered from the console keyboard or readout is obtained manually. The manner in which information is displayed is the same as that described under manual output display on page 27. The information and program locations at which information is to be observed are selected by the programer.

OPERATING CONTROLS AND INDICATORS

The operational controls (Fig. 13) provided on the control console are:

1. POWER switch - Depressing the POWER-ON switch applies power to the computer. The ON indicator immediately lights, and the disk begins to rotate. After a short warmup period, the disk is moved by an automatic solenoid into the proximity of the headplate, and the READY indicator light indicates the computer is ready for use.

Depressing the POWER-OFF switch causes the solenoid to release the disk from the proximity of the headplate. The READY indicator is turned off. After a 1-sec delay, the ON indicator is turned off indicating that all power to the computer is removed.

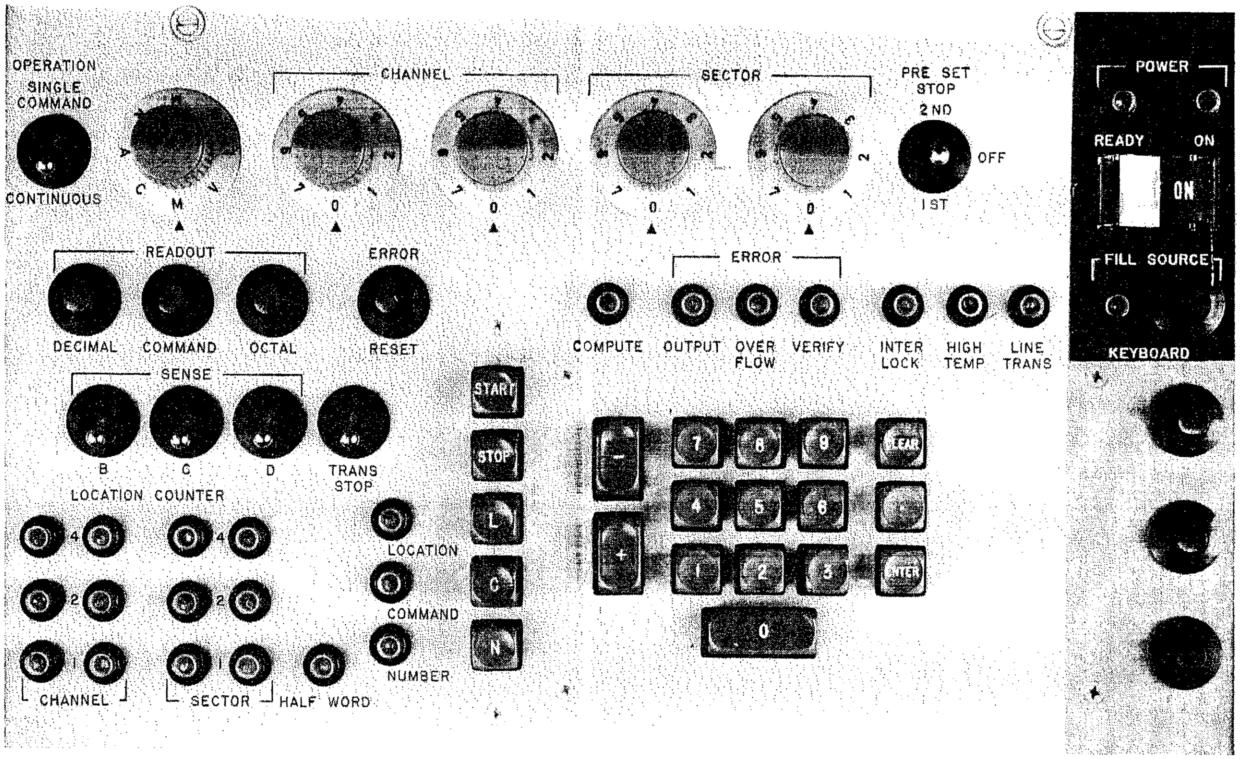


Fig. 13. RECOMP operational controls

2. FILL SOURCE button and indicator - The indicator located under the POWER switch is turned on by depressing the button located to the right of the indicator. It is turned off by operating the FILL switch on the typewriter or paper tape reader, or by executing the input commands RDY or RDZ. When the indicator is on, information can be entered from the control console keyboard; when the indicator is off, the keyboard becomes inactive and information can be entered from the keyboard of the typewriter or from the paper tape reader.

3. REGISTER, CHANNEL, and SECTOR selectors - An 8-position rotary switch permits the operator to select either the C, A, R, or X register or a full word from main memory (M) or the L or V loop for presentation on the display register. When the register selector is set to the M position, the octal address of the desired memory location must be set on the CHANNEL and SECTOR selectors; when set to the L or V position, only the SECTOR selectors need be set.

4. READOUT buttons - The three buttons (DECIMAL, COMMAND, and OCTAL) operate in conjunction with the REGISTER, CHANNEL, and SECTOR selectors to cause the display register to indicate the contents of the word or register selected in the format indicated.

5. PRE SET STOP switch - When this switch is placed at 1ST or 2ND, the computer stops after executing the command located in the left or right half word of the full word whose address has been set on the CHANNEL and SECTOR selectors. The location counter may display one of two things: the HALF WORD indicator will be advanced by a "1," or, if the order just executed was a transfer command, the new location will appear. Thus, the PRE SET STOP switch provides a means for stopping the computer at a preselected half-word location and is intended for use in the manual checkout of programs. When this switch is placed in the off (center) position, the computer will operate normally.

6. OPERATION switch - This switch has two positions: SINGLE COMMAND and CONTINUOUS. When it is at CONTINUOUS, the computer executes instructions sequentially; when at SINGLE COMMAND, only one instruction is executed in sequence each time the START key is depressed.

7. COMPUTE indicator - While computations are being performed, the COMPUTE indicator is lighted. When the computer stops, or during the execution of input instructions, this indicator is off.

8. ERROR RESET button - Pressing this button causes the three ERROR indicators (OUTPUT, OVER FLOW, and VERIFY) to be reset.

9. ERROR OUTPUT light - This indicator is normally off. Whenever a difference occurs between the information in the computer and what was punched or printed, the ERROR OUTPUT light flashes to indicate a discrepancy detected by the echo-checking circuits. The computer will halt at this point. The nature of the discrepancy may be determined by comparing the characters punched or typed with the contents of the output display register located on the punch unit. This indicator may be reset by pressing the ERROR RESET button.

10. ERROR OVER FLOW light - This indicator is normally off. When the result of an arithmetic operation exceeds the capacity of a register, or when certain restrictions inherent to acceptable instructions are violated, an overflow trigger is turned on. Once this trigger is turned on, only the instruction TOV (Transfer On Overflow) may be executed without causing the computer to stop. If any other instruction is attempted, the computer stops and causes the ERROR OVER FLOW light to flash, indicating an error. If no TOV is given, the computer will halt. A restart would require depressing the ERROR RESET switch and taking corrective program modification. This indicator may be reset either by pressing the ERROR RESET button or by a programed TOV following the instruction which caused the overflow.

11. ERROR VERIFY light - This indicator is used with the tape reader and is normally off. When a discrepancy between what is stored in memory and what is punched on paper tape is detected while operating in the verify mode, this indicator flashes and the tape reader stops. This indicator may be reset by pressing the ERROR RESET button.

12. INTER LOCK light - This neon indicator is normally off, but flashes when the hinged doors of the computer unit are open.

13. HIGH TEMP light - This indicator is normally off, but flashes when the temperature within the computer cabinet reaches 125 F. If this occurs, the computer automatically retracts the magnetic disk from the proximity of the recording heads, turns off all power to the disk, and turns off the READY light. It is necessary to restart the computer after correcting the cause of such high temperature.

14. LINE TRANS light - This indicator is normally off, but will flash when a sharp powerline transient of sufficient intensity to cause an error in computed results is detected.

15. SENSE switches B, C, and D - These three switches control conditional transfer instructions (TSB, TSC, and TSD) and permit manual control of the program, thus directing execution along various branches of the stored program.

16. LOCATION COUNTER lights - The LOCATION COUNTER (Fig. 14) contains 13 neon indicators which display the address of the location of the next instruction to be executed when the computer is operating in the compute mode. Also, it displays the address of the location into which data are being stored in the fill mode or selected for comparison in the verify mode. The indicator lights are grouped to facilitate conversion from the 13-bit binary form representing a location to the 5-digit (4 octal and 1 binary) form which is more convenient for the operator. Examination of the lights reveals that each of the first 4 columns contains 3 lights which are numbered 4, 2, and 1, reading from top to bottom. When reading an address from the lights, the numbers appearing on the lights which are on in each column should be added to obtain a 5-digit address. For example, the address 6014.1 would appear as shown in Fig. 14. It will be noted that the first 6 bits (2 left columns) of the address are used to specify 1 of the 64 available channels of the disk. The next 6 bits (next 2 columns) are used to specify the sector or the particular word location within the specified channel. The remaining bit is used to specify which half-word location (left or right) is to be

used. A decimal point is used only in the written notation for the programer's convenience.

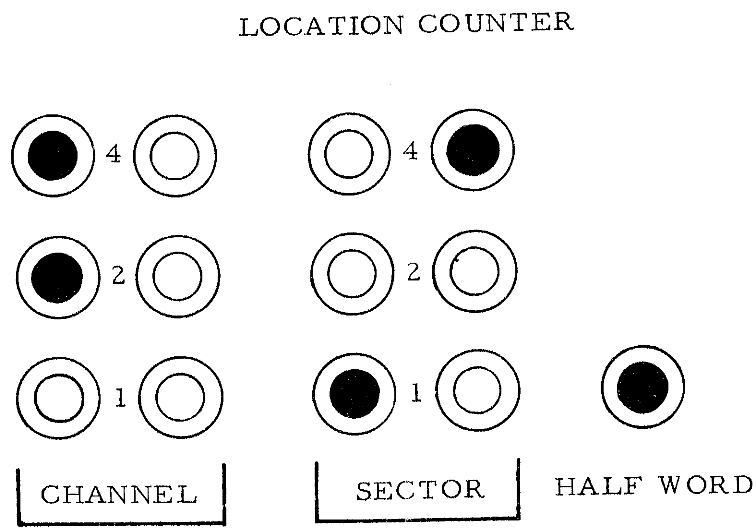


Fig. 14. Five-digit location counter display

17. Transfer START buttons - The three transfer START buttons (START 1, START 2, and START 3) cause the LOCATION COUNTER to be set to the addresses 0001.0, 0002.0, and 0003.0, respectively, and to start computation at that location.

18. TRANS STOP switch - When this switch is placed in the up position, the computer stops after executing any of the transfer instructions, even though conditions necessary for a transfer of control may not have been met. The location of the half word which contained the transfer instruction is placed in the X register (at B = 39). When placed in the down position, the computer executes instructions normally. This switch is of particular value in manual checkout of programs.

19. START key - If the keyboard FILL SOURCE light is on, pressing this key turns on the COMPUTE light, causes the computer to take its instruction from the location specified in the LOCATION COUNTER, and starts computation.

20. STOP key - Pressing this key causes the computer to stop, after executing the current instruction, and turns off the COMPUTE light.

21. L (Location) key - If the COMPUTE indicator is off, depressing this key turns on the adjacent indicator and sets the computer to the

location fill mode. The contents of the LOCATION COUNTER may then be changed by entering a new 5-digit (4 octal and 1 binary) address on the keyboard and depressing the ENTER key on the keyboard.

22. C (Command) key - If the COMPUTE indicator is off, depressing this key turns on the adjacent indicator and sets the computer to the command fill mode. When operating in this mode, each instruction word is entered by depressing the "+" or "-" key and depressing exactly 7 number keys for each instruction. After an instruction pair is keyed, depressing the ENTER key puts the pair of commands into memory. The instruction word thus entered is stored in the location which is displayed on the LOCATION COUNTER. The LOCATION COUNTER will then be increased to the next full word location. Subsequent instructions must be entered in the command format. (See Fig. 10.)

23. N (Number) key - If the COMPUTE indicator is off, depressing this key turns on the adjacent indicator and sets the computer to the number fill mode. When operating in this mode, each number entered on the keyboard is converted automatically from decimal to binary format and stored in the location displayed on the LOCATION COUNTER. After entering a word, the LOCATION COUNTER will be increased to the next full word location. Numbers are stored as binary integers at a binary scale (B = 39) when the "." key is depressed to enter the numbers. Since it has been assumed that the binary point is located between the sign and the most significant bit (i.e., bit No. 1) of a word, the notation B = y indicates the power of 2 scale assigned a number. For example, B = 39, i.e., $N = (n)\ 2^{+39}$ where the quantity n indicates the binary configuration of the number in the computer. Numbers are stored as binary fractions at a binary scale (B = 0) when the ENTER key is depressed after the number. Mixed numbers (numbers possessing both integer and fractional parts) are stored in successive words in memory, i.e., integer (B = 39) in the first memory location and the fraction (B = 0) in the next location. The maximum number of decimal digits which may be entered is the decimal equivalent of 13 octal digits, i.e., integer parts of numbers which are less than or equal to the number 549,755,813,887. Therefore, a safe rule is to limit integer sizes to 11 decimal digits. Fractional parts must also be limited to not more than 11 decimal places. Explicitly, the machine stores the quantity:

$$\left\{F \cdot 10^N\right\} \bmod 2^{39} \qquad \text{at } B = 39, \text{ for integers}$$

and the quantity:

$$\frac{\left\{F \cdot 10^N\right\} \bmod 2^{39}}{\left\{10^N\right\} \bmod 2^{39}} \qquad \text{at } B = 0, \text{ for fractions}$$

An attempt to enter numbers which violate these restrictions will cause the computer to store incorrect results for both integer and fractional parts of numbers and may cause the computer to stop. If the division required for fractional parts cannot be performed, the ERROR OVER FLOW light will flash.

24. Keyboard - Near the lower right-hand section of the control panel is a number keyboard which is used for manual input of information. (See Fig. 13.) It contains keys for the digits 0 through 9, 1 key each for "+" and "-", and 3 control keys (CLEAR, ".", and ENTER).

a. CLEAR key - This key is used to erase faulty information and keyboard input errors before they enter the computer. Depressing the CLEAR key will clear information from the visual display.

b. "." key - This key causes numbers to be stored in memory as binary integers (B = 39).

c. ENTER key - This key releases information for entry into memory in a format determined by the L, C, or N mode keys.

OPERATING PROCEDURES

POWER ON-OFF PROCEDURES

To prepare the computer for operation, the following steps are performed:

1. Depress the POWER ON key. The POWER ON indicator turns on immediately.

2. After a short warmup period, the READY indicator on the console turns on, and the computer is ready for operation. If this indicator fails to turn on, the OFF key should be depressed and a technician called to inspect the difficulty.

3. Turn on input-output equipment.

To turn off the computer, the following steps are performed:

1. Turn off input-output equipment.

2. Depress the POWER OFF key. The READY indicator turns off immediately, and the disk is retracted from the proximity of the head-plate.

3. When the POWER ON indicator turns off, all power to the computer has been removed.

OPERATIONAL STATES

Compute State of Operation

When the computer is operating in the compute state, the COMPUTE indicator on the control console is on. When executing input instructions, the computer leaves this state temporarily and the COMPUTE indicator is turned off until completion of the input operation. The COMPUTE state may be resumed by the appropriate code on the tape or by manual depression of the START button. When executing input instructions with addresses less than 7760.0, the computer enters the input state and remains in that state until a control code is supplied. When executing the input instructions with addresses equal to or greater than 7760.0, the computer enters the input state only until the required number of alpha-numeric characters has been entered into the A register, according to the codes described on page 48. The COMPUTE indicator is then turned on and the computer resumes operation in the compute state.

While in the compute state, the command execution mode is controlled by the position of the OPERATION switch. If the switch is set at SINGLE COMMAND, each command will be executed only after the START key is depressed. Another key depression is necessary to execute an additional command. At CONTINUOUS, the command execution control is under jurisdiction of the computer and the commands are executed at continuous rate.

Input State of Operation

The input state provides for two operational modes (FILL and VERIFY) whereby information is entered into the computer.

In the FILL mode, information enters memory by way of 1 of 3 permissible media: the photoelectric tape reader, the electric typewriter, or the control panel. In the VERIFY mode, information already on the magnetic disk is checked, bit for bit, with corresponding information being read into the computer from tape. This mode provides an operational check on the tape reading and memory recording processes.

When entering information by the control console or by tape, the operating procedure varies according to the unit used and the type of input data. Information from the control console may consist of commands or numeric data. Photoelectric tape reader or typewriter entries may be numeric data, alphanumeric data, or commands.

OPERATIONAL MODES

The following modes are utilized during certain computer operations.

Control Console Fill Mode

Location Selection

For control console entry, the operator first presses the keyboard FILL switch, then sets the LOCATION COUNTER to the desired location (the location where the first word of the input is to be stored). This is accomplished by pressing the following keys in order (see Fig. 13):

1. "L" (location).
2. Decimal keyboard - Depress keys for 5-digit address (4 octal and 1 binary). Address of origin location will appear in readout display as each key is depressed.
3. ENTER - When this key is depressed, readout display is cleared and the address now appears in the LOCATION COUNTER.

For example, if information is to be entered into memory beginning at location 3014.0, that location is first placed into the location counter.

> NOTE: When referring to locations, the decimal point (". ") separating the 4 octal symbols from the 1 binary symbol is a convention used in written notation only. As a consequence, it must not be interpreted as a symbol in itself to be keyed manually or punched into tape when placing a number into the location counter.

Thus, to manually place the above location into the 13-bit location counter, the operator depresses the L key and then the digits 3 0 1 4 0 on the keyboard. The same number also appears on the readout display, having not yet entered the location counter. If the location appears correct, pressing the ENTER key will place the number in the location counter, concurrently clearing the display. This location will then appear on the indicator as shown in Fig. 15, and the computer will be ready to fill locations from 3014.0 consecutively with any type of information acceptable.

When the ENTER key is depressed, the information enters the location counter, character by character, from the right-hand side. As each new character is entered, the previous information in the location counter is shifted left the equivalent of one full octal character. Thus, depressing the ENTER key before a sufficient number of numeric keys is depressed will result in the location counter having new information incorrectly positioned, as well as the left-hand side containing some

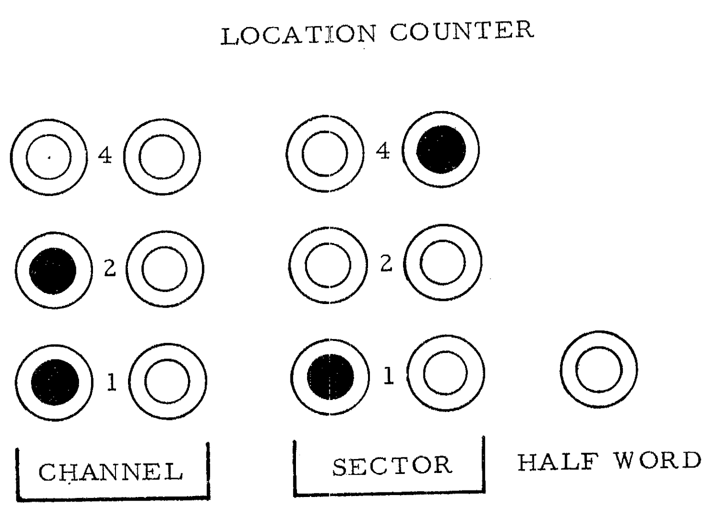


Fig. 15. Location counter display

information remaining from the previous setting. If a nonacceptable character (nonoctal character, 8 or 9) is depressed, the nonacceptable character will appear on the readout display and an erroneous location will be set up if the ENTER key is depressed.

Command Input

In command entry, for example, if the operator wants to enter an instruction pair into 3014.0 calling for the addition of the contents of location 6014.0 and the subtraction of the contents of location 7601.0, the C and + key (sign), the code for addition (01), and the address (6014.0) are depressed. Then the + key (sign), the code for subtraction (03), and the address (7601.0) are pressed. The readout display appears as shown in Fig. 16.

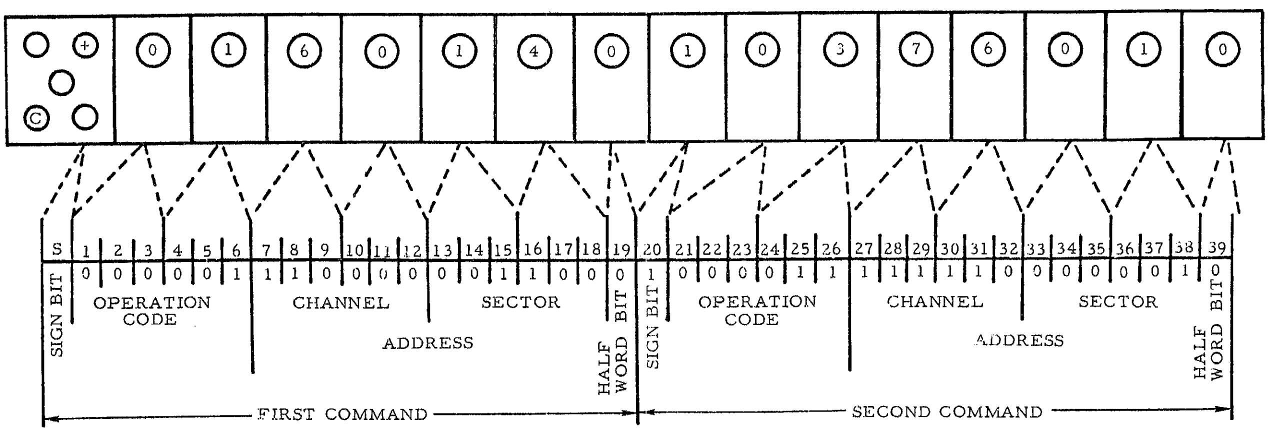


Fig. 16. Readout display

If the sign key "+" or "-" is omitted during command entry, then the first octal character which ordinarily follows will be interpreted as the sign and will be entered as + or - depending on whether the octal character is odd or even. If the fifth acceptable character (half-word bit) in the address portion of an instruction is not punched as a binary 1 or 0, but is punched as an octal character, then the fifth character will be entered into the computer as a 1 or 0, depending upon whether the octal character is odd or even. Hence, omitting a sign or half-word bit will result in the sign and half-word bit position being filled with the low-order bit of a subsequent octal character; the remaining 2 bits of the octal character are lost. Depressing the ENTER key enters the information into the computer.

If input consists of instructions (in command pairs), entry occurs when the following keys have been pressed in order:

1. C (command key)
2. + or - (sign bit)
3. Keyboard (2-digit operation code and 5-digit address)
4. + or - (sign bit)
5. Keyboard (2-digit operation code and 5-digit address for second command)
6. ENTER

As each keyboard depression occurs (steps 2 through 5), the word appears in command format on the readout display. When the ENTER key is depressed, the command pair is stored in memory in the location indicated by the LOCATION COUNTER and the readout display is cleared. The LOCATION COUNTER is then advanced by 1 to indicate that subsequent information entry will go into the next location.

For entry of additional instructions, steps 2 through 6 above are repeated. If commands are being entered in sequential locations, the LOCATION COUNTER is automatically set to the correct address. If the operator is modifying a program and nonsequential addresses are to be used, it is necessary to reset the LOCATION COUNTER for each modification. Steps 1 through 3, page 36, must be repeated.

Numeric Input

When the operator is using the control console, the series of operations required will vary with the type of numeric data to be processed (i. e., mixed number, integer, or fraction).

Mixed Number Input. To enter a mixed number (a whole number and fraction) the following keys are depressed in order.

1. N (number)
2. + or - (sign)
3. Keyboard - Depress keys for the integral or whole part of the number. Digits appear in readout display.
4. "." (decimal point) - The decimal integer is automatically converted to binary at a scale of B = 39 and stored in the address indicated in the LOCATION COUNTER. The LOCATION COUNTER is advanced by 1 word.
5. Keyboard - Enter fractional portion in decimal equivalents. Digits appear on readout display in decimal fraction form.

6. ENTER - The decimal fraction is automatically converted to binary at a scale of B = 0 and stored; the readout is cleared and the LOCATION COUNTER is advanced by 1 word.

For example, to enter the mixed number 12. 25 the operator presses the following keys in order:

1. N (number)
2. + (sign)
3. 1 and 2 (keys)
4. "." (decimal point)
5. 2 and 5
6. ENTER

Integral Number Input. To enter integers the following keys are depressed in order:

1. N (number)
2. + or - (sign)
3. Keyboard (decimal digits)
4. "." (decimal point)
5. ENTER

For example, to enter the integer (whole number) 12, the operator presses N, the + key, the 1 and 2 keys, a ".", and ENTER. The readout display contains a visual indication of input and the location counter advances by 1 word.

For additional integral input, repeat steps 2 through 5 above.

Fractional Numbers. To enter a fraction, the fraction is converted to a decimal equivalent and the following keys are depressed:

1. N (number)
2. + or - (sign)
3. "." (decimal point)
4. Keyboard (decimal equivalent)
5. ENTER

For example, to enter the fraction +0.75, the operator presses N, the + key, the "." key, the 7 and 5 keys, and ENTER.

6. LOCATION COUNTER is advanced by 1 word.

NOTE

In all control console input operations, if an error in pressing keys is detected by the readout display before the ENTER or "." keys have been depressed, the error can be corrected. Remedial action is taken by pressing the CLEAR key, which also clears the readout display.

Typewriter Fill Mode; Paper Tape Preparation

The procedures for entering information directly from the typewriter and preparing punched tape with the typewriter and paper tape punch vary according to the type of input data. When used for tape preparation, the typewriter operates only in conjunction with the paper tape punch, bypassing the computer. A switch on the paper tape punch control panel governs whether the typewriter will operate as a direct input unit or a tape preparation device. Tape preparation can take place (1) when the computer is operating in the COMPUTE state, (2) when the tape reader or console keyboard is the unit used for input, or (3) when the console visual display is the unit used for output. To enter data directly into computer memory from the typewriter, the switch on the punch control panel is set to COMPUTER; to punch data into tape, this switch is set to MANUAL. The tape prepared can then be entered later into the computer memory with the photoelectric paper tape reader. (See pages 45 to 50.) The same procedures applicable to direct input from the typewriter apply also to tape preparation.

For direct input from the typewriter (Fig. 17), the FILL key on the typewriter is depressed and the LOCATION COUNTER is initially set by depressing the following keys in order:

Location Input

1. LTRS (letter shift)
2. L (location counter)
3. FIGS (figure shift)
4. Numeric keys (5-digit address)
5. RETURN (carriage return)

Depressing the RETURN key sets the LOCATION COUNTER to the desired full- or half-word location when the input switch on the tape punch (Fig. 18) is set to COMPUTER. The above procedure is necessary only before entering the first command, numeric, alphabetic, or alphanumeric information except when the succeeding data are entered in a nonsequential location; then the entire procedure is repeated.

Fig. 17. RECOMP typewriter

Command Input

To place instructions directly into the system via the typewriter, the FILL key on the typewriter is depressed, the LOCATION COUNTER is set, and the operator then presses the following keys in sequence:

1. LTRS (letter shift)
2. C (command)
3. FIGS (figure shift)
4. + or - (sign)
5. Keyboard (2-digit operation code and 5-digit address)
6. + or - (sign)
7. Keyboard (2-digit operation code and 5-digit address)
8. RETURN (carriage return)

For input of additional instructions in sequential locations, repeat steps 4 through 8. Instructions not in sequential locations require a new LOCATION COUNTER setting for each pair.

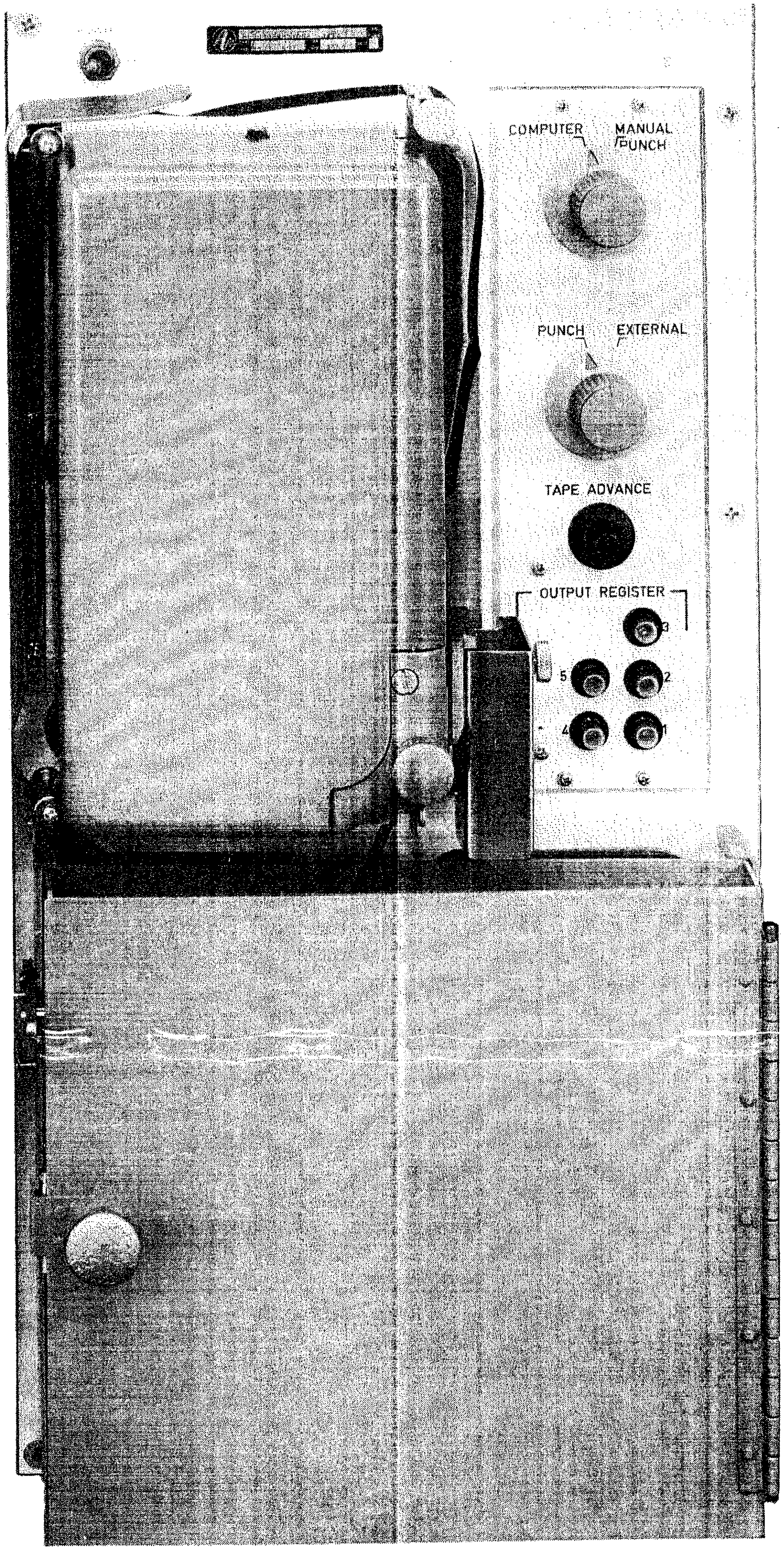


Fig. 18. RECOMP tape punch

Numeric Input

The method of feeding data via direct typewriter varies with the type of input material. There are three approaches: one each for mixed numbers, integers, and fractions.

Mixed Number Input. To insert mixed numbers, depress the FILL key on the typewriter, then the following typewriter keys in order:

1. LTRS (letter shift)
2. N (number)
3. FIGS (figure shift)
4. + or - (sign)
5. Keyboard (decimal digits for integral position of number)
6. "." (decimal point)
7. Keyboard (fractional portion of number in decimal equivalent)
8. RETURN (carriage return)

For additional data, repeat steps 4 through 8.

When the "." key is depressed, the integer is converted to binary (step 5) at a scale of B = 39 and stored in the location indicated in the LOCATION COUNTER. The LOCATION COUNTER is then advanced by 1 and this location is filled with the fractional portion of the number at a scale of B = 0. When the RETURN key is depressed, the LOCATION COUNTER is again advanced by 1.

Integer Input. To enter integers, depress the FILL key on the typewriter, then the following typewriter keys in order:

1. LTRS (letter shift)
2. N (number)
3. FIGS (figure shift)
4. + or - (sign)
5. Keyboard (decimal digits)
6. "." (decimal point)
7. RETURN (carriage return)

For additional numbers, repeat steps 4 through 7.

Decimal Fraction Input. To insert a fraction, depress the FILL key on the typewriter, then the following typewriter keys in order:

1. LTRS (letter shift)
2. N (number)

3. FIGS (figure shift)
4. + or - (sign)
5. "." (decimal point)
6. Keyboard (decimal equivalent of fraction)
7. RETURN (carriage return)

For additional sequential input, repeat steps 4 through 7.

Alphabetic or Alphanumeric Input

RECOMP II is not limited to working with numeric quantities. Alphabetic data can be processed by preceding the alphabetic input with the symbol F to prevent the number or command type of input conversion on the characters which follow. Thus the alphabetic character retains its unique identification in storage. This simplifies output, which is accomplished by programed selection of characters to be typed and/or punched and typing and/or punching them with the single output character commands.

To insert data containing alphanumeric information directly from the typewriter, depress the FILL key on the typewriter, then the following typewriter keys in order:

1. F
2. Keyboard - enter 8 alphanumeric characters
3. RETURN (carriage return)
4. Repeat steps 2 and 3 until all words of alphanumeric data are entered. When all alphanumeric information has been entered and the format is to be changed, the ninth character should be a C, followed by a CR to enter the last set of alphanumeric information. The CR should be followed by an L, C, or N if further information is to be entered, or by an S code if computation is to start. The same format applies to direct tape input. A carriage return as the ninth character enters the information and advances the location counter by 1.

Paper Tape Reader Fill Mode

Photoelectric Tape Reader

The photoelectric tape reader for the punched paper tape input unit is a product of Autonetics research. (See Fig. 19.) To provide an integrated reading unit and a tape magazine more compatible with the speed, size, and reliability of RECOMP II than could be found commer-

Fig. 19. RECOMP photo-electric tape reader

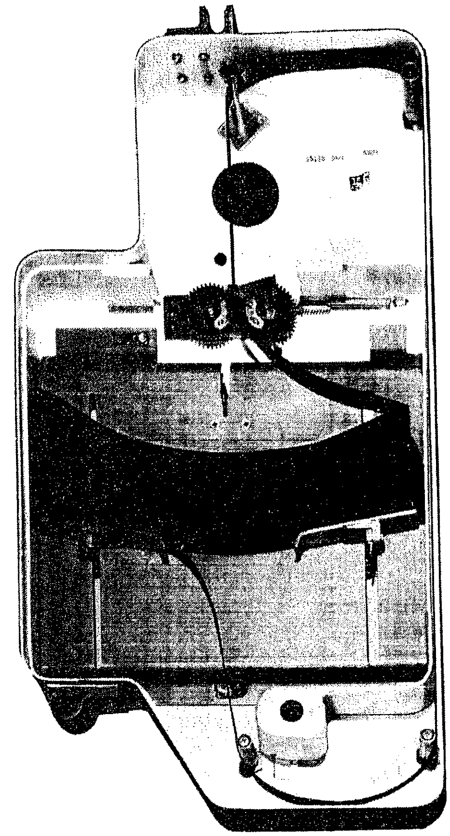


Fig. 20. Photoelectric tape reader cannister

cially, Autonetics designed this compact, high-speed, 400-character/second reader.

The combined paper tape input and output units and their racks are housed side by side in the desk console and weigh approximately 60 lb.

Because spools of paper tape have a comparatively large inertial factor, starting and stopping at high speeds are quite difficult. To minimize this factor, the spools have been replaced by black, 11/16 in. wide, folded paper tape which is contained in easy-loading cannisters (Fig. 20). Because of its high speed, the tape reader is unable to stop in a one-character distance. However, the simple precaution of leaving a blank leader between stops and starts assures correct reading of data.

Tape Configuration

The coding format used in the punched paper tape is standard 5-channel, Baudot teletype code. (See Fig. 21.) Numerical information, decimal, or instruction format for input is converted automatically by RECOMP to the proper binary form for internal use. For alphanumeric entries, a special code signifies that a word to follow is to enter the computer in a format conforming to the configuration in the paper tape.

Tape Modes

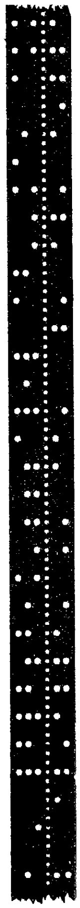
Information may be put on paper tape in any of the modes listed below. In all modes, the information recorded on tape must consist of a minimum of seven characters between Carriage Return and/or Decimal Point codes which enter the information in memory, and between Carriage Return and/or Decimal Point codes and any of the control codes, such as Halt or Start. This procedure ensures proper recording of the information in the computer memory. A blank may be substituted for any of the seven characters. When entering a mixed number or a number consisting of only a fraction, a minimum of five tape sprocket holes must be punched between the Carriage Return code and the next number. The additional sprocket holes ensure sufficient time to convert the fractions to their internal representation. A special key is provided on the typewriter for punching these sprocket holes. One sprocket hole is punched each time the key is depressed. Except for the above special procedures, tape preparation in the various modes is as outlined on pages 41 through 45.

<u>Alphanumeric.</u> When characters present some unique symbolic configuration such as a name or serial number, rather than expressing the concept of quantity, the alphanumeric mode may be used. When this mode is used, each lateral section representing one character is entered into the computer unaltered; i.e., with the internal binary configuration corresponding to the holes punched on the tape. For each character, five binary positions are used in memory. No conversion to a binary quantity is involved.

<u>Decimal.</u> In decimal mode, each decimal number is converted automatically to its equivalent in the binary number system (both fractions and integers).

<u>Instruction.</u> In instruction format, each character from tape (1 lateral section) represents one character of an instruction.

TELETYPE (BAUDOT) CODE
For Numeric, Alphabetic, and Alphanumeric Input-Output
with Typewriter, Tape Reader, and Tape Punch

Teletype Code Tape	Figures[1] Shift Characters[2]	Letters Shift Characters[2]	Octal Representation	Internal[3] Binary Representation	Decimal Representation	CONSOLE CONTROLS[4] Figure[1] or Operation	BINARY CODED DECIMAL[4] Figure[1]	BINARY CODED DECIMAL[4] Code
	ϕ	P	26	10110	22	ϕ	ϕ	0000
	1	Q	27	10111	23	1	1	0001
	2	W	23	10011	19	2	2	0010
	3	E	01	00001	01	3	3	0011
	4	R	12	01010	10	4	4	0100
	5	T	20	10000	16	5	5	0101
	6	Y	25	10101	21	6	6	0110
	7	U	07	00111	07	7	7	0111
	8	I	06	00110	06	8	8	1000
	9	O	30	11000	24	9	9	1001
	+	Z	21	10001	17	+		1100[6]
	-	A	03	00011	03	-		or
	.	M	34	11100	28	.		1101
	Tab/CR[5]	Car. Ret	10	01000	08	Enter	Car. Ret	1011
	/	X	35	11101	29	Clear		
	)	L	22	10010	18	Location		
	:	C	16	01110	14	Command		
	,	N	14	01100	12	Number		
	!	F	15	01101	13	Start		
	s	S	05	00101	05			
	H	H	24	10100	20			
	$	D	11	01001	09			
	&	G	32	11010	26			
	(	K	17	01111	15			
	Fig. Shift	Fig. Shift	33	11011	27			
	;	V	36	11110	30			
	?	B	31	11001	25			
	Ltr Shift	Ltr Shift	37	11111	31			
	Line Feed	Line Feed	02	00010	02			
	Space	Space	04	00100	04		Space	1010
	Blank[7]	Blank[7]	00	00000	00		Terminate	1111[6]
	'	J	13	01011	11			or
								1110

1. The numerals 0 through 9 are represented internally in true binary form and value. Manual entry from typewriter and console keyboard is accomplished by depressing the key for the character desired (no code is necessary).
2. The following nonalphanumeric characters read from tape or received from the typewriter initiate these respective computer actions: Carriage Return, Enter; X, Clear; L, Locations; C, Commands; N, Numbers; S, Start Compute.
3. Used only for internal representation of alphanumeric information operated on as described on pages 45 and 47.
4. The console controls and binary coded decimal columns have no relationship to the teletype code. However, the BCD figure is recorded on tape or paper in the corresponding teletype code. The console controls are used for decimal, octal, and binary input with the console keyboard; binary coded decimal is used for word output with the typewriter, visual readout, and tape punch. The + and - are used for input of binary 1's and 0's, respectively in the Command mode. On the visual readout BCD codes 1010, 1011, and 1101 are nonterminating blanks; 1100 is decimal point, and 1110 and 1111 are terminating blanks.
5. In Figures Shift switch on typewriter permits selection of tab or carriage return operation.
6. Decimal Point code 1100 and Terminate code 1111 are preferred. Terminate codes are neither typed nor punched.
7. The blank does not represent nor cause any action.

Fig. 21. RECOMP Codes and Representations

Control Codes. During control codes, special one-character instructions are interpreted immediately upon detection by the reader. Examples of these codes are the S (start) and H (halt) commands as well as the format codes C (command, i.e., instruction), L (location), N (decimal number), and F (alphanumeric), which specify the type of conversion which the computer is to perform on succeeding information.

Before or after a location entry, a minimum of 12 sprocket holes should be punched to allow proper reading rate for the photoelectric reader.

Tape Loading

Loading tape into the reading unit is a simple task. A tape is placed into a basket-type friction magazine. The tape is fed from the lower half of the cannister through the reader and automatically returned to the top section of the cannister after reading. (See Fig. 20.) The handle locks the magazine in place and properly alines the tape in the reader. Interlocks inhibit any attempt to perform a read operation until the unit has been properly connected. The beginning portion of the tape may be physically connected to the last portion and a continuous loop of tape may be formed.

Tape Reading

The contents of a tape may be read into RECOMP memory in two ways: (1) by using the FILL mode or (2) under program control. Automatic entry of information is initiated by depressing the FILL key on the front control panel of the tape unit. Information will then be read into memory until a halt code or start code is encountered in the tape. The location of the initial storage location for the information being read in this mode is normally punched on the tape as the first information read, or the location may be manually entered from the control panel or typewriter.

As stated, information may be filled until a halt code is encountered on tape stopping all operations, or until a start code is detected; in the latter case fill is stopped and computation is started, beginning with the instruction found at the location specified by the location counter. The type of information, data, or instructions may be alternated without interruption of the fill mode by merely preceding each type with the proper format code, i.e., an F, L, C, or N. (For alphanumeric tape fill information, refer to page 47.)

Tape Reading Rate

The tape reading rate, when the tape has accelerated to its maximum velocity, is 400 characters/second. This is equivalent to reading full 80-column Hollerith cards at a 300/minute rate.

Information can be read from tape under program control. Any number of instructions may be performed between read instructions. Information can be read as needed. The computer will not attempt to execute any other instruction until termination of an input-output operation.

Paper Tape Punch

The paper tape punch used on RECOMP II punches information into paper tape at the rate of 20 characters/second. The tape punch can be operated under computer control by programed instructions or independently of the computer in tape preparation by the electric typewriter.

Verify Mode of Operation

Because there are occasions when double assurance of the accuracy of information input is a decided advantage, the VERIFY mode in RECOMP II allows the contents of storage locations to be compared against information contained on paper tape. The VERIFY mode is similar to the tape FILL mode, differing only in that no information from the tape enters memory during VERIFY.

The paper tape is loaded into the photoelectric reader in the same manner as when preparing for the FILL mode. When the tape has been advanced, just prior to reading the first character, the VERIFY button on the tape reader panel is depressed. This activates the tape mechanism and the indicator just above the VERIFY button indicates the verify condition. The verification then commences in command, number, or alphanumeric format. The contents of the storage locations are compared to the original data appearing in the tape but are not affected by the verification. If a discrepancy is detected between corresponding words of information (one from storage and one from tape), the computer will stop the tape being read, and the ERROR VERIFY light on the operator control panel will flash to signal occurrence of an error in reading from tape or in writing into memory. The light flashes until the ERROR RESET key on the control panel is depressed. The VERIFY mode may also be stopped at any time by depressing the HALT key on the tape reader panel or operator control panel. Each full word from the tape increases the location counter by one word. The VERIFY mode will be dropped immediately, upon entering a COMPUTE mode, and subsequent input operations referring to paper tape will proceed in the normal FILL mode. The COMPUTE mode may be initiated immediately after verification with a START (S) code punched into the tape. This code follows the last character of information or a location code and address at which computing is to begin. When no starting location is designated, computing begins at the location following the last character of information entered.

Trapping Mode of Operation

The algebraic sign of normally executed commands is always positive. Should the sign of the instruction be negative, the normal execution, governed by the 2 octal-digit operation code, does not occur. Instead, the following sequence takes place:

1. The location counter, currently containing the address of the negative instruction, is stored in the second command address portion of the X-register.
2. The location counter is set to zero.
3. The instruction register is filled with the instruction pair from the storage location 0000 (octal); and this instruction pair is executed.

In other words, the actual execution of a negative command does not take place. Of great importance, however, is the transfer effected to location 0000. The SAX instruction (refer to Command List, page 69), in conjunction with the negative command or trapping mode, permits simple linkages to subroutines. Because the source of the transfer to channel 00 is available in the X register, the SAX command stores the contents of the A register and simultaneously brings the linkage address into the A register by exchanging the contents of A with X. A short routine to examine the operation code which had the negative sign would usually follow this instruction and thereby determine what pseudo-operation is to be executed. It can be seen that, since neither the linkage address nor the contents of the A register is destroyed, a return to the sequence of operations which numerically follows the negative instruction is a simple program function that can be made a part of every subroutine. Also inherent in the trapping mode scheme is the possibility of greatly expanding the number of operations which can be performed by the computer. Simple 3-character mnemonic codes may be written as synonyms for those subroutines most frequently used by a programer or group of programers. By assigning negative operation of codes during assembly to these pseudo-operations, those instructions which actually perform the routines can be placed wherever convenient in storage and may be addressed simply by the appropriate pseudo-code. Coding may be significantly simplified by application of the principles of the trapping mode with a versatile subroutine library. For example, codes for the pseudo-operation SIN (sine of an angle) and CMP (complex multiply) are as easily written as the real operation codes ADD (add), SUB (subtract), and MPY (multiply). The trapping mode also substantially assists in debugging operations on new programs; by means of strategic negative instructions with their associated subroutines, the contents of certain locations and arithmetic registers may be displayed or typed at critical times during checkout. After the program has been checked out, the negative signs may be removed, and the storage space assigned to the diagnostic subroutines can be utilized for more general utility subroutines.

INTERRUPTIONS

During the execution of a program, various interruptions may occur due to the detection of errors in operation (exceeding certain operating limits or safety conditions). Interruptions may also be due to the operator's desire to alter stored information or the course of the program. Because more than one course of action will produce the same result, a brief resume of the more common interruptions and the restart, transfer of mode, and transfer of control types of operation is given on the following pages.

Halt Condition

A halt condition can be effected by the following means:

1. Computer control

   a. When the computer detects an overflow, unless followed by a TOV
   b. When conditions which affect safe operation occur, such as a large line transient sufficient to contribute to incorrect operation, or a high temperature which would affect the operational characteristics of the solid-state electronic components
   c. When a programed halt is received
   d. When the computer detects an output or verify error

2. Control panel control

a. Depressing the STOP key will cause a halt of any computing or input-output operation.
b. Changing the OPERATION SWITCH from CONTINUOUS to SINGLE-COMMAND will cause a halt after the current instruction is executed. Depressing the START button will cuase one instruction to be executed. Subsequently, a halt will follow each instruction executed, and only one instruction will be executed each time the START button is depressed while operating in SINGLE-COMMAND mode.
c. After an address has been set up on the rotary switches (for channel and sector) and the PRE SET STOP switch is used, a stop will occur after the particular instruction located at the indicated address has been executed.
d. If the TRANS STOP (transfer stop) switch is turned on, detection and interpretation of any transfer-of-control command will cause a halt before execution of the next instruction following the transfer of control. This stop will occur following conditional transfer of control commands, regardless of whether the conditions of transfer have been met.

Tape Control

1. Detection of an H, C, N, L, CR, or S (Halt, Command format, Number format, Location code, Carriage Return, or Start code) character, detected at a point where it may be interpreted as a control rather than as an alphabetic character, will cause the appropriate mode change or halt. For example, the S code will terminate a FILL or VERIFY mode, and initiate the COMPUTE mode with the address in the location counter.

2. When in the VERIFY mode, differences in the bit pattern of a translated word from tape and the corresponding word from memory will cause a halt.

3. Pressing the STOP button on the tape control panel causes a halt of any FILL or VERIFY activity.

Modification of Location Counter

Modification of the LOCATION COUNTER during the course of a program (following a HALT and/or whenever filling, verifying, or computing is to proceed from a particular location) may be accomplished by one of the following methods:

1. Control Panel. The LOCATION COUNTER may be filled with a new address while the computer is in the noncompute state by depressing the L key, followed by the 4 octal digits and 1 binary digit necessary to fill the register (using the numeric keyboard). The numbers will be properly entered by depressing the ENTER key.

2. Typewriter control. If the computer is in the noncompute state, the typewriter may be used as a means of setting up a location in the LOCATION COUNTER. This is accomplished by depressing the alphabetic L key, followed by the numeric keys representing the 4 octal digits and 1 binary digit. Entering the address is accomplished by next depressing the CARRIAGE RETURN key.

3. Paper tape control. When filling or verifying, with reference to command or number fill, encountering an L character will terminate the previous mode, and the next 5 characters will be considered to be the 4 octal digits and 1 binary digit needed to fill the location counter followed by a carriage return (CR) code.

PROGRAMING

For the RECOMP II to accomplish its main purpose of processing scientific and engineering data at high speeds, detailed instructions that indicate the processing path and the sequence of execution must be provided. These instructions (or program) must be carefully transcribed in detail, because the computer is able to follow only a prescribed course of operations. Thus, an incorrect sequence will lead the computer into a futile attempt of problem solution.

The logical design pattern of RECOMP calls for obedience of 49 instructions having logical, arithmetic, control, information transfer, and input-output functions. This provides a wide latitude of methods in developing programs or routines for the solution of specific problems.

Ease of programing many scientific problems for RECOMP II is facilitated by a complete set of floating point instructions, including a floating point square root instruction, in addition to fixed point instructions. Built-in echo checking circuits assure accurate recording of output. A verify mode of operation checks accurate loading of programs or data from tape. Once a program has been written, the instructions are usually punched into paper tape and read into the system through the photoelectric reader.

INFORMATION STORAGE

All information instructions and data are stored in binary form on the magnetic disk memory unit, regardless of input medium, with 40 binary positions representing 1 word. On input, the conversion of information to this form is completely automatic from the familiar arabic decimal system, or other configurations described under format conversion, to the binary language of the machine. For output, the programer may design the output format which best satisfies his needs. Results may be delivered in decimal, instruction, or alphanumeric form.

After entering and storing the instructions or program, the processing cycle begins. The disk provides space for the storage of the information needed for the solution of most problems.

Magnetic Memory

RECOMP's internal storage unit, or memory, is a rotating (3450 rpm at 60 cps) disk coated with iron oxide. (See Fig. 4.) Through the read-write heads this surface is magnetized to represent binary values so that a change in flux in an area represents a binary 1 for the value of that position, while no change in flux indicates a binary 0. An autolubricated air bearing is generated in operation between a plate, in which magnetic read-write heads are distributed, and the rotating disk. The air bearing permits an extremely close spacing to be maintained between the rotating disk and the headplate. This highly efficient recording system permits recording densities of 360 bits/inch. In addition, adjacent channels or recording areas can be spaced very closely, resulting in an extremely compact, low-power requirement memory.

Main Memory

The main memory is divided into 65 concentric channels (64 information storage and one clock). Information channels are numbered octally from 00 through 77. With one exception, every channel contains 64 sectors, or addressable locations also labeled octally from 00 through 77, each holding a single computer word. The last 16 words of channel 77_8 are not considered part of the main memory. Information stored in main memory is nonvolatile and will remain on the disk after power is turned off. Data or instructions may be entered through any of the FILL modes into any location except the last 16 words of channel 77_8. These locations are actually physically separated from main memory and are delegated a special function, described below.

Rapid-Access Loops

The two rapid-access recirculating loops, designated the L and V loops, each with an 8-word capacity, are available for use by the programer. They are addressed as though they were the next-to-last and last eight words of the sixty-fourth channel, though physically located in a separate region. Storage locations in the L loop are addressed sequentially from 7760. 0 through 7767. 1 and storage locations in the V loop are addressed sequentially from location 7770. 0 through 7777. 1. Though physically located on a different portion of the magnetic disk, these two loops may be considered to be an extension of the 4080 words of main memory, and as indicated, may be addressed as such. While the location of read and write heads for the L and V loops makes possible their utilization as rapid-access storage alone, the loops generally serve the function of providing a portion of storage where a series of instructions may be completely executed in a more rapid manner than in main memory. This is because the average access time to locate the instructions and data stored is much less than in main memory. Included in the list of instructions, available to the programer to enable full utilization of these rapid-access loops, are 4 commands which cause the computer to "copy" contents of 8 consecutive storage locations from main memory into the L or V loop, or to copy from either of the loops into main memory. Because the two 8-word loops form one continuous 16-word loop, up to 32 sequential instructions may be placed in the loops for execution. Short iterative loops, subroutines, etc, can be placed in L and V, as examples of ideal utilization of the loops.

NOTE

Because of normal sequencing of instructions, after executing the instruction found in the last main memory location (7757. 1) control automatically goes to the first instruction on the L loop, 7760. 0, unless otherwise specified by the program. From the last instruction on the L loop (7767. 1), control goes to the first instruction on the V loop (7770. 0) automatically. From 7777. 1 control transfers to 0000. 0.

RECOMP Words

Because the term "number" has an ambiguous meaning, in that it denotes a single numeric symbol or a group of such symbols, the designation "word" has acquired acceptance in computer terminology as an indication of a standard length group of digits.

The 40-bit positions comprising a RECOMP word are shown in Fig. 6 and 7 where S refers to the single bit allocated the sign position; 1 refers to bit position 1 (i. e. the first bit position to the right of the sign bit), 2 refers to bit position 2, etc.

A word may be stored in any one of 4080 main memory locations or the 16 rapid-access storage locations, which are loaded only by the special copy instructions.

Data Word

When the word contains numerical data and the sign position contains a binary 1, the word is positive; if it contains a 0, the word is negative. As stated earlier, when used as a binary number (with algebraic sign), a word is equivalent to slightly more than 11 decimal digits. Because any 3 binary digits can be exactly expressed by 1 octal digit, a word may be considered to consist of 13 octal digits to express the absolute value of the numeric quantity, and an algebraic sign to designate the positive or negative condition of that quantity.

Instruction Word

The two commands of a RECOMP instruction pair consist of an operation portion with an associated sign bit, and an address portion, sometimes designated the operand address.

Two instructions or commands are contained within one word as shown in Fig. 10. Although the sign bit associated with the first instruction of a pair should be made positive for normal computing, a negative sign can be used to indicate that a trapping mode operation is desired. From Fig. 7 it may be seen that the operation code requires 6 bits and the address portion requires 13 bits. The operation code is used to designate the particular operation the machine will perform. The address part will have one of the following meanings:

1. The number of binary positions that information in the A register is to be shifted to the left or right
2. The location in which information is to be stored by the instruction
3. The location of information which is to be used by the instruction
4. The number of characters to be read from an input device
5. The location of the next instruction to be executed

It is emphasized that the RECOMP II operates as a binary machine and that the use of octal notation in no way affects this operation. Octal notation is used by the programer as a convenient method of expressing long sequences of binary ones and zeros.

In addition to the storage channels, the disk also contains the basic timing channels and the arithmetic registers.

The timing channels are two permanently recorded tracks that furnish basic timing controls for the computer. The clock channel controls the triggering of the logical gates on input to the circuits. The origin channel has a single pulse that resets counters used in memory addressing.

Registers

The registers are five individual recirculating loops, each containing one word. The information path is indicated in Fig. 3. All control and processing of information are accomplished through the action of information within these number registers. The contents of the various recirculating registers are not retained when power to the computer is off. The contents of the registers will be discussed for each operation shown in the command list.

The A Register

The A register, which is used in all arithmetic operations, is an accumulator register consisting of 39 bits and a sign. Following some instructions (for example, addition or subtraction) the contents of the A register may overflow from position 1; that is, the result of the operation may be too large to be contained within the register consequently losing that information which overflows. When such an overflow occurs, the overflow trigger is turned on. When this trigger is on, only one instruction, TOV (transfer on overflow), can be executed which will prevent the computer from stopping. If any other instruction is attempted, the computer will stop and cause the overflow error light to flash. This safeguard prevents further arithmetic operations from being performed with data which have lost the most significant binary digit. Pressing the ERROR RESET button will turn the overflow trigger off.

The C Register

The C register consists of 39 bits and a sign and contains the commands to be executed. This register is used by the internal operations of the computer and is not available for use by the programer.

The R Register

The R register consists of 39 bits and a sign, and has three major uses:

1. After the execution of a fixed-point multiplication instruction, the R register contains the less significant part of the product and, in this respect, may be considered an extension of the A register.

2. Before the execution of a fixed-point division instruction, the least significant half of the dividend appears in the R register and after the division, the remainder appears in this register.

3. The exponential part of the result of all floating point instructions is placed in the R register.

The B Register

This auxiliary register is used by the computer to provide intermediate storage of numbers and is not available for use by the programer.

The X Register

This auxiliary register is used for intermediate storage during transfer or floating point operations. After a transfer of control operation, the X register will contain the 13 bit location which was in the location counter at the time of interpretation of the transfer operation. Execution of floating point instructions or additional transfer instructions will alter the C (X); therefore if it is desirable to save the contents of X after a transfer command it should be accomplished before execution of any floating point or additional transfer of control command.

RECOMP COMMANDS

This section of the programing guide sets forth in detail the comprehensive list of instructions of RECOMP II. Through the efficient use of these commands, the full potential and flexibility of the computer are realized. From the preceding material, it is evident that no one operation effects input to the registers, obedience to commands, and output of results. Rather, it is through a series of operations called for by the computer commands. Various combinations of commands may achieve the same results, but the most efficient program is the one that achieves the goal in the least amount of time, without loss of the desired degree of accuracy.

To prepare an efficient program, therefore, careful attention must be paid to the limits and functions of individual instructions.

EXECUTION TIMES

The total time required to carry out one command stored in the computer is a function of the time required: (1) to obtain the command from memory, (2) to obtain the operand from memory, and (3) to execute the command (refer to pages 22 and 103). Because two instructions are stored in a single word in RECOMP II, no "look-up" time is required to obtain the second instruction in a word. Access time depends upon whether the instruction or operand was obtained from main memory or the L or V loops, whether the command is available as the second instruction of a word, and also upon the relative position of the reading heads on the disk with respect to the location of the command or operand. It may be noted also that the time required for execution of the transfer instructions TSB, TSC, and TSD is 3 word times if the condition for transfer is met, and 2 word times if the computer proceeds to the next instruction in sequence and the condition for transfer is not met. Thus, no simple algebraic expression exists which encompasses all of the general timing considerations required for such an analysis.

INSTRUCTION CONFIGURATION

In the command register, a word from an address in storage will be interpreted as two instructions (Fig. 7). The location counter on the control panel indicates which of the instructions is being executed, as well as the location from which the instruction pair was selected.

Sign

The sign of the instruction is first examined. A negative sign will cause the operation to be ignored and initiate entry to a "Trapping Mode." The normal interpretation of the six digit binary operation code will follow a positive sign.

Address

Depending on the particular operation code preceding the address portion, the first 12 binary digits of the address portion may refer to a location in memory, a control number to indicate shifts of register contents for scaling, a specific binary configuration for a character to be output, input configuration control, or location of information for output.

Half Word Bit

The half word bit or thirteenth binary digit of the address has significance where (1) it is necessary to refer to the left or right half word in a register or (2) in specific readout and punch or type commands where it controls output format. On other occasions, it has no effect, whether 0 or 1. In written descriptions, the half word bit is generally dropped from addresses where a reference to this binary digit would be irrelevant. Although a reference is not necessary, the space assigned to this bit in a computer must be filled and is normally coded with a 0. For example, in manual entry of an instruction through the keyboard, after the normal depression of the four keys to represent the channel and sector digits of an address, a 0 or 1 key should be depressed. The binary representation of each key depressed will be properly entered and alined by the computer when the balance of the digits is keyed in and the ENTER key is depressed. Omitting the spare bit, even though it might not be interpreted during execution, results in an incorrect alinement of the input information, thus changing the value of other digits which are subsequently entered.

To refer to a specific location in memory one may express the 12 binary digits which precisely locate the channel and sector by their octal equivalent. As an example, if the location 5612 (octal) were the address in reference, this would appear in the address portion of an instruction in the computer, as 101110001010 in binary with the half word bit (the least significant binary digit of the address) noted separately. If the example refers to a whole-word location, the expression 5612 locates the word in memory. If the operation called for reference to a specific half word, a binary 0 or 1 to indicate a left or right half word would be required as in the case of the control transfer or address modification commands. In the case above, the designation 5612.0 would refer to the left half of the word found at octal location 5612 in memory. In the same manner a spare bit 1, if it had appeared in the binary configuration, would have referred to the right half of the word found in location 5612 (octal). The address would have been expressed 5612.1. The decimal point does not appear in memory or display, and does not imply an octal fraction, but is merely a convenient method of separating the half word bit from the octal expression in written references. The punch and type commands require the half word bit to be indicated for another purpose -- to indicate a format conversion to be performed on information located at the specified address. A 1 indicates decimal and a 0 indicates command format.

Mnemonic Notation

In explaining the RECOMP II command list, a number of symbols are utilized. These symbols eliminate unnecessary wordage and permit the reader to concentrate on perfecting his knowledge of command usage. The symbols must be replaced by the actual numerical configuration before the computer can interpret and execute the program. This translation may be accomplished manually or by assembly-type programs.

In the list of command descriptions, whenever the address portion of an instruction refers to some arbitrary location in memory (the actual location will be a function of a particular program), it is referred to as "W." For example, in the instruction ADD W, the W is not added, but merely states where the addend is located. If an address portion does not refer to a location, the letter "M" may be used to avoid ambiguity. In such cases, the value assigned M is the actual configuration of a character to be output. Several operations have operands which refer to either the left or right half of a word. The symbol W may be modified accordingly to allow "W. 0" or "W. 1" to indicate the left or right half of word location W.

The contents of the entire word at location W are referred to as C (W). The same abbreviation may be used when referring to any register, eg, C (A) or C (R). When reference to specific bit positions in a word is necessary, it is accomplished by subscripting the previous symbols with the decimal number of the desired bit positions. The subscript S designates the sign position. The symbol $C\ (W)_{S,\ 1,\ 2,\ 3}$ or $C\ (W)_{S,\ 1-3}$ would refer to the binary digits to be found in the first sign position plus the first three bit positions of the word at location W. Because the symbols W. 0 and W. 1 have been used to indicate the left or right half word, $C\ (W)_{S,\ 1-19}$ and $C\ (W)_{20-39}$ may be used synonymously with C (W. 0) and C (W. 1). T represents access time. S indicates the number of binary places the information is shifted.

The three letter mnemonic codes which have been assigned to represent the actual numeric equivalent of the operation code were selected as most briefly and clearly conveying the meaning of each command.

The heading for each instruction or command gives the title and the machine code (octally) and the three letter mnemonic code for the operation portion. If the address portion of an instruction has no significance, a dashed line will appear in that position in the heading.

COMMAND LIST

The following list comprises the commands for the RECOMP computer:

Clear and Add

00 CLA W

The C(A) are replaced by the C(W). The C(W) remain unchanged. The R register is not affected by this command.

Add

01 ADD W

The C(W) are algebraically added to the C(A). The sum replaces the original C(A). If the sum is zero, the sign of the original C(A) will remain. Should the capacity of the A register be exceeded during the summation, the overflow indicator will flash and the computer will stop unless the next command is a transfer on overflow (TOV). The C(W) and C(R) remain unchanged.

Clear and Subtract

02 CLS W

The C(A) are replaced by the negative of C(W). The C(W) are not changed; neither are the C(R).

Subtract

03 SUB W

This instruction algebraically subtracts the C(W) from the C(A), the difference replacing the original C(A). If the difference is zero, the sign of the original C(A) remains. The C(W) and the C(R) remain unchanged by this instruction. Overflow is possible as a result of this operation. If it occurs, the overflow indicator will flash and the computer will stop unless the next command is a TOV.

Multiply

11 MPY W

This instruction multiplies C(W) by C(A) and holds the double-length 78 bit product with the 39 most significant bits stored in A and the 39 least significant bits in R. The C(W) are unchanged and the sign positions of A and R contain the sign of the product. It is not possible to overflow by this operation.

Multiply and Round

13 MPR W

This instruction multiplies the C(W) by C(A). The 39 most significant bits of the product, in A, are rounded. The R register will contain the 39 least significant bits with the first bit reversed. The sign position in R will be the same as A.

Divide

22 DIV W

The C(A) and C(R) are divided by the C(W). In this instruction the contents of the A and R registers are used as a continuous 78 bit dividend with the 39 most significant bits in the A register and 39 least significant bits in the R register. The sign associated with the 78 bit dividend is in the A register. The sign of the R register prior to execution of the instruction has no effect on the results.

After execution of the divide instruction, the 39 bit quotient plus its sign supplants the C(A). The remainder replaces the C(R) and the sign of the remainder is the same as that of the dividend. The C(W) are unchanged.

It is possible for an overflow condition to occur as a result of this operation. It occurs if a division is attempted when the magnitude of the dividend is greater than or equal to the magnitude of the divisor which is in W.

Divide And Round

23 DVR W

This instruction is executed in a manner similar to the DIV instruction. The 39 most significant bits and the sign of the dividend are placed in A. The 39 least significant bits of the dividend are placed in R.

After execution of the DVR operation, the 78 bit dividend in A and R is replaced by a rounded 39 bit quotient and its sign in A. R contains the same digit configuration as the rounded 39 bit quotient in A; however, the sign of the R register is the same as the original 78 bit dividend.

If the dividend is greater or equal to the divisor, an overflow will occur.

In rounding, the magnitude of the remainder in R is compared with the magnitude of the divisor. If the C(R) are greater, the quotient is increased by a 1 in position 39. If this causes the A register to overflow, then the overflow trigger will be turned on. This rounding process replaces the remainder in R with the rounded quotient.

Divide Single Length

20 DSL W

This instruction operates in a manner similar to the DIV instruction above. In this case only the 39 bits plus sign in the A register will be divided by the C(W). Any information in R prior to execution will not be a factor in the division. The quotient and sign replace the initial contents of A, while the contents of R are replaced by the remainder. The sign of the remainder will be the same as the original dividend.

If the magnitude of the dividend is equal to or greater than the divisor, an overflow will occur.

Divide Single Length and Round

21 DSR W

Operation will proceed in a manner similar to DVR operation, except that only the 39 bits plus sign of the dividend which is in A will be divided by the C(W). The rounded quotient will replace the initial C(A). The contents of R will be replaced by the rounded quotient with the same sign as the original dividend.

The C(W) are unchanged and an overflow will occur if the dividend is equal to or greater than the divisor.

Square Root

25 SQR W

This instruction extracts the square root of the quantity found at W. The square root will replace the contents of the A register, with a binary scale of one half that of the number in W. The instruction assumes the binary scale of the number in W to be an even number. If it is not an even number the result will be incorrect. That is, the number may be off by a factor of 2 before the extraction; consequently, the result will be off by $\sqrt{2}$.

In the execution of the instruction the C(R) are destroyed and the sign of the A register is made positive. If the C(W) are minus the overflow trigger is turned on and the computer stops unless followed by a TOV. The C(W) remain unchanged.

Store

60 STO W

This instruction will cause the sign and 39 bits of information in the A register to supplant the sign and 39 bits at location W. This destroys the information which was previously in location W. The C(A) and C(R) are unchanged.

Store Address

42 STA W

This instruction will store an address from either the left or right half word of the A register into the respective left or right half word at the location W. When this word containing this STA command is in the instruction register for interpretation and execution, the thirteenth or least significant bit of the address portion of the command is examined. If this bit position is occupied by a zero, information from binary positions 7 through 19 in the A register replaces the contents of the corresponding numbered positions at memory location W. If this examined bit position contains a one, the bits from binary positions 27 through 39 of the A register replace the contents of the correspondingly numbered positions at memory location W. In either case, the C(A) and all remaining bits of C(W) remain unchanged.

The number of word times required for this instruction is (32 +T) when W is in main memory. When W is in the L or V loops the execution time is 4 + T to 8 + T; the variation is due to the physical location of the read and write heads.

SHIFTING AND LOGICAL COMMANDS

Accumulator Left Shift

41 ALS M

The address portion of this command does not refer to a location in memory, but indicates the number of binary positions through which the configuration in the A register is to be shifted. Only the six bits in the sector part of the address portion of the command are interpreted. On execution of this command the contents of the A register are shifted left the number of binary positions corresponding to the number in the six sector bit positions of the operand. Positions left vacant as the quantity in A shifts left are filled with zeros. Any bits shifted beyond the limits of the A register are lost. The sign position of A and the C(R) are unaffected by this command. It is not possible to turn on the overflow trigger by this command.

Accumulator Right Shift

40 ARS M

The address portion of this command does not refer to a location in memory but indicates the number of positions through which the configuration in the A register is to be shifted. Only the six bits in the sector part of the address portion (M) of the command are interpreted. On execution of this command, the contents of the A register are shifted right a number of binary positions corresponding to the octal number in the sector portion of M. Positions vacated to the left of the number being shifted in the A register are filled with zeros. Any bits shifted beyond the limits of the A register are lost. The sign position of A and the C(R) are unaffected by this command. It is not possible to turn on the overflow trigger by this command.

Exchange A and R

43 XAR --

The address portion of this command is immaterial. The sign and 39 bit contents of A are exchanged with the sign and 39 bit contents of R.

Store A and Exchange A and X

15 SAX W

This command will first cause the sign and 39 bits in A to supplant the sign and 39 bit contents at W. The C(X) and the C(A) are then exchanged.

The SAX command is useful for retrieving the address of transfer type commands because, following their execution, the X register contains the location of the command from which transfer of control was made. Because there are a number of commands which alter the contents of the X register, the SAX command should always be located in the first address of the sequence of commands which will be executed following a transfer operation.

Extract

33 EXT W

All information in the A register, including the sign position, will be compared to the corresponding bit positions of the word in location W.

After execution, every place the extractor in W contains a zero, the corresponding bit position in A will contain a zero and every place the extractor contains a one, the corresponding bit position in A will remain the same as before extraction. In other words, if the corresponding bit positions of both W and A are zero the result in A will be zero; if either W or A is zero, the result in A will be zero; if both W and A are one, the result in A will be one.

Because a plus sign is represented by a binary one and a minus sign by a zero, the result placed in the A register will be positive after executing an EXT command only if both C(A) and C(W) are positive.

The C(W) are not changed by this operation. Likewise, the C(R) are neither changed nor involved.

FLOATING POINT COMMANDS

All floating point commands proceed on the assumption that the floating point numbers involved in the operation will each consist of a one-word fraction (mantissa) and a one-word exponent (characteristic). Reference to floating point numbers in storage may be accomplished by addressing only the fraction portion W. The computer assumes the exponent to be located at the address W + 1. (W and W + 1 must be in the same channel.) Consequently, the position of the floating point number in the arithmetic registers A and R necessarily corresponds to the extent that the C(A) are a fraction and the C(R) are its exponent. The exponent may be plus or minus in value.

Overflow of the fractional portion is not possible on floating point numbers. Overflow of the exponent portion is possible, but there will be no indication of this condition.

The exponent is limited only in that its binary representation should never be so large that it extends into binary positions 1 and 2 to the right of the sign position. The remaining 37 positions may be used for the expression of the exponent, thus permitting operation on an extremely large range of numbers. If these exponent limits are exceeded, incorrect execution will occur.

If any sequence in a program requires saving the contents of the X register, this should be performed prior to the execution of floating point operations. The reason is that floating point operations modify the contents of the X register.

Floating Clear and Add

30 FCA W

The contents of W and W + 1 replace the contents of the A register and R register respectively. Normalization of the floating point number does not occur on this operation.

Floating Clear and Subtract

34 FCS W

The contents of W and W + 1 will replace the contents of the A and R registers. The sign of C(A) is the reverse of the sign of the C(W). The sign of C(R) will be identical with the sign of C(W + 1). Normalization does not occur on this instruction.

Floating Add

04 FAD W

The C(W) and C(A) are properly and automatically positioned along with adjustment of their respective exponents before addition occurs. After addition, the normalized sum replaces the C(A), and the exponent corresponding to this sum replaces the C(R). The C(W) and C(W + 1) are unaltered.

The details of how this is accomplished are as follows:

1. The C(W) and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. The C(A) and C(B) are shifted to the right one bit position and the C(R) and C(X) are increased algebraically by a 1 in the least significant bit position.
3. If the magnitude of the C(A) or the C(B) is zero after the shift, the computer proceeds to step 4. If neither the magnitude of the C(A) nor the C(B) is zero, the C(A) or C(B) are shifted to the right one binary position. A 1 is added to the least significant position of the corresponding exponent part in the R or X register until the C(R) and C(X) are made equal or until the C(A) or C(B) are made zero. The C(A) are shifted when the C(R) are less than the C(X); the C(B) are shifted when the C(R) are greater than the C(X).
4. The C(A) and the C(B) are, then, added algebraically and the sum replaces the C(A). Note that overflow cannot occur from this step since the original C(A) and C(B) were shifted one bit position to the right in step 2.
5. The floating point number, with fraction part in the A register and exponent part in the R register, is then normalized by shifting the C(A) (exclusive of the sign bit) to the left and subtracting a 1 from the least significant bit position of the C(R) for each position shifted.

Floating Subtract

06 FSB W

The C(W) and C(A) are properly and automatically positioned along with adjustment of their respected exponents before subtraction occurs. After subtraction, the normalized difference replaces the C(A) and the exponent corresponding to this difference replaces the C(R). The C(W) and C(W + 1) are unaltered.

Floating subtraction takes place in a manner similar to the FAD instruction. The details are as follows:

1. The C(W) and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. Steps 2 to 5 of the FAD instruction are performed, except that a subtraction (instead of addition) of the fractional parts occurs in step 4.

Floating Multiply

07 FMP W

The floating point number contained in the A and R registers is multiplied by the floating point number contained in W and W + 1. This forms a normalized floating point product, the fractional part of which replaces the C(A) and the exponent part of which replaces the C(R). The C(W) and C(W + 1) are unaltered.

The details of how floating multiplication is accomplished are as follows:

1. The C(W) and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. The C(A) are shifted one bit position to the right and a 1 is added algebraically to the C(R) in the least significant bit position.
3. The C(X) are added algebraically to the C(R) and the sum replaces the C(X). Exponent overflow may occur at this time but is not indicated in any manner.
4. The C(A) are multiplied by the C(B) and the product thus formed (rounded to the most significant 38 bits) replaces the C(A) and C(R).
5. The C(X) replaces the C(R).
6. The floating point number, with the fractional part in the A register and the exponent part in the R register, is then normalized

by shifting the C(A) to the left and subtracting algebraically a 1 from the least significant bit position of the C(R) for each position shifted.

Floating Divide

05 FDV W

The floating point number contained in the A and R registers is divided by the floating point number in W and W+1. This forms a normalized floating point quotient; the fractional part replaces the C(A) and the exponent part replaces the C(R). The C(W) and C(W + 1) are unaltered.

The overflow trigger will be turned on if the fractional part of the divisor contains all zeros in bit positions 1 through 39, or if the divisor is not in normalized form and its fractional part is less than or equal to the fractional part of the dividend after the right shift described in step 2.

The details of how floating point division is accomplished follow:

1. The C(W) and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. The C(A) are shifted one bit position to the right and a binary 1 is added algebraically to the C(R) in the least significant bit position.
3. The C(X) are subtracted algebraically from the C(R) and the difference replaces the C(X). Exponent overflow may occur at this time but is not indicated in any manner.
4. The C(A) are divided by the C(B) and the quotient thus formed is rounded to the 39 most significant bits. These 39 bits replace C(A) unless the overflow trigger is turned on under the conditions described above. During the last word time of this step the C(X) replace the C(R).
5. The floating point quotient, with the fractional part in the A register and the exponent part in the R register, is then normalized by shifting the C(A) to the left and subtracting algebraically a 1 from the least significant bit position of the C(R) for each position shifted. Finally, the $C(R)_{1,2}$ are replaced with zeros.

Floating Square Root

44 FSQ W

This instruction takes the square root of the floating point number in W and W + 1. The resultant normalized square root is a floating point number; the fractional portion replaces the C(A) and the exponent portion replaces the C(R). The C(W) and C(W + 1) are unchanged. If this instruction attempts to operate on a negative number, the overflow trigger will be turned on.

The details of how the floating point square root operation is performed are as follows:

1. The C(W) and C(W + 1) replace the C(B) and C(X), respectively. Simultaneously, the exponent part (which is placed in the X register) is examined to prepare for step 2.
2. If the exponent part is odd, the C(B) are shifted one bit position to the right, a 1 is added algebraically in the least significant bit position of the C(X), and the computer proceeds to step 3. If the exponent is even, the C(B) are not changed and the computer proceeds to step 3.
3. The operations used for a fixed point square root (SQR) are used to obtain the square root of the C(A) excluding the sign bit.
4. During the last word time required to extract the square root, the C(X) are shifted right one binary position and the result replaces the C(R). The sign of C(A) is made plus and, if the sign of the C(W) is minus, the overflow trigger is turned on.

The number of word times required for the execution of this instruction is 43 if the C(W + 1) consist of an even number and 44 if the C(W + 1) consist of an odd number.

Floating Normalize

45 FNM

The address portion of this instruction has no significance. Execution of this instruction proceeds on the assumption that the fractional portion of the unnormalized number is in the A register and the exponent is in the R register. The FNM instruction results in the binary bit configuration in the A register being shifted to the left until the most significant 1 bit occupies the leftmost binary position. The number of binary positions through which this bit has shifted is then subtracted from the C(R). This new number in R is then the correct exponent of the normalized fraction in the A register.

If the fractional number in A is found to consist of all zeros, this instruction will not attempt to operate on it and the computer will proceed to the next instruction in sequence.

The sign of the fractional number in A is not effected by this operation.

Floating Store

35 FST W

This instruction causes the C(A) and C(R) to replace C(W) and C(W + 1), respectively. The C(A) and C(R) are not changed by this instruction.

CONTROL COMMANDS

Halt and Transfer

77 HTR W

This instruction causes the computer to stop after replacing the contents of the location counter with the address W. If the START key on the operator's console is depressed, the machine takes the next instruction from location W and proceeds from there.

Transfer on Zero

50 TZE W

The contents of the location counter replace the C(X). If the C(A) are zero, the machine takes the next instruction from location W and proceeds from there; if the C(A) are nonzero, the machine takes the next instruction in sequence. Because the algebraic sign of a number must be expressed as a binary digit for computer operation, it is possible to have either a positive or negative zero. A TZE instruction will cause a transfer in either case.

Transfer on Plus

52 TPL W

The contents of the location counter replace the C(X). If the

S position of the A register is positive (contains a one), the machine takes the next instruction from location W and proceeds from there; if negative, the machine takes the next instruction in sequence.

Transfer on Minus

51 TMI W

The contents of the location counter replace the C(X). If the S position of the A register is minus (contains a zero), the machine takes the next instruction from location W and proceeds from there; if positive, the machine takes the next instruction in sequence.

Transfer on Overflow

53 TOV W

The contents of the location counter replace the C(X) after which the status of the overflow trigger is tested. If the trigger is on, the machine turns it off, takes the next instruction from location W, and proceeds from there. If the trigger is off, the machine proceeds to the next instruction in sequence. This is the only instruction which can be executed without causing the machine to stop and flash the overflow indicator on and off when the overflow trigger has been turned on by the preceding instruction.

Transfer

57 TRA W

The contents of the location counter replace the C(X) after which the computer takes its next instruction from location W and proceeds from there.

Transfer on Sense Switch B

54 TSB W

The contents of the location counter replace the C(X) after which, if sense switch B is on, the computer takes its next instruction from location W and proceeds from there. If off, the computer takes the next instruction in sequence.

Transfer on Sense Switch C

55 TSC W

This instruction operates in the same manner as the TSB instruction except that sense switch C is used instead of sense switch B.

Transfer on Sense Switch D

56 TSD W

This instruction operates in the same manner as the TSB instruction except that sense switch D is used instead of sense switch B.

Type Character

72 TYC M

A single character is typed corresponding to the address part of this instruction according to the codes given in Fig. 21. These five bits correspond to full word bit positions 14 through 18 in the left half instruction or 34 through 38 in the right half instruction. The remaining bit positions of the address part of this instruction are not examined and do not affect its operation. As a character is typed, an "echo" signal is generated and is checked against the desired bit configuration within the computer. If a discrepancy occurs, the output error light on the operator's console flashes and the computer enters the noncompute mode. From the standpoints of reliability and accuracy, it is important to note that the echo signal and check are made after the selected type bar has started moving toward the platen. Words stored in alphanumeric format can be typed one character at a time by placing each in the address portion of this instruction.

Punch Character

74 PNC M

A single character is punched in the 5-channel paper tape corresponding to the address part of this instruction according to the codes given in Fig. 21. These five bits correspond to full word bit positions 14 through 18 in the left half instruction or 34 through 38 in the right half instruction. The remaining bit positions of the address part of this instruction are not examined and do not affect its operation. A 1 in a bit position specifies a hole to be punched, while 0 specifies no hole to be punched. As a character is punched, an "echo" signal is generated and is checked against the desired bit configuration within the computer. If a discrepancy occurs, the output error light on the control console flashes and the computer enters the noncompute mode. From the standpoints of reliability and accuracy it is important to note that the echo signal and check are made after the punch pins have started moving toward the paper tape. Words stored in alphanumeric format can be punched one character at a time by placing each in the address portion of this instruction.

Punch and Type Character

76 PTC M

A single character is punched and typed simultaneously corresponding to the address part of this instruction according to the codes given in Fig. 21. These five bits correspond to full word bit positions 14 through 18 in the left half instruction or 34 through 38 in the right half instruction.

The remaining bit positions of the address part of this instruction are not examined and do not affect its operation. As a character is punched and typed, an "echo" signal is generated and is checked against the desired bit configuration within the computer. If a discrepancy occurs, the output error light on the operator's console flashes and the computer enters the noncompute mode.

From the standpoints of reliability and accuracy it is important to note that the echo signal and check are made after the selected type bar and punches have started moving. Words stored in alphanumeric format can be typed and punched one character at a time by placing each in the address portion of this instruction.

Punch Word

14 PNW W

If the least significant bit of the address part of this instruction, half word bit, is a 0, the C(W) are punched in COMMAND format which requires 16 characters to be punched per word: 1 for sign, 2 for operation, 5 for address, 1 for sign, 2 for operation, and 5 for address. No space or carriage return code is punched after the sixteenth character.

If the least significant bit of the address part of this instruction, half word bit, is a 1, the C(W) and the C(W + 1) are punched in teletype code configuration corresponding to the character represented, using the 4-bit BCD codes and decimal format as explained under the display instruction DIS. For correct display, the information must be put into

BCD format by programing. From this figure it may be seen that only the sign and first 32 binary positions of W and the first 28 binary positions of W + 1 are interpreted. In addition to the decimal characters 0, 1, 2, . . . 9, four special BCD characters are available. The decimal point is obtained by using the BCD code "12_{10}"; the space is obtained by using the BCD code "10_{10}"; the terminate code is indicated by the BCD code "15_{10}"; and the carriage return is obtained by the BCD code "11_{10}". Under decimal output, if the C(W) are " + ", a Fig Shift is punched instead of the sign. This permits punching up to 15 digits without interruption for sign. The " + " is typed and/or punched by a Type and/or Punch Character instruction. If " - ", the " - " is punched. An echo check is performed on each character as in the PUNCH CHARACTER command. The C(R) is destroyed.

Type Word

12 TYW W

This instruction operates in the same manner as the Punch Word instruction (PNW) described above except that the typewriter, instead of the punch unit, is used. An echo check on each character is performed as in the TYPE CHARACTER command. The C(R) is destroyed.

Punch and Type Word

16 PTW W

This instruction operates in the same manner as the PNW except that each character is punched and typed simultaneously. An echo check is performed on each character as in the PUNCH AND TYPE CHARACTER command. The C(R) is destroyed.

Read Y (Typewriter or Other Input Devices)

71 RDY W

If W is less than 7760.0 the computer enters the INPUT state, the contents of the location counter are replaced with W, and control is transferred to the Y input device. In this mode, information is read continuously until carriage return (CR) code is received. At this time, information is stored at the designated location, the location counter is increased by one full word, and reading is continued. This method of reading and storing in successive locations continues until a start code is supplied. After receipt of a start code, control will be returned to the location then specified in the location counter. The contents of the location may be changed by entering the location code followed by the octal location desired.

The input format used is specified by supplying the COMMAND or NUMBER code. When the COMMAND code is used, the first character read must be algebraic sign (+ or -) followed by 7 characters for each of two commands and then an ENTER code. The information thus entered is stored in the location indicated by the location counter which is then increased to the next full-word location. If the NUMBERS code is supplied, the first character read must be an algebraic sign (+ or -) followed by 0 to 11 characters.

If W is greater than or equal to 7760.0, a number of 5-bit characters is read from the Y input device, and enters the A register from the least significant end. As each character is placed in the A register, the previous contents are shifted to the left, the bits leaving bit position 1 enter S position, and the bits shifted out of S are lost. The number of characters to be read is specified according to the following convention:

7760.0	=	8	characters
7761.0	=	1	"
7762.0	=	2	"
7763.0	=	3	"
7764.0	=	4	"
7765.0	=	5	"
7766.0	=	6	"
7767.0	=	7	"

After executing this instruction, the machine proceeds to the next instruction in normal sequence with the characters read from the Y input device appearing in the A register. While executing this instruction the compute light is turned off. However, the computer does not enter the INPUT state and, after receiving the last character, the compute light is turned ON.

Read Z (Photoelectric Paper Tape Reader)

73 RDZ W

This instruction operates in the same manner as the Read Y instruction (RDY) except that the paper tape reader is used instead of the Y input device.

Figure 21 gives the code used on paper tape and the typewriter keyboard. The table also gives the equivalent of the paper tape and typewriter keyboard codes in decimal and octal. Also listed are the console controls associated with input of data from the console and the binary coded decimal codes.

Display

36 DIS W

If the half-word bit of the address part of this instruction (W_{13}) contains a 0, the contents of the word in location W are displayed in command format on positions 1 through 15 of the display register as designated in Fig. 10. The sign of the left instruction is displayed on the appropriate neon in the section designated "S" in Fig. 10. The sign of the right-hand instruction is displayed at position 8, and appears as a 1 or 0, for + or -, respectively. The proper neon in Section S will indicate that the interpretation is in command format.

If the least significant bit of the address part of this instruction is a 1, the C(W) and C(W + 1) are displayed in decimal number format. From Fig. 10 it may be seen that the sign and only the first 32 binary positions of W and the first 28 binary positions of W + 1 are interpreted. This special configuration must be arranged, by programing, to conform to a maximum of 15 characters plus the sign (60 binary bits plus sign) output. Of these 60 binary positions 4 are used for each display position. This four-binary position code is known as a Binary Coded Decimal (BCD) format. In addition to the decimal characters 0, 1, 2, ... 9, three special display characters are also available. The decimal point is obtained by using the BCD code 12_{10}; the blank is obtained by using the BCD code 10_{10}; and the terminate code is indicated by the BCD code 15_{10}. The terminate code causes the display position for which it is coded and all other positions to the right to be made blank. Hence, this BCD code may be used to truncate the numbers displayed to any desired length. If the terminate code is preceded by 8 or less BCD characters plus the sign, the C(W + 1) are not examined, nor are any characters in W, which follow the terminate code. Note that the blank code and the decimal point code may be used in any position of the display.

The number N, used in the algebraic expression for the number of word times required by this instruction, refers to the number of positions of the display register which is used by the instruction (excluding the sign position). The C(R) are destroyed by the DIS instruction.

The "D" neon in section S indicates that the display format is decimal.

COPY COMMANDS

Copy to L Loop

64 CTL W

The contents of the 8 consecutive word locations in the L loop (7760.0 through 7767.0) are replaced by the 8 consecutive full words starting with the word in W. The fourth octal digit of W determines the word location in the L loop into which copying is begun. For example, if W = 3463.0 copying is started in 7763.0 and continues until the word at 3467.0 is copied into 7767.0. The word at 3470.0 is then copied into 7760.0, and copying continues until the remainder of the eight words has been copied into the L loop. The contents of the main memory are unaltered.

Copy to V Loop

66 CTV W

This instruction operates in the same manner as the Copy to L Loop command except that the contents of the 8 consecutive word locations of the V loop (7770.0 through 7777.0) are replaced by the 8 consecutive full words starting with the word in W. The contents of the main memory are unaltered.

Copy from L Loop

65 CFL W

The contents of the 8 consecutive word locations starting at W are replaced by the contents of the 8 consecutive full words in the L loop (7760.0 through 7767.0). The fourth octal digit of the word location at W determines the word location in the L loop from which copying is begun. For example, if W = 2752.0 copying is started from 7762.0 and continues until the word at 7767.0 is copied into 2757.0. The word at 7760.0 is then copied into 2760.0 and copying continues until the remainder of the eight words has been copied into main memory. The eight word locations in main memory must be in the same channel.

Copy from V Loop

67 CFV W

This instruction operates in the same manner as the Copy from L Loop command except that the contents of the eight consecutive word locations starting at W are replaced by the contents of the eight consecutive full words in the V loop (7770.0 through 7777.0). The eight word locations in the main memory must be in the same channel.

SYMBOLIC PROGRAMING

RECOMP II can execute a program only after instructions have been supplied in absolute or numerical codes. However, a programer usually writes a program in symbolic notation first. He must often refer to word locations that have not been assigned numerical addresses. In making storage assignments, he may find it difficult to estimate the number of storage locations necessary for the various sections of his program. Often coding errors are made in which the number of new instructions exceeds the original number of incorrect instructions. Therefore, the numbering of all locations in which this instruction sequence occurs must be modified, as well as all succeeding locations. In addition, the address part of all instructions which refer to modified locations must be changed. Therefore, some system of programing is needed which permits the use of nonabsolute or relative locations. One such system is called symbolic programing.

The program is first written in a symbolic form and the programer can make storage assignments of certain quantities. When such assignments are made, the symbolic quantities can be translated into machine language and an absolute program can be prepared.

APPENDIX A

GLOSSARY OF COMPUTER TERMINOLOGY

Access time - A time interval which is characteristic of a storage unit and is essentially a measure of the time required to communicate with that unit. The time interval between the instant at which information is: (a) called for from storage and the instant at which delivery is completed, i. e., the read time; or (b) ready for storage and the instant at which storage is completed, i. e., write time.

Accumulator - The main arithmetic register in which sums and other arithmetical and logical results are formed. In RECOMP II, this is known as the A register.

Address - A label which identifies a register, location, or device in which information is stored.

Absolute address - The number which specifies an absolute storage location in memory; that is, the number assigned as an address which is the same as that used by the machine when executing instructions.

Symbolic address - The description of a storage location by means of symbols, usually alphabetic letters, or alphabetic letters and numbers (eg, A2, AG7, etc).

Base (radix) - A quantity used explicitly to define some system of representing numbers by positional notation.

Binary - As in binary number system (base 2).

Binary-coded-decimal system - A system of number representation in which each decimal digit is represented by a group of four binary digits.

Bit - A binary digit.

Block - A group of words considered or handled as a unit: a record.

Breakpoint - A location in a routine at which the computer may, under any type of control (a manually set switch, a signal set into the code, etc), be stopped or made to produce information for a check of progress. This is a type of trapping.

Calling sequence - The group of instructions within a routine which includes the following:

Linkage - The instruction or instructions which transfer control to a subroutine and supply the necessary information for it to set up an exit (or return) to the appropriate place and to obtain error return, normal return, and parameters.

Error return - The location to which a subroutine exits if it has not succeeded in its function.

Check - A process of partial or complete testing of (1) the correctness of machine operations or (2) the correctness of results produced by a routine.

Built-in check - An automatic indication arising from the failure of the computer to verify a test built into the hardware of the computer (eg, echo check, overflow check, etc).

Clear - To fill a storage location or a register with zeros and, when not specifically stated otherwise, a plus sign. To replace information in a storage device by zero.

Code (verb) - To write a routine or part of a routine for a particular machine.

Coding (verb) - The act of writing computer instructions.

Operation code - That part of an instruction that designates the operation to be performed.

Code or coding (noun) - Symbolic or numerical equivalents of the instructions intelligible to a computer which are used in solving a problem.

Absolute, relative, or symbolic coding - Coding in which one uses absolute, relative, or symbolic data to represent operations and/or addresses.

Computer - A device capable of accepting information, applying prescribed processes to the information, and supplying the results of these processes.

Control sequence - The normal order of selection of instructions for execution. In RECOMP II this sequence is consecutive with the exception of transfer of control instructions.

Counter - A device, register, or storage location for storing integers, permitting these intergers to be increased or decreased by unity or by an arbitrary integer, and capable of being reset to zero or to an arbitrary integer.

Debug (verb) - To pinpoint and correct mistakes in a routine.

Diagnostic routine - A routine used to locate a malfunction in the computer.

Double length number - A number having twice as many digits as are ordinarily used in a given computer.

Dummy - An artificial address, instruction, or other unit of information inserted solely to fulfill prescribed conditions without affecting operations.

Dump (verb) - To copy the contents or a large part of the contents of one type of storage into another or to an output mechanism, usually for the purpose of providing a rerun point.

Extract - To erase or clear all bits within a word which do not compare with the contents of a specified storage location. Thus, the content of each bit position of the accumulator is compared with the content of the corresponding bit position of the storage location. If both bits are "1" the accumulator is left unaffected; otherwise a "0" replaces the contents of the corresponding bit position in the accumulator. See also "mask."

Error - The amount of loss of precision in a quantity; the difference between an accurate quantity and its calculated approximation.

File - A set of records.

File gap - An interval of space or time associated with a file to indicate or signal the end of the file.

Fixed point number - A number in which the point is explicitly located at some predetermined position.

Flag - Any device (usually a word) which conveys information from within a stored program computer to the routine being executed; this device may be set automatically by the computer or from the routine being executed: a storage device having a capacity of one binary digit.

Floating point number - A number which is represented by a pair of numbers F and E and a constant B which is an understood parameter in any given representation by the relation $F \cdot B^E$ = the original number. The quantity F is called the fractional (mantissa) part and E is called the exponent (characteristic) part of the floating point number. The exponent (characteristic) part E is an integer and ordinarily the fractional (mantissa) part is limited to the range: $0 \leq F < 1$. In RECOMP II, the fractional part is limited to the range $\pm (1/2 \leq F < 1)$ or $F=0$.

Flow chart - A graphic representation of the sequence of operations that comprises a routine.

Indicator - Any device which presents information from within a stored programed computer on the console of the computer. This device may be set automatically by the computer as the result of certain conditions or from the routine being executed.

Input (noun) - Information transferred from a secondary or external source (eg, paper tape) into the computer.

Instruction - A set of characters which defines an operation together with one or more addresses (or no address) and which, as a unit, causes the computer to operate accordingly on the indicated quantities.

Location - A position in memory which may be used to store information; a storage register, storage cell, or memory cell.

Loop - The repetition of a group of instructions in a routine.

Open loop - A loop which has an exit.

Closed (or endless loop) - A loop which has no exit (and which therefore, generally represents an error in coding).

Magnitude - The absolute value of a number, or the size of a quantity.

Mask - A control or screening word used with the extract instruction; an extractor.

Modify - To alter the address of the operand in an instruction.

Operand - The contents of the memory location specified by the address part of an instruction; one of the quantities involved in an arithmetic operation.

Output - Information transferred from the computer to an external medium.

Overflow (fixed point) - The generation of a quantity beyond the capacity of the register which is to receive the result.

Overflow (floating point) - The generation, in an arithmetic operation, of an exponent part (characteristic) whose magnitude exceeds the maximum allowable value.

Overflow indicator - An indicator which is turned on when an overflow is produced as the result of the execution of a command.

Program (noun) - A plan for the solution of a problem, including the coding for the problem. (It is usually desirable to include a flow chart and a numerical analysis in a complete program.)

Program (verb) - To set up a problem for solution by a computer.

Record (noun) - A sequential set of computer words or parts thereof; a block. See also file.

Record (verb) - To write information for the purpose of preserving for use at a later time.

Record gap - An interval of space or time associated with a unit record to indicate or signal the end of a record.

Register - The hardware for storing one or more computer words.

Restore (reset) - To return a device to zero or to an initial condition.

Routine - A set of coded instructions arranged in proper sequence to direct the computer to solve a given problem or perform certain calculations.

Run - The execution of a routine once in a computer, including input and output.

Run - The execution of a routine once in a computer, involving input and output.

Scale - A positive or negative integer which defines the position of the radix point (decimal, binary, etc) counted from some arbitrary reference position in the register which contains the number for which the scale is given. The scale is said to be positive if the radix point is to the right of the reference position, and negative if to the left. The symbol B is used for the binary scale and the symbol D is used for the decimal scale to change the scale in which a variable is expressed so as to bring it within the capacity of the machine or of the routine at hand.

Scale factor - A number required to transform the elements of mathematical equations so that the new elements will have numerical ranges compatible with the numerical ranges of the computing machine.

Scaling (optimum) - The scaling of a set of numerical quantities which yields a set of scales, the majority of which are minimum values. Thus, optimum scaling provides for the maximum degree of precision attainable for quantities of the given set.

Storage - The memory of a computer.

Volatile storage - A type of storage which does not retain information when the supply of electrical power to the computer is interrupted.

Nonvolatile storage - A type of storage which preserves information when the supply of electrical power to the computer is interrupted.

Storage location - See location.

Store (verb) - To record information into a storage location.

Subroutine - A set of instructions, forming part of a routine, which performs a well-defined mathematical or logical operation.

Closed subroutine - A subroutine which may be executed many times in a program, but whose instructions need appear only once in a program. To transfer control from the main routine to the subroutine, a set of instruction (known as a calling sequence) is given. The subroutine obtains sufficient information from this sequence to perform its various functions and to determine the return address to which control is transferred after completion of the subroutine.

Open subroutine - A subroutine which is incorporated in a program by direct insertion into the main routine. It is necessary to insert the subroutine at each point at which it is to be used.

Switch - Any manually set device which conveys information from the console of a computer to the routine stored within the computer.

Tracing - A technique for debugging a routine, whereby during its computation, information concerning the execution of commands (and/or the status, immediately following such execution, of registers, storage locations, indicators, etc) is transmitted to an output device.

Selective tracing - Tracing in which information is obtained on only a few of the instructions executed.

Tracing routine - A routine to provide automatic tracing.

Transfer (jump) - An instruction or signal which, conditionally or unconditionally, specifies the location of the next instruction and directs the computer to that instruction.

Trapping - A technique used in debugging to obtain information during the execution of a routine, whereby an instruction is inserted to transfer to a debugging routine. The trapping technique may also be used to transfer to an interpretive routine for execution of pseudo-operation codes. The location at which the instruction is inserted is usually called a breakpoint.

Word - A set of characters occupying one memory location. This is usually a number or an instruction.

Write - To record information in a register, location, or other storage device.

APPENDIX B

TABLE OF POWERS OF 2

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.062 5
32	5	0.031 25
64	6	0.015 625
128	7	0.007 812 5
256	8	0.003 906 25
512	9	0.001 953 125
1 024	10	0.000 976 562 5
2 048	11	0.000 488 281 25
4 096	12	0.000 244 140 625
8 192	13	0.000 122 070 312 5
16 384	14	0.000 061 035 156 25
32 768	15	0.000 030 517 578 125
65 536	16	0.000 015 258 789 062 5
131 072	17	0.000 007 629 394 531 25
262 144	18	0.000 003 814 697 265 625
524 288	19	0.000 001 907 348 632 812 5
1 048 576	20	0.000 000 953 674 316 406 25
2 097 152	21	0.000 000 476 837 158 203 125
4 194 304	22	0.000 000 238 418 579 101 562 5
8 388 608	23	0.000 000 119 209 289 550 781 25
16 777 216	24	0.000 000 059 604 644 775 390 625
33 554 432	25	0.000 000 029 802 322 387 695 312 5
67 108 864	26	0.000 000 014 901 161 193 847 656 25
134 217 728	27	0.000 000 007 450 580 596 923 828 125
268 435 456	28	0.000 000 003 725 290 298 461 914 062 5
536 870 912	29	0.000 000 001 862 645 149 230 957 031 25
1 073 741 824	30	0.000 000 000 931 322 574 615 478 515 625
2 147 483 648	31	0.000 000 000 465 661 287 307 739 257 812 5
4 294 967 296	32	0.000 000 000 232 830 643 653 869 628 906 25
8 589 934 592	33	0.000 000 000 116 415 321 826 934 814 453 125
17 179 869 184	34	0.000 000 000 058 207 660 913 467 407 226 562 5
34 359 738 368	35	0.000 000 000 029 103 830 456 733 703 613 281 25
68 719 476 736	36	0.000 000 000 014 551 915 228 366 851 806 640 625
137 438 953 472	37	0.000 000 000 007 275 957 614 183 425 903 320 312 5
274 877 906 944	38	0.000 000 000 003 637 978 807 091 712 951 660 156 25
549 755 813 888	39	0.000 000 000 001 818 989 403 545 856 475 830 078 125

OCTAL-DECIMAL INTEGER CONVERSION TABLE

0000 to 0777 (Octal) | 0000 to 0511 (Decimal)

	0	1	2	3	4	5	6	7
0000	0000	0001	0002	0003	0004	0005	0006	0007
0010	0008	0009	0010	0011	0012	0013	0014	0015
0020	0016	0017	0018	0019	0020	0021	0022	0023
0030	0024	0025	0026	0027	0028	0029	0030	0031
0040	0032	0033	0034	0035	0036	0037	0038	0039
0050	0040	0041	0042	0043	0044	0045	0046	0047
0060	0048	0049	0050	0051	0052	0053	0054	0055
0070	0056	0057	0058	0059	0060	0061	0062	0063
0100	0064	0065	0066	0067	0068	0069	0070	0071
0110	0072	0073	0074	0075	0076	0077	0078	0079
0120	0080	0081	0082	0083	0084	0085	0086	0087
0130	0088	0089	0090	0091	0092	0093	0094	0095
0140	0096	0097	0098	0099	0100	0101	0102	0103
0150	0104	0105	0106	0107	0108	0109	0110	0111
0160	0112	0113	0114	0115	0116	0117	0118	0119
0170	0120	0121	0122	0123	0124	0125	0126	0127
0200	0128	0129	0130	0131	0132	0133	0134	0135
0210	0136	0137	0138	0139	0140	0141	0142	0143
0220	0144	0145	0146	0147	0148	0149	0150	0151
0230	0152	0153	0154	0155	0156	0157	0158	0159
0240	0160	0161	0162	0163	0164	0165	0166	0167
0250	0168	0169	0170	0171	0172	0173	0174	0175
0260	0176	0177	0178	0179	0180	0181	0182	0183
0270	0184	0185	0186	0187	0188	0189	0190	0191
0300	0192	0193	0194	0195	0196	0197	0198	0199
0310	0200	0201	0202	0203	0204	0205	0206	0207
0320	0208	0209	0210	0211	0212	0213	0214	0215
0330	0216	0217	0218	0219	0220	0221	0222	0223
0340	0224	0225	0226	0227	0228	0229	0230	0231
0350	0232	0233	0234	0235	0236	0237	0238	0239
0360	0240	0241	0242	0243	0244	0245	0246	0247
0370	0248	0249	0250	0251	0252	0253	0254	0255

	0	1	2	3	4	5	6	7
0400	0256	0257	0258	0259	0260	0261	0262	0263
0410	0264	0265	0266	0267	0268	0269	0270	0271
0420	0272	0273	0274	0275	0276	0277	0278	0279
0430	0280	0281	0282	0283	0284	0285	0286	0287
0440	0288	0289	0290	0291	0292	0293	0294	0295
0450	0296	0297	0298	0299	0300	0301	0302	0303
0460	0304	0305	0306	0307	0308	0309	0310	0311
0470	0312	0313	0314	0315	0316	0317	0318	0319
0500	0320	0321	0322	0323	0324	0325	0326	0327
0510	0328	0329	0330	0331	0332	0333	0334	0335
0520	0336	0337	0338	0339	0340	0341	0342	0343
0530	0344	0345	0346	0347	0348	0349	0350	0351
0540	0352	0353	0354	0355	0356	0357	0358	0359
0550	0360	0361	0362	0363	0364	0365	0366	0367
0560	0368	0369	0370	0371	0372	0373	0374	0375
0570	0376	0377	0378	0379	0380	0381	0382	0383
0600	0384	0385	0386	0387	0388	0389	0390	0391
0610	0392	0393	0394	0395	0396	0397	0398	0399
0620	0400	0401	0402	0403	0404	0405	0406	0407
0630	0408	0409	0410	0411	0412	0413	0414	0415
0640	0416	0417	0418	0419	0420	0421	0422	0423
0650	0424	0425	0426	0427	0428	0429	0430	0431
0660	0432	0433	0434	0435	0436	0437	0438	0439
0670	0440	0441	0442	0443	0444	0445	0446	0447
0700	0448	0449	0450	0451	0452	0453	0454	0455
0710	0456	0457	0458	0459	0460	0461	0462	0463
0720	0464	0465	0466	0467	0468	0469	0470	0471
0730	0472	0473	0474	0475	0476	0477	0478	0479
0740	0480	0481	0482	0483	0484	0485	0486	0487
0750	0488	0489	0490	0491	0492	0493	0494	0495
0760	0496	0497	0498	0499	0500	0501	0502	0503
0770	0504	0505	0506	0507	0508	0509	0510	0511

1000 to 1777 (Octal) | 0512 to 1023 (Decimal)

	0	1	2	3	4	5	6	7
1000	0512	0513	0514	0515	0516	0517	0518	0519
1010	0520	0521	0522	0523	0524	0525	0526	0527
1020	0528	0529	0530	0531	0532	0533	0534	0535
1030	0536	0537	0538	0539	0540	0541	0542	0543
1040	0544	0545	0546	0547	0548	0549	0550	0551
1050	0552	0553	0554	0555	0556	0557	0558	0559
1060	0560	0561	0562	0563	0564	0565	0566	0567
1070	0568	0569	0570	0571	0572	0573	0574	0575
1100	0576	0577	0578	0579	0580	0581	0582	0583
1110	0584	0585	0586	0587	0588	0589	0590	0591
1120	0592	0593	0594	0595	0596	0597	0598	0599
1130	0600	0601	0602	0603	0604	0605	0606	0607
1140	0608	0609	0610	0611	0612	0613	0614	0615
1150	0616	0617	0618	0619	0620	0621	0622	0623
1160	0624	0625	0626	0627	0628	0629	0630	0631
1170	0632	0633	0634	0635	0636	0637	0638	0639
1200	0640	0641	0642	0643	0644	0645	0646	0647
1210	0648	0649	0650	0651	0652	0653	0654	0655
1220	0656	0657	0658	0659	0660	0661	0662	0663
1230	0664	0665	0666	0667	0668	0669	0670	0671
1240	0672	0673	0674	0675	0676	0677	0678	0679
1250	0680	0681	0682	0683	0684	0685	0686	0687
1260	0688	0689	0690	0691	0692	0693	0694	0695
1270	0696	0697	0698	0699	0700	0701	0702	0703
1300	0704	0705	0706	0707	0708	0709	0710	0711
1310	0712	0713	0714	0715	0716	0717	0718	0719
1320	0720	0721	0722	0723	0724	0725	0726	0727
1330	0728	0729	0730	0731	0732	0733	0734	0735
1340	0736	0737	0738	0739	0740	0741	0742	0743
1350	0744	0745	0746	0747	0748	0749	0750	0751
1360	0752	0753	0754	0755	0756	0757	0758	0759
1370	0760	0761	0762	0763	0764	0765	0766	0767

	0	1	2	3	4	5	6	7
1400	0768	0769	0770	0771	0772	0773	0774	0775
1410	0776	0777	0778	0779	0780	0781	0782	0783
1420	0784	0785	0786	0787	0788	0789	0790	0791
1430	0792	0793	0794	0795	0796	0797	0798	0799
1440	0800	0801	0802	0803	0804	0805	0806	0807
1450	0808	0809	0810	0811	0812	0813	0814	0815
1460	0816	0817	0818	0819	0820	0821	0822	0823
1470	0824	0825	0826	0827	0828	0829	0830	0831
1500	0832	0833	0834	0835	0836	0837	0838	0839
1510	0840	0841	0842	0843	0844	0845	0846	0847
1520	0848	0849	0850	0851	0852	0853	0854	0855
1530	0856	0857	0858	0859	0860	0861	0862	0863
1540	0864	0865	0866	0867	0868	0869	0870	0871
1550	0872	0873	0874	0875	0876	0877	0878	0879
1560	0880	0881	0882	0883	0884	0885	0886	0887
1570	0888	0889	0890	0891	0892	0893	0894	0895
1600	0896	0897	0898	0899	0900	0901	0902	0903
1610	0904	0905	0906	0907	0908	0909	0910	0911
1620	0912	0913	0914	0915	0916	0917	0918	0919
1630	0920	0921	0922	0923	0924	0925	0926	0927
1640	0928	0929	0930	0931	0932	0933	0934	0935
1650	0936	0937	0938	0939	0940	0941	0942	0943
1660	0944	0945	0946	0947	0948	0949	0950	0951
1670	0952	0953	0954	0955	0956	0957	0958	0959
1700	0960	0961	0962	0963	0964	0965	0966	0967
1710	0968	0969	0970	0971	0972	0973	0974	0975
1720	0976	0977	0978	0979	0980	0981	0982	0983
1730	0984	0985	0986	0987	0988	0989	0990	0991
1740	0992	0993	0994	0995	0996	0997	0998	0999
1750	1000	1001	1002	1003	1004	1005	1006	1007
1760	1008	1009	1010	1011	1012	1013	1014	1015
1770	1016	1017	1018	1019	1020	1021	1022	1023

1

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
2000	1024	1025	1026	1027	1028	1029	1030	1031
2010	1032	1033	1034	1035	1036	1037	1038	1039
2020	1040	1041	1042	1043	1044	1045	1046	1047
2030	1048	1049	1050	1051	1052	1053	1054	1055
2040	1056	1057	1058	1059	1060	1061	1062	1063
2050	1064	1065	1066	1067	1068	1069	1070	1071
2060	1072	1073	1074	1075	1076	1077	1078	1079
2070	1080	1081	1082	1083	1084	1085	1086	1087
2100	1088	1089	1090	1091	1092	1093	1094	1095
2110	1096	1097	1098	1099	1100	1101	1102	1103
2120	1104	1105	1106	1107	1108	1109	1110	1111
2130	1112	1113	1114	1115	1116	1117	1118	1119
2140	1120	1121	1122	1123	1124	1125	1126	1127
2150	1128	1129	1130	1131	1132	1133	1134	1135
2160	1136	1137	1138	1139	1140	1141	1142	1143
2170	1144	1145	1146	1147	1148	1149	1150	1151
2200	1152	1153	1154	1155	1156	1157	1158	1159
2210	1160	1161	1162	1163	1164	1165	1166	1167
2220	1168	1169	1170	1171	1172	1173	1174	1175
2230	1176	1177	1178	1179	1180	1181	1182	1183
2240	1184	1185	1186	1187	1188	1189	1190	1191
2250	1192	1193	1194	1195	1196	1197	1198	1199
2260	1200	1201	1202	1203	1204	1205	1206	1207
2270	1208	1209	1210	1211	1212	1213	1214	1215
2300	1216	1217	1218	1219	1220	1221	1222	1223
2310	1224	1225	1226	1227	1228	1229	1230	1231
2320	1232	1233	1234	1235	1236	1237	1238	1239
2330	1240	1241	1242	1243	1244	1245	1246	1247
2340	1248	1249	1250	1251	1252	1253	1254	1255
2350	1256	1257	1258	1259	1260	1261	1262	1263
2360	1264	1265	1266	1267	1268	1269	1270	1271
2370	1272	1273	1274	1275	1276	1277	1278	1279

	0	1	2	3	4	5	6	7
2400	1280	1281	1282	1283	1284	1285	1286	1287
2410	1288	1289	1290	1291	1292	1293	1294	1295
2420	1296	1297	1298	1299	1300	1301	1302	1303
2430	1304	1305	1306	1307	1308	1309	1310	1311
2440	1312	1313	1314	1315	1316	1317	1318	1319
2450	1320	1321	1322	1323	1324	1325	1326	1327
2460	1328	1329	1330	1331	1332	1333	1334	1335
2470	1336	1337	1338	1339	1340	1341	1342	1343
2500	1344	1345	1346	1347	1348	1349	1350	1351
2510	1352	1353	1354	1355	1356	1357	1358	1359
2520	1360	1361	1362	1363	1364	1365	1366	1367
2530	1368	1369	1370	1371	1372	1373	1374	1375
2540	1376	1377	1378	1379	1380	1381	1382	1383
2550	1384	1385	1386	1387	1388	1389	1390	1391
2560	1392	1393	1394	1395	1396	1397	1398	1399
2570	1400	1401	1402	1403	1404	1405	1406	1407
2600	1408	1409	1410	1411	1412	1413	1414	1415
2610	1416	1417	1418	1419	1420	1421	1422	1423
2620	1424	1425	1426	1427	1428	1429	1430	1431
2630	1432	1433	1434	1435	1436	1437	1438	1439
2640	1440	1441	1442	1443	1444	1445	1446	1447
2650	1448	1449	1450	1451	1452	1453	1454	1455
2660	1456	1457	1458	1459	1460	1461	1462	1463
2670	1464	1465	1466	1467	1468	1469	1470	1471
2700	1472	1473	1474	1475	1476	1477	1478	1479
2710	1480	1481	1482	1483	1484	1485	1486	1487
2720	1488	1489	1490	1491	1492	1493	1494	1495
2730	1496	1497	1498	1499	1500	1501	1502	1503
2740	1504	1505	1506	1507	1508	1509	1510	1511
2750	1512	1513	1514	1515	1516	1517	1518	1519
2760	1520	1521	1522	1523	1524	1525	1526	1527
2770	1528	1529	1530	1531	1532	1533	1534	1535

2000 to 2777 (Octal)	1024 to 1535 (Decimal)

	0	1	2	3	4	5	6	7
3000	1536	1537	1538	1539	1540	1541	1542	1543
3010	1544	1545	1546	1547	1548	1549	1550	1551
3020	1552	1553	1554	1555	1556	1557	1558	1559
3030	1560	1561	1562	1563	1564	1565	1566	1567
3040	1568	1569	1570	1571	1572	1573	1574	1575
3050	1576	1577	1578	1579	1580	1581	1582	1583
3060	1584	1585	1586	1587	1588	1589	1590	1591
3070	1592	1593	1594	1595	1596	1597	1598	1599
3100	1600	1601	1602	1603	1604	1605	1606	1607
3110	1608	1609	1610	1611	1612	1613	1614	1615
3120	1616	1617	1618	1619	1620	1621	1622	1623
3130	1624	1625	1626	1627	1628	1629	1630	1631
3140	1632	1633	1634	1635	1636	1637	1638	1639
3150	1640	1641	1642	1643	1644	1645	1646	1647
3160	1648	1649	1650	1651	1652	1653	1654	1655
3170	1656	1657	1658	1659	1660	1661	1662	1663
3200	1664	1665	1666	1667	1668	1669	1670	1671
3210	1672	1673	1674	1675	1676	1677	1678	1679
3220	1680	1681	1682	1683	1684	1685	1686	1687
3230	1688	1689	1690	1691	1692	1693	1694	1695
3240	1696	1697	1698	1699	1700	1701	1702	1703
3250	1704	1705	1706	1707	1708	1709	1710	1711
3260	1712	1713	1714	1715	1716	1717	1718	1719
3270	1720	1721	1722	1723	1724	1725	1726	1727
3300	1728	1729	1730	1731	1732	1733	1734	1735
3310	1736	1737	1738	1739	1740	1741	1742	1743
3320	1744	1745	1746	1747	1748	1749	1750	1751
3330	1752	1753	1754	1755	1756	1757	1758	1759
3340	1760	1761	1762	1763	1764	1765	1766	1767
3350	1768	1769	1770	1771	1772	1773	1774	1775
3360	1776	1777	1778	1779	1780	1781	1782	1783
3370	1784	1785	1786	1787	1788	1789	1790	1791

	0	1	2	3	4	5	6	7
3400	1792	1793	1794	1795	1796	1797	1798	1799
3410	1800	1801	1802	1803	1804	1805	1806	1807
3420	1808	1809	1810	1811	1812	1813	1814	1815
3430	1816	1817	1818	1819	1820	1821	1822	1823
3440	1824	1825	1826	1827	1828	1829	1830	1831
3450	1832	1833	1834	1835	1836	1837	1838	1839
3460	1840	1841	1842	1843	1844	1845	1846	1847
3470	1848	1849	1850	1851	1852	1853	1854	1855
3500	1856	1857	1858	1859	1860	1861	1862	1863
3510	1864	1865	1866	1867	1868	1869	1870	1871
3520	1872	1873	1874	1875	1876	1877	1878	1879
3530	1880	1881	1882	1883	1884	1885	1886	1887
3540	1888	1889	1890	1891	1892	1893	1894	1895
3550	1896	1897	1898	1899	1900	1901	1902	1903
3560	1904	1905	1906	1907	1908	1909	1910	1911
3570	1912	1913	1914	1915	1916	1917	1918	1919
3600	1920	1921	1922	1923	1924	1925	1926	1927
3610	1928	1929	1930	1931	1932	1933	1934	1935
3620	1936	1937	1938	1939	1940	1941	1942	1943
3630	1944	1945	1946	1947	1948	1949	1950	1951
3640	1952	1953	1954	1955	1956	1957	1958	1959
3650	1960	1961	1962	1963	1964	1965	1966	1967
3660	1968	1969	1970	1971	1972	1973	1974	1975
3670	1976	1977	1978	1979	1980	1981	1982	1983
3700	1984	1985	1986	1987	1988	1989	1990	1991
3710	1992	1993	1994	1995	1996	1997	1998	1999
3720	2000	2001	2002	2003	2004	2005	2006	2007
3730	2008	2009	2010	2011	2012	2013	2014	2015
3740	2016	2017	2018	2019	2020	2021	2022	2023
3750	2024	2025	2026	2027	2028	2029	2030	2031
3760	2032	2033	2034	2035	2036	2037	2038	2039
3770	2040	2041	2042	2043	2044	2045	2046	2047

3000 to 3777 (Octal)	1536 to 2047 (Decimal)

2

OCTAL-DECIMAL INTEGER CONVERSION TABLE

4000 to 4777 (Octal) | 2048 to 2559 (Decimal)

	0	1	2	3	4	5	6	7
4000	2048	2049	2050	2051	2052	2053	2054	2055
4010	2056	2057	2058	2059	2060	2061	2062	2063
4020	2064	2065	2066	2067	2068	2069	2070	2071
4030	2072	2073	2074	2075	2076	2077	2078	2079
4040	2080	2081	2082	2083	2084	2085	2086	2087
4050	2088	2089	2090	2091	2092	2093	2094	2095
4060	2096	2097	2098	2099	2100	2101	2102	2103
4070	2104	2105	2106	2107	2108	2109	2110	2111
4100	2112	2113	2114	2115	2116	2117	2118	2119
4110	2120	2121	2122	2123	2124	2125	2126	2127
4120	2128	2129	2130	2131	2132	2133	2134	2135
4130	2136	2137	2138	2139	2140	2141	2142	2143
4140	2144	2145	2146	2147	2148	2149	2150	2151
4150	2152	2153	2154	2155	2156	2157	2158	2159
4160	2160	2161	2162	2163	2164	2165	2166	2167
4170	2168	2169	2170	2171	2172	2173	2174	2175
4200	2176	2177	2178	2179	2180	2181	2182	2183
4210	2184	2185	2186	2187	2188	2189	2190	2191
4220	2192	2193	2194	2195	2196	2197	2198	2199
4230	2200	2201	2202	2203	2204	2205	2206	2207
4240	2208	2209	2210	2211	2212	2213	2214	2215
4250	2216	2217	2218	2219	2220	2221	2222	2223
4260	2224	2225	2226	2227	2228	2229	2230	2231
4270	2232	2233	2234	2235	2236	2237	2238	2239
4300	2240	2241	2242	2243	2244	2245	2246	2247
4310	2248	2249	2250	2251	2252	2253	2254	2255
4320	2256	2257	2258	2259	2260	2261	2262	2263
4330	2264	2265	2266	2267	2268	2269	2270	2271
4340	2272	2273	2274	2275	2276	2277	2278	2279
4350	2280	2281	2282	2283	2284	2285	2286	2287
4360	2288	2289	2290	2291	2292	2293	2294	2295
4370	2296	2297	2298	2299	2300	2301	2302	2303

	0	1	2	3	4	5	6	7
4400	2304	2305	2306	2307	2308	2309	2310	2311
4410	2312	2313	2314	2315	2316	2317	2318	2319
4420	2320	2321	2322	2323	2324	2325	2326	2327
4430	2328	2329	2330	2331	2332	2333	2334	2335
4440	2336	2337	2338	2339	2340	2341	2342	2343
4450	2344	2345	2346	2347	2348	2349	2350	2351
4460	2352	2353	2354	2355	2356	2357	2358	2359
4470	2360	2361	2362	2363	2364	2365	2366	2367
4500	2368	2369	2370	2371	2372	2373	2374	2375
4510	2376	2377	2378	2379	2380	2381	2382	2383
4520	2384	2385	2386	2387	2388	2389	2390	2391
4530	2392	2393	2394	2395	2396	2397	2398	2399
4540	2400	2401	2402	2403	2404	2405	2406	2407
4550	2408	2409	2410	2411	2412	2413	2414	2415
4560	2416	2417	2418	2419	2420	2421	2422	2423
4570	2424	2425	2426	2427	2428	2429	2430	2431
4600	2432	2433	2434	2435	2436	2437	2438	2439
4610	2440	2441	2442	2443	2444	2445	2446	2447
4620	2448	2449	2450	2451	2452	2453	2454	2455
4630	2456	2457	2458	2459	2460	2461	2462	2463
4640	2464	2465	2466	2467	2468	2469	2470	2471
4650	2472	2473	2474	2475	2476	2477	2478	2479
4660	2480	2481	2482	2483	2484	2485	2486	2487
4670	2488	2489	2490	2491	2492	2493	2494	2495
4700	2496	2497	2498	2499	2500	2501	2502	2503
4710	2504	2505	2506	2507	2508	2509	2510	2511
4720	2512	2513	2514	2515	2516	2517	2518	2519
4730	2520	2521	2522	2523	2524	2525	2526	2527
4740	2528	2529	2530	2531	2532	2533	2534	2535
4750	2536	2537	2538	2539	2540	2541	2542	2543
4760	2544	2545	2546	2547	2548	2549	2550	2551
4770	2552	2553	2554	2555	2556	2557	2558	2559

5000 to 5777 (Octal) | 2560 to 3071 (Decimal)

	0	1	2	3	4	5	6	7
5000	2560	2561	2562	2563	2564	2565	2566	2567
5010	2568	2569	2570	2571	2572	2573	2574	2575
5020	2576	2577	2578	2579	2580	2581	2582	2583
5030	2584	2585	2586	2587	2588	2589	2590	2591
5040	2592	2593	2594	2595	2596	2597	2598	2599
5050	2600	2601	2602	2603	2604	2605	2606	2607
5060	2608	2609	2610	2611	2612	2613	2614	2615
5070	2616	2617	2618	2619	2620	2621	2622	2623
5100	2624	2625	2626	2627	2628	2629	2630	2631
5110	2632	2633	2634	2635	2636	2637	2638	2639
5120	2640	2641	2642	2643	2644	2645	2646	2647
5130	2648	2649	2650	2651	2652	2653	2654	2655
5140	2656	2657	2658	2659	2660	2661	2662	2663
5150	2664	2665	2666	2667	2668	2669	2670	2671
5160	2672	2673	2674	2675	2676	2677	2678	2679
5170	2680	2681	2682	2683	2684	2685	2686	2687
5200	2688	2689	2690	2691	2692	2693	2694	2695
5210	2696	2697	2698	2699	2700	2701	2702	2703
5220	2704	2705	2706	2707	2708	2709	2710	2711
5230	2712	2713	2714	2715	2716	2717	2718	2719
5240	2720	2721	2722	2723	2724	2725	2726	2727
5250	2728	2729	2730	2731	2732	2733	2734	2735
5260	2736	2737	2738	2739	2740	2741	2742	2743
5270	2744	2745	2746	2747	2748	2749	2750	2751
5300	2752	2753	2754	2755	2756	2757	2758	2759
5310	2760	2761	2762	2763	2764	2765	2766	2767
5320	2768	2769	2770	2771	2772	2773	2774	2775
5330	2776	2777	2778	2779	2780	2781	2782	2783
5340	2784	2785	2786	2787	2788	2789	2790	2791
5350	2792	2793	2794	2795	2796	2797	2798	2799
5360	2800	2801	2802	2803	2804	2805	2806	2807
5370	2808	2809	2810	2811	2812	2813	2814	2815

	0	1	2	3	4	5	6	7
5400	2816	2817	2818	2819	2820	2821	2822	2823
5410	2824	2825	2826	2827	2828	2829	2830	2831
5420	2832	2833	2834	2835	2836	2837	2838	2839
5430	2840	2841	2842	2843	2844	2845	2846	2847
5440	2848	2849	2850	2851	2852	2853	2854	2855
5450	2856	2857	2858	2859	2860	2861	2862	2863
5460	2864	2865	2866	2867	2868	2869	2870	2871
5470	2872	2873	2874	2875	2876	2877	2878	2879
5500	2880	2881	2882	2883	2884	2885	2886	2887
5510	2888	2889	2890	2891	2892	2893	2894	2895
5520	2896	2897	2898	2899	2900	2901	2902	2903
5530	2904	2905	2906	2907	2908	2909	2910	2911
5540	2912	2913	2914	2915	2916	2917	2918	2919
5550	2920	2921	2922	2923	2924	2925	2926	2927
5560	2928	2929	2930	2931	2932	2933	2934	2935
5570	2936	2937	2938	2939	2940	2941	2942	2943
5600	2944	2945	2946	2947	2948	2949	2950	2951
5610	2952	2953	2954	2955	2956	2957	2958	2959
5620	2960	2961	2962	2963	2964	2965	2966	2967
5630	2968	2969	2970	2971	2972	2973	2974	2975
5640	2976	2977	2978	2979	2980	2981	2982	2983
5650	2984	2985	2986	2987	2988	2989	2990	2991
5660	2992	2993	2994	2995	2996	2997	2998	2999
5670	3000	3001	3002	3003	3004	3005	3006	3007
5700	3008	3009	3010	3011	3012	3013	3014	3015
5710	3016	3017	3018	3019	3020	3021	3022	3023
5720	3024	3025	3026	3027	3028	3029	3030	3031
5730	3032	3033	3034	3035	3036	3037	3038	3039
5740	3040	3041	3042	3043	3044	3045	3046	3047
5750	3048	3049	3050	3051	3052	3053	3054	3055
5760	3056	3057	3058	3059	3060	3061	3062	3063
5770	3064	3065	3066	3067	3068	3069	3070	3071

3

OCTAL-DECIMAL INTEGER CONVERSION TABLE

	0	1	2	3	4	5	6	7
6000	3072	3073	3074	3075	3076	3077	3078	3079
6010	3080	3081	3082	3083	3084	3085	3086	3087
6020	3088	3089	3090	3091	3092	3093	3094	3095
6030	3096	3097	3098	3099	3100	3101	3102	3103
6040	3104	3105	3106	3107	3108	3109	3110	3111
6050	3112	3113	3114	3115	3116	3117	3118	3119
6060	3120	3121	3122	3123	3124	3125	3126	3127
6070	3128	3129	3130	3131	3132	3133	3134	3135
6100	3136	3137	3138	3139	3140	3141	3142	3143
6110	3144	3145	3146	3147	3148	3149	3150	3151
6120	3152	3153	3154	3155	3156	3157	3158	3159
6130	3160	3161	3162	3163	3164	3165	3166	3167
6140	3168	3169	3170	3171	3172	3173	3174	3175
6150	3176	3177	3178	3179	3180	3181	3182	3183
6160	3184	3185	3186	3187	3188	3189	3190	3191
6170	3192	3193	3194	3195	3196	3197	3198	3199
6200	3200	3201	3202	3203	3204	3205	3206	3207
6210	3208	3209	3210	3211	3212	3213	3214	3215
6220	3216	3217	3218	3219	3220	3221	3222	3223
6230	3224	3225	3226	3227	3228	3229	3230	3231
6240	3232	3233	3234	3235	3236	3237	3238	3239
6250	3240	3241	3242	3243	3244	3245	3246	3247
6260	3248	3249	3250	3251	3252	3253	3254	3255
6270	3256	3257	3258	3259	3260	3261	3262	3263
6300	3264	3265	3266	3267	3268	3269	3270	3271
6310	3272	3273	3274	3275	3276	3277	3278	3279
6320	3280	3281	3282	3283	3284	3285	3286	3287
6330	3288	3289	3290	3291	3292	3293	3294	3295
6340	3296	3297	3298	3299	3300	3301	3302	3303
6350	3304	3305	3306	3307	3308	3309	3310	3311
6360	3312	3313	3314	3315	3316	3317	3318	3319
6370	3320	3321	3322	3323	3324	3325	3326	3327

	0	1	2	3	4	5	6	7
6400	3328	3329	3330	3331	3332	3333	3334	3335
6410	3336	3337	3338	3339	3340	3341	3342	3343
6420	3344	3345	3346	3347	3348	3349	3350	3351
6430	3352	3353	3354	3355	3356	3357	3358	3359
6440	3360	3361	3362	3363	3364	3365	3366	3367
6450	3368	3369	3370	3371	3372	3373	3374	3375
6460	3376	3377	3378	3379	3380	3381	3382	3383
6470	3384	3385	3386	3387	3388	3389	3390	3391
6500	3392	3393	3394	3395	3396	3397	3398	3399
6510	3400	3401	3402	3403	3404	3405	3406	3407
6520	3408	3409	3410	3411	3412	3413	3414	3415
6530	3416	3417	3418	3419	3420	3421	3422	3423
6540	3424	3425	3426	3427	3428	3429	3430	3431
6550	3432	3433	3434	3435	3436	3437	3438	3439
6560	3440	3441	3442	3443	3444	3445	3446	3447
6570	3448	3449	3450	3451	3452	3453	3454	3455
6600	3456	3457	3458	3459	3460	3461	3462	3463
6610	3464	3465	3466	3467	3468	3469	3470	3471
6620	3472	3473	3474	3475	3476	3477	3478	3479
6630	3480	3481	3482	3483	3484	3485	3486	3487
6640	3488	3489	3490	3491	3492	3493	3494	3495
6650	3496	3497	3498	3499	3500	3501	3502	3503
6660	3504	3505	3506	3507	3508	3509	3510	3511
6670	3512	3513	3514	3515	3516	3517	3518	3519
6700	3520	3521	3522	3523	3524	3525	3526	3527
6710	3528	3529	3530	3531	3532	3533	3534	3535
6720	3536	3537	3538	3539	3540	3541	3542	3543
6730	3544	3545	3546	3547	3548	3549	3550	3551
6740	3552	3553	3554	3555	3556	3557	3558	3559
6750	3560	3561	3562	3563	3564	3565	3566	3567
6760	3568	3569	3570	3571	3572	3573	3574	3575
6770	3576	3577	3578	3579	3580	3581	3582	3583

6000 to 6777 (Octal)	3072 to 3583 (Decimal)

	0	1	2	3	4	5	6	7
7000	3584	3585	3586	3587	3588	3589	3590	3591
7010	3592	3593	3594	3595	3596	3597	3598	3599
7020	3600	3601	3602	3603	3604	3605	3606	3607
7030	3608	3609	3610	3611	3612	3613	3614	3615
7040	3616	3617	3618	3619	3620	3621	3622	3623
7050	3624	3625	3626	3627	3628	3629	3630	3631
7060	3632	3633	3634	3635	3636	3637	3638	3639
7070	3640	3641	3642	3643	3644	3645	3646	3647
7100	3648	3649	3650	3651	3652	3653	3654	3655
7110	3656	3657	3658	3659	3660	3661	3662	3663
7120	3664	3665	3666	3667	3668	3669	3670	3671
7130	3672	3673	3674	3675	3676	3677	3678	3679
7140	3680	3681	3682	3683	3684	3685	3686	3687
7150	3688	3689	3690	3691	3692	3693	3694	3695
7160	3696	3697	3698	3699	3700	3701	3702	3703
7170	3704	3705	3706	3707	3708	3709	3710	3711
7200	3712	3713	3714	3715	3716	3717	3718	3719
7210	3720	3721	3722	3723	3724	3725	3726	3727
7220	3728	3729	3730	3731	3732	3733	3734	3735
7230	3736	3737	3738	3739	3740	3741	3742	3743
7240	3744	3745	3746	3747	3748	3749	3750	3751
7250	3752	3753	3754	3755	3756	3757	3758	3759
7260	3760	3761	3762	3763	3764	3765	3766	3767
7270	3768	3769	3770	3771	3772	3773	3774	3775
7300	3776	3777	3778	3779	3780	3781	3782	3783
7310	3784	3785	3786	3787	3788	3789	3790	3791
7320	3792	3793	3794	3795	3796	3797	3798	3799
7330	3800	3801	3802	3803	3804	3805	3806	3807
7340	3808	3809	3810	3811	3812	3813	3814	3815
7350	3816	3817	3818	3819	3820	3821	3822	3823
7360	3824	3825	3826	3827	3828	3829	3830	3831
7370	3832	3833	3834	3835	3836	3837	3838	3839

	0	1	2	3	4	5	6	7
7400	3840	3841	3842	3843	3844	3845	3846	3847
7410	3848	3849	3850	3851	3852	3853	3854	3855
7420	3856	3857	3858	3859	3860	3861	3862	3863
7430	3864	3865	3866	3867	3868	3869	3870	3871
7440	3872	3873	3874	3875	3876	3877	3878	3879
7450	3880	3881	3882	3883	3884	3885	3886	3887
7460	3888	3889	3890	3891	3892	3893	3894	3895
7470	3896	3897	3898	3899	3900	3901	3902	3903
7500	3904	3905	3906	3907	3908	3909	3910	3911
7510	3912	3913	3914	3915	3916	3917	3918	3919
7520	3920	3921	3922	3923	3924	3925	3926	3927
7530	3928	3929	3930	3931	3932	3933	3934	3935
7540	3936	3937	3938	3939	3940	3941	3942	3943
7550	3944	3945	3946	3947	3948	3949	3950	3951
7560	3952	3953	3954	3955	3956	3957	3958	3959
7570	3960	3961	3962	3963	3964	3965	3966	3967
7600	3968	3969	3970	3971	3972	3973	3974	3975
7610	3976	3977	3978	3979	3980	3981	3982	3983
7620	3984	3985	3986	3987	3988	3989	3990	3991
7630	3992	3993	3994	3995	3996	3997	3998	3999
7640	4000	4001	4002	4003	4004	4005	4006	4007
7650	4008	4009	4010	4011	4012	4013	4014	4015
7660	4016	4017	4018	4019	4020	4021	4022	4023
7670	4024	4025	4026	4027	4028	4029	4030	4031
7700	4032	4033	4034	4035	4036	4037	4038	4039
7710	4040	4041	4042	4043	4044	4045	4046	4047
7720	4048	4049	4050	4051	4052	4053	4054	4055
7730	4056	4057	4058	4059	4060	4061	4062	4063
7740	4064	4065	4066	4067	4068	4069	4070	4071
7750	4072	4073	4074	4075	4076	4077	4078	4079
7760	4080	4081	4082	4083	4084	4085	4086	4087
7770	4088	4089	4090	4091	4092	4093	4094	4095

7000 to 7777 (Octal)	3584 to 4095 (Decimal)

4

APPENDIX D. OCTAL-DECIMAL FRACTION CONVERSION TABLE

OCTAL	DEC.	OCTAL	DEC.
0.000	0.000000000	0.061	0.095703125
0.001	0.001953125	0.062	0.097656250
0.002	0.003906250	0.063	0.099609375
0.003	0.005859375	0.064	0.101562500
0.004	0.007812500	0.065	0.103515625
0.005	0.009765625	0.066	0.105468750
0.006	0.011718750	0.067	0.107421875
0.007	0.013671875	0.070	0.109375000
0.010	0.015625000	0.071	0.111328125
0.011	0.017578125	0.072	0.113281250
0.012	0.019531250	0.073	0.115234375
0.013	0.021484375	0.074	0.117187500
0.014	0.023437500	0.075	0.119140625
0.015	0.025390625	0.076	0.121093750
0.016	0.027343750	0.077	0.123046875
0.017	0.029296875	0.100	0.125000000
0.020	0.031250000	0.101	0.126953125
0.021	0.033203125	0.102	0.128906250
0.022	0.035156250	0.103	0.130859375
0.023	0.037109375	0.104	0.132812500
0.024	0.039062500	0.105	0.134765625
0.025	0.041015625	0.106	0.136718750
0.026	0.042968750	0.107	0.138671875
0.027	0.044921875	0.110	0.140625000
0.030	0.046875000	0.111	0.142578125
0.031	0.048828125	0.112	0.144531250
0.032	0.050781250	0.113	0.146484375
0.033	0.052734375	0.114	0.148437500
0.034	0.054687500	0.115	0.150390625
0.035	0.056640625	0.116	0.152343750
0.036	0.058593750	0.117	0.154296875
0.037	0.060546875	0.120	0.156250000
0.040	0.062500000	0.121	0.158203125
0.041	0.064453125	0.122	0.160156250
0.042	0.066406250	0.123	0.162109375
0.043	0.068359375	0.124	0.164062500
0.044	0.070312500	0.125	0.166015625
0.045	0.072265625	0.126	0.167968750
0.046	0.074218750	0.127	0.169921875
0.047	0.076171875	0.130	0.171875000
0.050	0.078125000	0.131	0.173828125
0.051	0.080078125	0.132	0.175781250
0.052	0.082031250	0.133	0.177734375
0.053	0.083984375	0.134	0.179687500
0.054	0.085937500	0.135	0.181640625
0.055	0.087890625	0.136	0.183593750
0.056	0.089843750	0.137	0.185546875
0.057	0.091796875	0.140	0.187500000
0.060	0.093750000	0.141	0.189453125

APPENDIX D. OCTAL-DECIMAL FRACTION CONVERSION TABLE (CONT)

OCTAL	DEC.	OCTAL	DEC.
0.142	0.191406250	0.220	0.281250000
0.143	0.193359375	0.221	0.283203125
0.144	0.195312500	0.222	0.285156250
0.145	0.197265625	0.223	0.287109375
0.146	0.199218750	0.224	0.289062500
0.147	0.201171875	0.225	0.291015625
0.150	0.203125000	0.226	0.292968750
0.151	0.205078125	0.227	0.294921875
0.152	0.207031250	0.230	0.296875000
0.153	0.208984375	0.231	0.298828125
0.154	0.210937500	0.232	0.300781250
0.155	0.212890625	0.233	0.302734375
0.156	0.214843750	0.234	0.304687500
0.157	0.216796875	0.235	0.306640625
0.160	0.218750000	0.236	0.308593750
0.161	0.220703125	0.237	0.310546875
0.162	0.222656250	0.240	0.312500000
0.163	0.224609375	0.241	0.314453125
0.164	0.226562500	0.242	0.316406250
0.165	0.228515625	0.243	0.318359375
0.166	0.230468750	0.244	0.320312500
0.167	0.232421875	0.245	0.322265625
0.170	0.234375000	0.246	0.324218750
0.171	0.236328125	0.247	0.326171875
0.172	0.238281250	0.250	0.328125000
0.173	0.240234375	0.251	0.330078125
0.174	0.242187500	0.252	0.332031250
0.175	0.244140625	0.253	0.333984375
0.176	0.246093750	0.254	0.335937500
0.177	0.248046875	0.255	0.337890625
0.200	0.250000000	0.256	0.339843750
0.201	0.251953125	0.257	0.341796875
0.202	0.253906250	0.260	0.343750000
0.203	0.255859375	0.261	0.345703125
0.204	0.257812500	0.262	0.347656250
0.205	0.259765625	0.263	0.349609375
0.206	0.261718750	0.264	0.351562500
0.207	0.263671875	0.265	0.353515625
0.210	0.265625000	0.266	0.355468750
0.211	0.267578125	0.267	0.357421875
0.212	0.269531250	0.270	0.359375000
0.213	0.271484375	0.271	0.361328125
0.214	0.273437500	0.272	0.363281250
0.215	0.275390625	0.273	0.365234375
0.216	0.277343750	0.274	0.367187500
0.217	0.279296875	0.275	0.369140625

OCTAL	DEC.	OCTAL	DEC.
0.276	0.371093750	0.354	0.460937500
0.277	0.373046875	0.355	0.462890625
0.300	0.375000000	0.356	0.464843750
0.301	0.376953125	0.357	0.466796875
0.302	0.378906250	0.360	0.468750000
0.303	0.380859375	0.361	0.470703125
0.304	0.382812500	0.362	0.472656250
0.305	0.384765625	0.363	0.474609375
0.306	0.386718750	0.364	0.476562500
0.307	0.388671875	0.365	0.478515625
0.310	0.390625000	0.366	0.480468750
0.311	0.392578125	0.367	0.482421875
0.312	0.394531250	0.370	0.484375000
0.313	0.396484375	0.371	0.486328125
0.314	0.398437500	0.372	0.488281250
0.315	0.400390625	0.373	0.490234375
0.316	0.402343750	0.374	0.492187500
0.317	0.404296875	0.375	0.494140625
0.320	0.406250000	0.376	0.496093750
0.321	0.408203125	0.377	0.498046875
0.322	0.410156250	0.400	0.500000000
0.323	0.412109375	0.401	0.501953125
0.324	0.414062500	0.402	0.503906250
0.325	0.416015625	0.403	0.505859375
0.326	0.417968750	0.404	0.507812500
0.327	0.419921875	0.405	0.509765625
0.330	0.421875000	0.406	0.511718750
0.331	0.423828125	0.407	0.513671875
0.332	0.425781250	0.410	0.515625000
0.333	0.427734375	0.411	0.517578125
0.334	0.429687500	0.412	0.519531250
0.335	0.431640625	0.413	0.521484375
0.336	0.433593750	0.414	0.523437500
0.337	0.435546875	0.415	0.525390625
0.340	0.437500000	0.416	0.527343750
0.341	0.439453125	0.417	0.529296875
0.342	0.441406250	0.420	0.531250000
0.343	0.443359375	0.421	0.533203125
0.344	0.445312500	0.422	0.535156250
0.345	0.447265625	0.423	0.537109375
0.346	0.449218750	0.424	0.539062500
0.347	0.451171875	0.425	0.541015625
0.350	0.453125000	0.426	0.542968750
0.351	0.455078125	0.427	0.544921875
0.352	0.457031250	0.430	0.546875000
0.353	0.458984375	0.431	0.548828125

OCTAL	DEC.
0.432	0.550781250
0.433	0.552734375
0.434	0.554687500
0.435	0.556640625
0.436	0.558593750
0.437	0.560546875
0.440	0.562500000
0.441	0.564453125
0.442	0.566406250
0.443	0.568359375
0.444	0.570312500
0.445	0.572265625
0.446	0.574218750
0.447	0.576171875
0.450	0.578125000
0.451	0.580078125
0.452	0.582031250
0.453	0.583984375
0.454	0.585937500
0.455	0.587890625
0.456	0.589843750
0.457	0.591796875
0.460	0.593750000
0.461	0.595703125
0.462	0.597656250
0.463	0.599609375
0.464	0.501562500
0.465	0.603515625
0.466	0.605468750
0.467	0.607421875
0.470	0.609375000
0.471	0.611328125
0.472	0.613281250
0.473	0.615234375
0.474	0.617187500
0.475	0.619140625
0.476	0.621093750
0.477	0.623046875
0.500	0.625000000
0.501	0.626953125
0.502	0.628906250
0.503	0.630859375
0.504	0.632812500
0.505	0.634765625
0.506	0.636718750
0.507	0.638671875
0.510	0.640625000
0.511	0.642578125
0.512	0.644531250
0.513	0.646484375
0.514	0.648437500
0.515	0.650390625
0.516	0.652343750
0.517	0.654296875
0.520	0.656250000
0.521	0.658203125
0.522	0.660156250
0.523	0.662109375
0.524	0.664062500
0.525	0.666015625
0.526	0.667968750
0.527	0.669921875
0.530	0.671875000
0.531	0.673828125
0.532	0.675781250
0.533	0.677734375
0.534	0.679687500
0.535	0.681640625
0.536	0.683593750
0.537	0.685546875
0.540	0.687500000
0.541	0.689453125
0.542	0.691406250
0.543	0.693359375
0.544	0.695312500
0.545	0.697265625
0.546	0.699218750
0.547	0.701171875
0.550	0.703125000
0.551	0.705078125
0.552	0.707031250
0.553	0.708984375
0.554	0.710937500
0.555	0.712890625
0.556	0.714843750
0.557	0.716796875
0.560	0.718750000
0.561	0.720703125
0.562	0.722656250
0.563	0.724609375
0.564	0.726562500
0.565	0.728515625

OCTAL	DEC	OCTAL	DEC.
0.566	0.730468750	0.645	0.822265625
0.567	0.732421875	0.646	0.824218750
0.570	0.734375000	0.647	0.826171875
0.571	0.736328125	0.650	0.828125000
0.572	0.738281250	0.651	0.830078125
0.573	0.740234375	0.652	0.832031250
0.574	0.742187500	0.653	0.833984375
0.575	0.744140625	0.654	0.835937500
0.576	0.746093750	0.655	0.837890625
0.577	0.748046875	0.656	0.839843750
0.600	0.750000000	0.657	0.841796875
0.601	0.751953125	0.660	0.843750000
0.602	0.753906250	0.661	0.845703125
0.603	0.755859375	0.662	0.847656250
0.604	0.757812500	0.663	0.849609375
0.605	0.759765625	0.664	0.851562500
0.606	0.761718750	0.665	0.853515625
0.607	0.763671875	0.666	0.855468750
0.610	0.765625000	0.667	0.857421875
0.611	0.767578125	0.670	0.859375000
0.612	0.769531250	0.671	0.861328125
0.613	0.771484375	0.672	0.863281250
0.614	0.773437500	0.673	0.865234375
0.615	0.775390625	0.674	0.867187500
0.616	0.777343750	0.675	0.869140625
0.617	0.779296875	0.676	0.871093750
0.620	0.781250000	0.677	0.873046875
0.621	0.783203125	0.700	0.875000000
0.622	0.785156250	0.701	0.876953125
0.623	0.787109375	0.702	0.878906250
0.624	0.789062500	0.703	0.880859375
0.625	0.791015625	0.704	0.882812500
0.626	0.792968750	0.705	0.884765625
0.627	0.794921875	0.706	0.886718750
0.630	0.796875000	0.707	0.888671875
0.631	0.798828125	0.710	0.890625000
0.632	0.800781250	0.711	0.892578125
0.633	0.802734375	0.712	0.894531250
0.634	0.804687500	0.713	0.896484375
0.635	0.806640625	0.714	0.898437500
0.636	0.808593750	0.715	0.900390625
0.637	0.810546875	0.716	0.902343750
0.640	0.812500000	0.717	0.904296875
0.641	0.814453125	0.720	0.906250000
0.642	0.816406250	0.721	0.908203125
0.643	0.818359375	0.722	0.910156250
0.644	0.820312500	0.723	0.912109375

APPENDIX D. OCTAL-DECIMAL FRACTION CONVERSION TABLE (CONT)

OCTAL	DEC.	OCTAL	DEC.
0.724	0.914062500	0.752	0.957031250
0.725	0.916015625	0.753	0.958984375
0.726	0.917968750	0.754	0.960937500
0.727	0.919921875	0.755	0.962890625
0.730	0.921875000	0.756	0.964843750
0.731	0.923828125	0.757	0.966796875
0.732	0.925781250	0.760	0.868750000
0.733	0.927734375	0.761	0.970703125
0.734	0.929687500	0.762	0.972656250
0.735	0.931640625	0.763	0.974609375
0.736	0.933593750	0.764	0.976562500
0.737	0.935546875	0.765	0.978515625
0.740	0.937500000	0.766	0.980468750
0.741	0.939453125	0.767	0.982421875
0.742	0.941406250	0.770	0.984375000
0.743	0.943359375	0.771	0.986328125
0.744	0.945312500	0.772	0.988281250
0.745	0.947265625	0.773	0.990234375
0.746	0.949218750	0.774	0.992187500
0.747	0.951171875	0.775	0.994140625
0.750	0.953125000	0.776	0.996093750
0.751	0.955078125	0.777	0.998046875

APPENDIX E
OPERATIONS BY ALPHABETIC CODES

T = Access Time (See Computer Operation Time Factors, pages 22 and 23).
n = (1) in Shift commands the number of bit positions through which the number in the accumulator is shifted (2) in Display command the number of characters displayed (3) in Floating Add, Floating Subtract, Floating Multiply, Floating Divide, and Floating Normalize commands the number of shifts required to normalize the fraction.
d = difference in size of exponents; in Floating Square Root, d represents the additional one word time required when exponent is odd.

Alpha code	Octal code	Word times*	Time (ms)**	Operation	Page
ADD	01	2+T	0.54 + T	Add	64
ALS	41	2+1n	0.54 + 0.27n	Accumulator Left Shift	68
ARS	40	2+1n	0.54 + 0.27n	Accumulator Right Shift	68
CFL	65	7+T	1.89 + T	Copy from L Loop	82
CFV	67	7+T	1.89 + T	Copy from V Loop	83
CLA	00	1+T	0.27 + T	Clear and Add	64
CLS	02	1+T	0.27 + T	Clear and Subtract	64
CTL	64	7+T	1.89 + T	Copy to L Loop	82
CTV	66	7+T	1.89 + T	Copy to V Loop	82
DIS	36	Approx. 2+2n+T	Approx. 0.54 + 0.54n + T	Display	81
DIV	22	41+T	11.1 + T	Divide	65
DSL	20	41+T	11.1 + T	Divide Single Length	66
DSR	21	42+T	11.3 + T	Divide Single Length & Round	66
DVR	23	42+T	11.3 + T	Divide and Round	66
EXT	33	1+T	0.27 + T	Extract	69
FAD	04	5+1d+1n +T	1.35 + 0.27d + 0.27n + T	Floating Add	71
FCA	30	1+T	0.27 + T	Floating Clear and Add	70
FCS	34	1+T	0.27 + T	Floating Clear and Subtract	71
FDV	05	47+1n+T	12.7 + 0.27n + T	Floating Divide	73
FMP	07	46+1n+T	12.4 + 0.27n + T	Floating Multiply	72
FNM	45	2+1n	0.54 + 0.27n	Floating Normalize	74
FSB	06	5+1d+1n +T	1.35 + 0.27d + 0.27n + T	Floating Subtract	72
FSQ	44	43+1d+T	11.6 + 0.27d + T	Floating Square Root	74
FST	35	1+T	0.27 + T	Floating Store	75
HTR	77	3	0.81	Halt and Transfer	75
MPR	13	40+T	10.8 + T	Multiply and Round	65
MPY	11	40+T	10.8 + T	Multiply	65
PNC	74	--	Approx 20 Char/Sec	Punch Character	77
PNW	14	--	Approx 22 Char/Sec	Punch Word	78
PTC	76	--	Approx 10 Char/Sec	Punch and Type Character	78

*One word time = approximately 0.27 ms
**Milliseconds

Alpha code	Octal code	Word times	Time (ms)	Operation	Page
PTW	16	--	Approx 10 Char/Sec	Punch and Type Word	79
RDY	71	--	Approx 2.5	Read Y	79
RDZ	73	--	Approx 2.5	Read Z	80
SAX	15	1+T	0.27 + T	Store A and Exchange A and X	69
SQR	25	42+T	11.3 + T	Square Root	67
STA	42	32+T or 4+T or 8+T	8.64 + T or 1.08 + T or 2.16 + T	Store Address	67
STO	60	1+T	0.27 + T	Store	67
SUB	03	2+T	0.54 + T	Subtract	64
TMI	51	3	0.81	Transfer on Minus	76
TOV	53	3	0.81	Transfer on Overflow	76
TPL	52	3	0.81	Transfer on Plus	75
TRA	57	3	0.81	Transfer	76
TSB	54	3 or 2	0.81 or 0.54	Transfer on Sense Switch B	76
TSC	55	3 or 2	0.81 or 0.54	Transfer on Sense Switch C	77
TSD	56	3 or 2	0.81 or 0.54	Transfer on Sense Switch D	77
TYC	72	--	Approx 10 Char/Sec	Type Character	77
TYW	12	--	Approx 10 Char/Sec	Type Word	79
TZE	50	3	0.81	Transfer on Zero	75
XAR	43	2	0.54	Exchange A and R	69

RECOMP II CODING SHEET

ASD-17

JOB NO. ______ DATE ______

TITLE ______ ANALYST ______ PAGE ____ OF ____

ABSOLUTE						SYMBOLIC				ASAP		REMARKS
LOCATION	H	S	OP	ADDRESS	H	LOCATION	OP	T	ADDRESS	INCREMENT	L	
7700							CTL		7701			
							TR		7761			
1							FCA		A			
							FSB		+1			
2							FMP		N			
							FAD		B			
3							FAD		+65535			
							ADD		KCov			
4							STA		Tag			Cov(A,B)
							STA		Tag			
5							FCA		A			
							FAD		+65535			
6							ADD		K Mean			
							STA		Tag			Mean(A)
7							CTL		7710			
							TR		7760			
7710							FCA		B			
							FAD		+65535			
1							ADD		K Mean			
							STA		Tag			Mean(B)
2							FCA		C			
							FSB		+1			
3							FAD		A			
							FAD		+65535			
4							ADD		K Data			
							STA		Tag			Data(C,A)
5							FCA		C			
							FSB		+1			
6							FMP		N			
							FAD		B			
7							CTL		7720			
							TR		7760			

176 at B39

177

No. of Vectors = 4

No of elements/vector = 3

	200	202	204
V_1			
A_1	206	210	212
A_2	214	216	220
A_3	222	224	226
A_4	230	232	234

200 + 2N

+ 2N

+ 2N

+ 2N

Relative

0000	CAD	0176	
	STO	0020	N
1	CTL	0010	
	TR	7760	
2			
3			
4			
5			
6			
7			
10			
11			
12			