RICE UNIVERSITY COMPUTER

BASIC MACHINE OPERATION

JANUARY, 1962

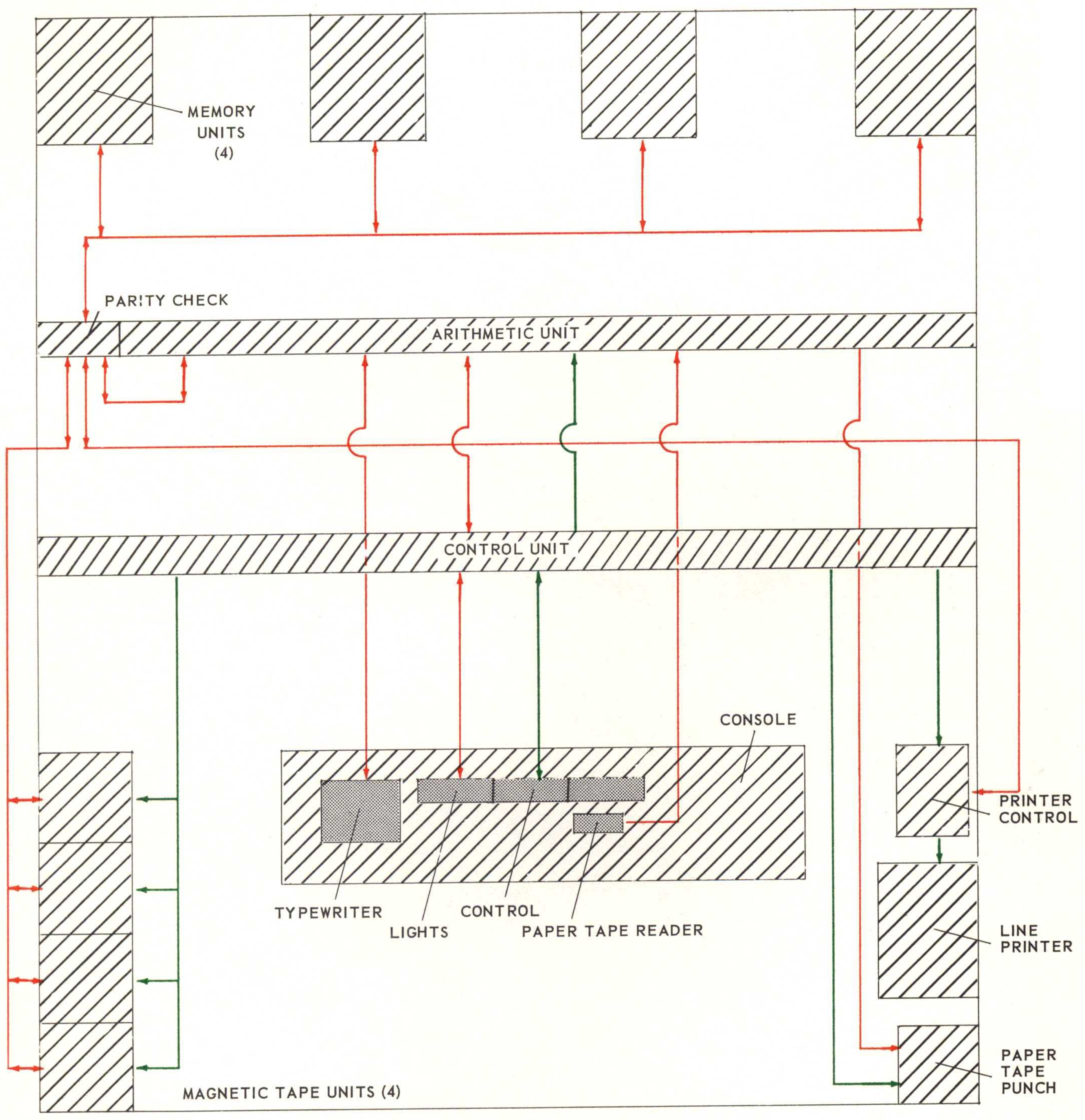
RICE UNIVERSITY COMPUTER

BASIC MACHINE OPERATION

January, 1962

The computer described in this publication was constructed under UNITED STATES ATOMIC ENERGY COMMISSION CONTRACT AT-(40-1)-1825.

TABLE OF CONTENTS



RICE COMPUTER
FLOOR PLAN AND INFORMATION FLOW

SCALE: 1" = APPROX. 3'

🟥 DATA
🟩 CONTROL

FIG. 1

I.

COMPUTER ORGANIZATION

The modern digital computer consists of five distinct groups of equipment which perform the following functions:

(1) input

(2) memory or storage

(3) arithmetic

(4) control

(5) output

The <u>input section</u> consists of a photoelectric reader which reads information from punched paper tape and stores it in memory and an electric typewriter which can be used to type information into the arithmetic and control sections. The arithmetic unit is always an intermediate in the flow of input information to memory. The information in question may be anything which can be stored in memory: numbers, instructions or alphabetical and numerical comments.

The <u>memory</u> is an information-holding device composed of electrostatic storage tubes. One memory contains 56 storage tubes and is subdivided into distinct units called <u>words</u>. The memory is needed to record numbers and hold instructions. Thus, each word may be a number, an instruction or a coded comment. Each memory unit is capable of recording up to 8,192 words, and the computer in its final form will have 4 memory units.

The memory may be thought of as N boxes or locations where numbers or instructions can be located. Each of the locations is given an identification number from 8 to N (the numbers 0 to 7 are reserved for a purpose to be explained later). The label of a location is called its address (synonyms: cell, location, box). Note that the address 1371 does not mean that we can find the number 1371 stored there - except by accident; the address is purely a label or identifying number.

A memory location can hold only one word at a time, and placing a word in a location automatically destroys whatever was there previously. It is possible to read a number out of memory without destroying or removing it.

The arithmetic section does what its name implies. In addition to the basic arithmetic operations, this unit can shift numbers right and left and assist in certain operations which make it possible for the computer to make decisions. If we use the analogy of a desk calculator, this section corresponds to the upper, lower and middle dials plus the wheels and gears that actually do the calculation.

"Register" is a term commonly used in connection with these various units. It denotes a device for temporarily storing a piece of information while or until it is used. A register corresponds quite closely to the dials on a desk calculator. Not only numbers but also instructions may be stored in a register.

The whole computer is controlled by a certain set of specified permissible operations, and no two such operations can occur simultaneously. The permissible operations may be executed in any desired sequence. It is up to the user to specify the sequence of operations or, as it is commonly called, the program. Each permissible operation can be specified in a concise coded form called an order (synonym: instruction). In order to be solved on a computer, a problem must be broken down into a series of precise steps and stored in memory as code numbers. The correspondence between the set of permissible operations and the set of numbers which specify them is called the order code and is described in the section on instructions.

The _control section_ of the machine has the function of accepting orders one by one and of interpreting or decoding these instructions and then sending signals to the other units telling them what to do. The control unit is equivalent to the operation buttons which are pushed on a desk calculator. The control section is described in detail in another section.

The _output units_ are an automatic punch for paper tape and a fast line printer. The printer can print up to 600 lines per minute - each line containing up to 108 characters. Information may also be permanently recorded (or written) on magnetic tape.

<u>SUMMARY</u> of <u>MACHINE CHARACTERISTICS</u>:

The Rice University Computer is a megacycle computer (i.e., a basic pulse time of about 1 microsecond) with a speed that is appropriate to:

(1) memory access time for reading of 10 microseconds

(2) memory access time for writing of 20 microseconds

(3) an addition time of 4 microseconds

(4) an average multiplication time of 120 microseconds.

The machine is asynchronous, binary and parallel in operation and will have a random access memory of 32,000 words.

II.

OCTAL NOTATION

Binary numbers are very well adapted to representation by electronic circuits. Since each digit can have only two different values, zero or one, the digits of a binary number can be put into one-to-one correspondence with the electrical conditions of off-on, open-closed, non-conducting-conducting, etc. We pay for this simplicity (i.e., small amount of information per digit) by needing more digits to represent a given amount of information than if we had used a larger number base. For example, a decimal number with N significant figures is equivalent to a binary number with $N \ln 10/\ln 2 = N/0.30103 = 3.321 N$ digits. The standard numerical word in the Rice Computer has from 40 to 47 significant binary places. This is equivalent to about 12 to 14 decimal places.

The problem of conversion between base two and base ten is actually simple but need not concern the reader at the moment. The process will be carried out essentially automatically by the computer by means of subroutines, so that the average machine user will supply decimal input data and the computer will deliver decimal final results.

In order to discuss the instruction word and numerical word structure of the computer, we must use the full

54 bit binary machine words. It is very inconvenient to write out such words in full and it is equally inconvenient to type them into a typewriter-tape punch. As a shorthand, we use "octal" notation. The binary number is divided into triads (groups of three bits). Instead of writing each triad in full, we write instead an integer between zero and seven inclusive:

binary	octal
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

Each triad is thought of as an octal integer, and the digit written is the usual symbol for this integer. The reader is advised to memorize this conversion table. This conversion is of course very easy in either direction. The resulting shorthand number is actually the equivalent of the binary number written to base eight, i.e., an octal number. A 54 bit machine word becomes an 18 octal digit number, much more manageable in length. We shall use expressions such as "the second octal figure" and "the second triad" essentially synonymously. In the computer we have triads; on paper or at the flexowriter we shall use octal figures.

As an example, 000101011001010100111 is equivalent to 000, 101, 011, 001, 010, 100, 111, or the octal number

0531247. The octal form is obviously much easier to write and to absorb at a glance.

In referring to an octal or binary number we read it from left to right. For example, "the first octal figure" refers to the figure furthest to the left (0 in the above example); "the second octal figure" or "the second triad" in the number above is 5.

III.

NUMERICAL WORD STRUCTURE

The standard word of 54 bits is divided into two numerical fields for arithmetic operations. Bits 1 through 6 represent an exponent E, which is an integer in the range $[-31,+31]$. Bits 7 through 54 represent a mantissa M, which in floating point work is taken to be a fraction with magnitude in the range $[0,1-2^{-47}]$.

In each field, the first bit represents the sign of the number. If the sign is positive, it is represented by a "0", and the remaining bits in the field give the magnitude of the number in binary form. If the sign is negative, it is represented by a "1", and the remaining bits in the field must then be complemented to obtain the magnitude of the number in binary form. This is the so-called 'one's complement' representation of numbers, which is used throughout the machine.

Examples

$E = 27_8 = 010111_2 = 23$ in decimal notation.

$E = 72_8 = 111010_2 = -(00101_2) = -5$ in decimal (sign and magnitude) notation.

$M = 200\ldots0_8 = 0.10000\ldots000_2 = 0.5$ decimal

$M = 477\ldots7_8 = 1.00111\ldots111_2 = -(0.11000\ldots000_2) = -.75$ decimal.

The main advantage of one's complement notation is that addition and subtraction of two numbers can be carried out electronically without distinguishing the sign bits from any of the others, provided any 'spill' out of the sign position is added back into the low order position of the sum. This effect is termed 'end-around' carry, and is illustrated by the following example of exponent addition:

$E_1 = 010101_2 = 21$ in decimal notation

$E_2 = 111001_2 = -6$ in decimal notation

$E_1 + E_2 = 001110$

$\hookrightarrow +1$ (end around carry)

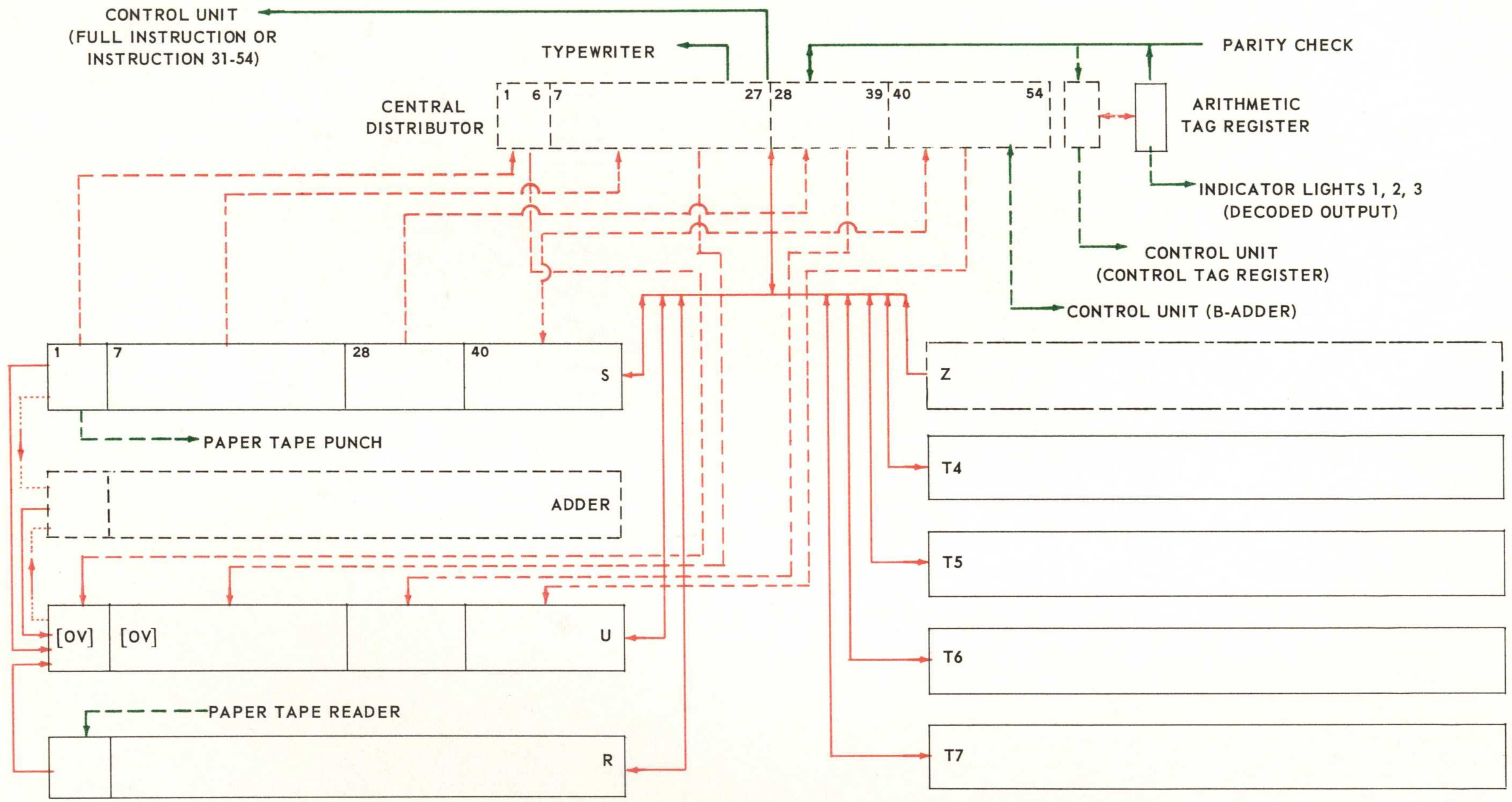
$\underline{= 001111} = 15$ in decimal notation.

There are two 'zeros' in this system, namely numerical fields of all 0's or all 1's. As in the sign-and-magnitude scheme, these are termed 'plus zero' and 'minus zero' when it is necessary to distinguish between them.

The floating point value of a machine word is given by $M \times 256^E$, which is a number in the approximate range $[10^{-75}, 10^{73}]$. For full details on floating point operations, see Section XI.

RICE COMPUTER

INFORMATION FLOW ARITHMETIC UNIT



🟥 (solid)	FULL WORD (54 BITS) INTRA-FLOW	🟩 (solid)	FULL WORD INTER-FLOW	———	ADDRESSABLE REGISTER
🟥 (dashed)	PARTIAL WORD INTRA-FLOW	🟩 (dashed)	PARTIAL WORD INTER-FLOW	– – – –	NON-ADDRESSABLE REGISTER
🟥 (dotted)	STATIC FULL WORD CONNECTION			——– – –	SYNTHESIZED NON-ADDRESSABLE REGISTER

FIG. 2

IV.

ARITHMETIC UNIT

The arithmetic unit accomplishes all arithmetic functions and has, in addition, facilities for temporary (or erasable) storage. The execution time for each operation varies from function to function and from operand to operand depending upon the number of zero bits in the operand.

The arithmetic unit is built around an information distributing device called "the central distributor", or CD. CD is not a register since it cannot store information but is only used to transfer information between registers and to and from the other units in the machine. Figure 2 is a block diagram of the arithmetic unit and is an expansion of part of the general diagram in Figure 1. The various registers are listed below and a paragraph description of each register and its function is given:

registers

abbreviation	name
U	universal register
R	remainder register
S	second operand register
T4	temporary store no. 4
T5	temporary store no. 5
T6	temporary store no. 6
T7	temporary store no. 7

Universal Register

The U register is involved in all arithmetic and logical operations. It is sometimes said to contain the 'first operand' (to distinguish this from the number brought to S before executing an operation). In arithmetic operations, the first operand will normally be the addend, minuend, multiplicand, or the high order part of the dividend. At the end of each operation, the principal result is in U, viz, a sum or difference, the high order part of a product or a quotient. U may also be shifted left or right either logically (all 54 bits) or arithmetically (mantissa only).

Remainder Register

The R register is used partly as an intermediate store in some operations, and partly as storage for results. Before division, R contains the low order part of the dividend; after division, it contains the remainder. After multiplication, R contains the low order part of a product. R may also be shifted, with or without connections to U.

Second Operand Register

Numbers coming from memory or from the control unit first appear in S before an operation is executed. Thus in arithmetic operations S initially contains the subtrahend, the multiplier or divisor. The contents of S after an operation is normally a complicated intermediate result and as such it is seldom used.

Temporary Storage Registers

There are four fast registers primarily used for storing temporary results. Instructions may be stored here to minimize execution time and may be fetched by the control unit in a manner similar to instructions in main memory.

RICE COMPUTER
INFORMATION FLOW IN CONTROL UNIT

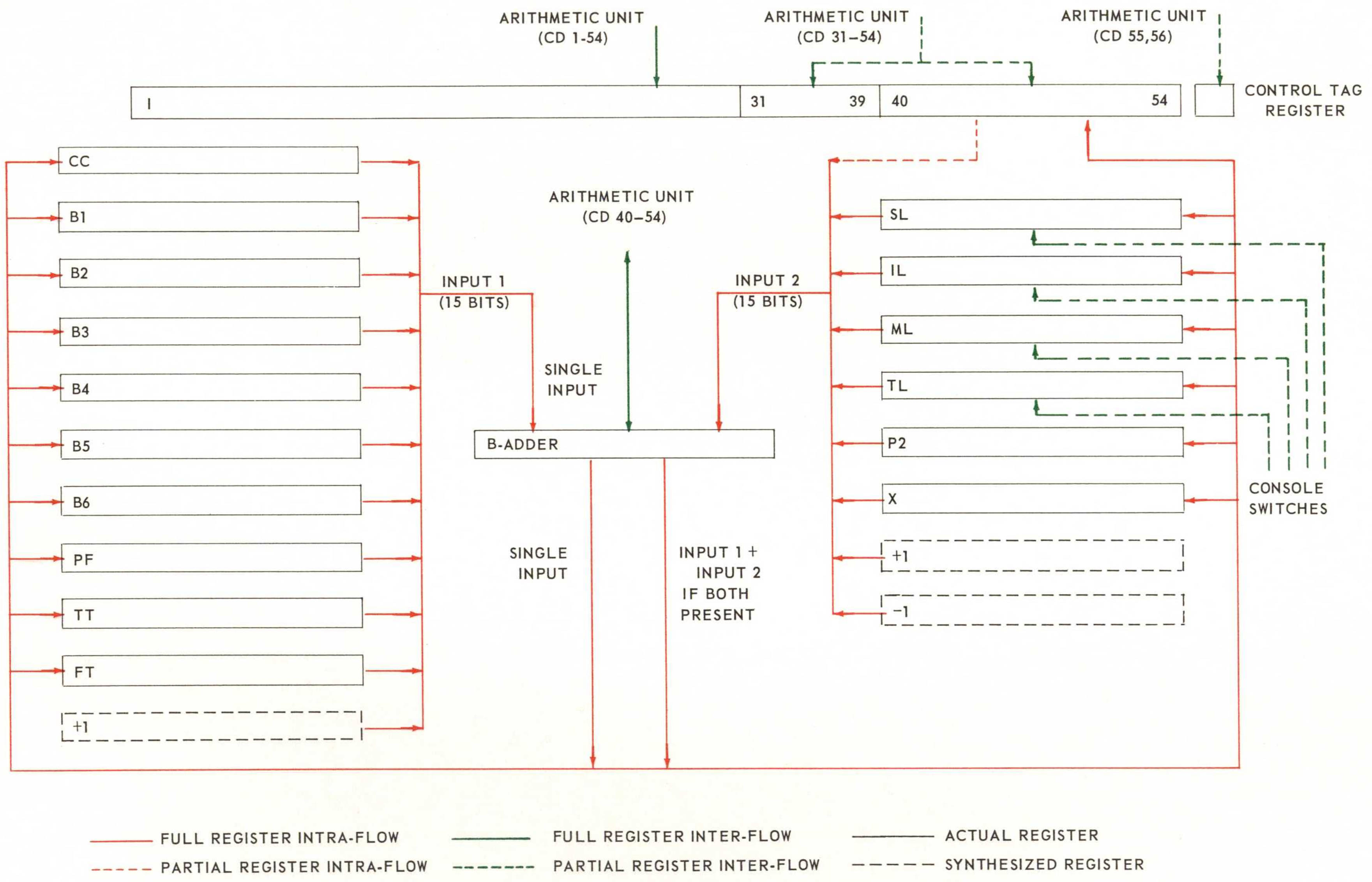


FIG. 3

V.
CONTROL UNIT

The machine's control unit has the task of accepting orders one by one into the I register and of causing the machine to carry out the operations specified according to the order in the I register. All address modification also takes place in this unit. Normally orders are obeyed by the control unit in the sequence in which they are stored in the memory. Sometimes, however, this sequence is broken and the control unit starts over at some new position in the memory. This is called a transfer of control. If control is transferred to a few locations back in the memory, the machine will repeat the operations specified by the intervening orders. It is possible to cause this repetition to occur any number of times. The machine also has special facilities for the repetition of a single instruction (see Section IX).

Because of the importance of program cyclic control, emphasis has been placed on special facilities in the control section to help the coder. This tends to make the description of the control section involved. However, the prospective coder is advised to master these additional features since they make possible most of the interesting calculations and will greatly shorten his time spent in coding.

The control unit is centered around a B-adder. This plays the part of a central distributor for the control section. The last 15 bits of the instruction register or the X register* can be connected to one side of this adder and one of 8 other registers described below can be connected to the other side. The adder output may then be gated (i.e., transferred) to any of these control registers, the arithmetic central distributor, or the memory. Figure 3 is a block diagram of the control registers and their interrelations. The various registers are listed below, and a brief description of each one is given:

Control registers

abbreviation	name
I	instruction register
CC	control counter (or location register)
B1	B register 1
B2	B register 2
B3	B register 3
B4	B register 4
B5	B register 5
B6	B register 6
PF	pathfinder

* Details of the special registers are given in Section IX.

Instruction Register

When an instruction is brought from memory into the control unit, it is placed in the instruction register, where it is decoded. The instruction register is a full length 54 bit word. The last 15 binary digits (bits 40-54) specify an address or location number. It is this number that is subject to modification. A summary of the instruction codes is given in Section VII.

Control Counter or Location Register

This register with a capacity of 15 bits determines the location in memory from which the next instruction is taken. Whenever a new instruction is brought into the instruction register from memory, CC (control counter) is advanced by 1. However, during the execution of a transfer or skip, the contents of CC may be changed to any number in the address range.

The control counter may also be used in exactly the same way as one of the B-registers.

B Registers

The Rice Computer has six B registers each containing 15 bits. One of the primary uses of B registers arises from their ability to modify the instruction address. When a given B register is appropriately specified (this is explained in the detailed discussion of the order code), the instruction is executed as if its address had contained the stated address plus the contents of the specified B register. The actual addition is carried out in the B-adder and only the right hand 15 bits of I are affected by this addition. The result of the addition is placed in M, the address section of I.

As in the arithmetic section, the B-adder operates on numbers in the one's complement notation. Thus, if it is required to decrement the address portion of I by 1, the number 77776 should be placed in an appropriate B-register.

Pathfinder Register PF

Whenever the order 'Transfer to subroutine' (operation code 40000) is about to be executed, the contents of CC are placed in PF: normally this will have the effect of automatically recording the address following that from which the transfer was made, and it is used in 'returning' from a subroutine.

The pathfinder may also be used as a B-register.

VI.

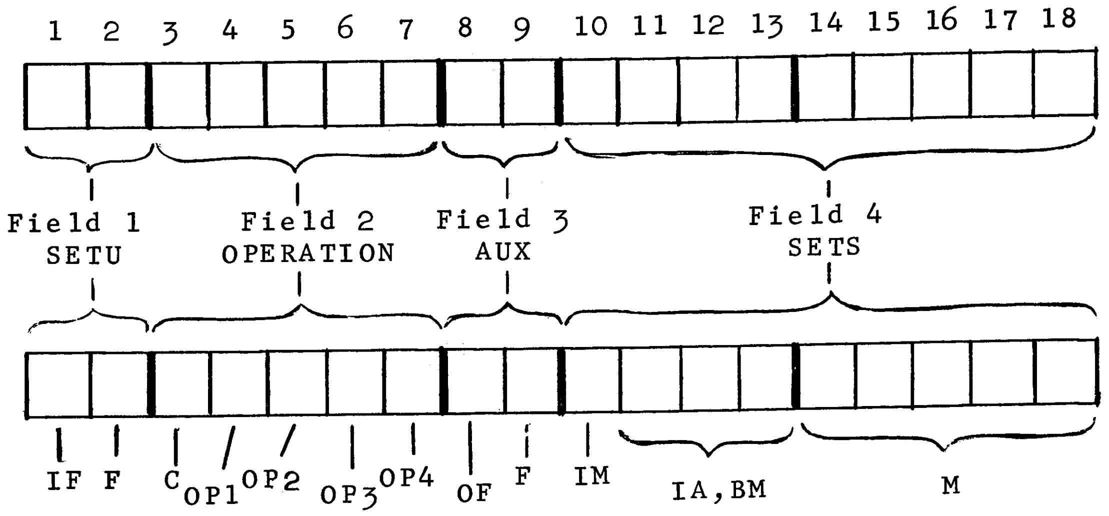
A SUMMARY OF THE RICE MACHINE ADDRESSING SYSTEM

OCTAL ADDR	NAME	DESCRIPTION AND USE
A Series F Registers		
0	Z	54 zeros, used for clearing.
1	U	54 bits + 2 overflow bits. Universal.
2	R	54 bits. Remainder.
3	S	54 bits. Secondary operand.
4	T4	54 bits. Temporary storage.
5	T5	54 bits. Temporary storage.
6	T6	54 bits. Temporary storage.
7	T7	54 bits. Temporary storage.
Electrostatic Memory Addresses		
10 to 77767		54 bits + 2 tag bits + 7 automatic parity check bits.
Special Purpose Registers		
77770	SL	15 bits. Sense lights.
77771	IL	15 bits. Indicator lights.
77772	ML	15 bits. Mode lights.
77773	TL	15 bits. Trapping lights.
77774	P2	15 bits. Pathfinder 2. CC→P2 on all non-sequential transfers of control.
77775	X	15 bits. Increment register.
77776	TT	15 bits. Magnetic tape output.
77777	FT	15 bits. Magnetic tape input.
B Series F Registers (addressable only in Field 1 and Field 3)		
0	CC	15 bits. Control counter. Holds address of next instruction.
1 to 6	B1...B6	15 bits. Used to modify addresses.
7	PF	15 bits. Pathfinder. CC→PF on Class 4 transfers (Field 2 = octal 40000).

VII.

SUMMARY OF THE INSTRUCTION WORD STRUCTURE AND OPERATION CODES

All instructions are divided into four major fields as illustrated, each field being given a name which indicates its general function. In addition, certain subfields are named for reference purposes.



Instructions are interpreted in the following sequence.

1. The contents of the location indicated by CC are brought to I, the Instruction register. CC is advanced by 1.
2. Field 1 is decoded, and a word in an F address is sent to U.

3. Field 4 is decoded, and a word is sent to S. An address (possibly new) is left in the M portion of I.
4. Field 2 is decoded, and arithmetical or logical work is done using the contents of U and S and/or the final address M.
5. Field 3 is decoded, and the contents of either U or R is sent to an F address, or certain changes are made in the contents of one of the B registers.

FIELD 1. Individual bits of IF are interpreted as follows:

Bit 1. 0, A series address.

1, B series address.

Bit 2. 0, algebraic value of contents of F goes to U.

1, absolute value of contents of F goes to U.

Bit 3. 0, do not change sign of U.

1, change sign of U.

F is interpreted octally as an address. Octal interpretation of IF yields the following:

0, +A	1, -A	2, $+\|A\|$	3, $-\|A\|$
4, +B	5, -B	6, $+\|B\|$	7, $-\|B\|$

FIELD 4. Individual bits are interpreted as follows:

	Bit 1.	0, contents of location M, (M), goes to S.
		1, address M goes to S.
IM	Bit 2.	0, algebraic value of (M) or M goes to S.
		1, absolute value of (M) or M goes to S.
	Bit 3.	0, do not change sign of S.
		1, change sign of S.
IA	Bit	0, do nothing.
		1, replace last 24 bits of I by last 24 bits of (M).
BM	PF B6 B5 B4 B3 B2 B1 CC	0, do nothing. 1, add contents of corresponding B-register to address M and leave sum in M portion of I.
M	15 bits.	Permissible M addresses include A series F registers (0-7), electrostatic memory addresses (10-77767)*, and special purpose registers (77770-77777).

* 10-20007 when only one memory bank is used.

Octal interpretation of IM yields the following:

0, $+(M)$	1, $-(M)$	2, $+\|(M)\|$	3, $-\|(M)\|$
4, $+M$	5, $-M$	6, $+\|M\|$	7, $-\|M\|$

Field 4 is decoded in the following sequence:

1. Increment M by the sum of all the B-registers corresponding to ones in BM, leaving the result in the M portion of I.
2. If IA bit is 1, replace last 24 bits of I by last 24 bits of (M) and return to step 1.
3. If IM bit 1 is 0, send (M) to S. If IM bit 1 is 1, send M to S.
4. Modify the sign of S as indicated by IM bits 2 and 3.

NOTE. Whenever a 15-bit quantity is sent to a 54-bit register, it occupies the last 15 bits of that register. Bits 1-6 are cleared to zero, and bits 7-39 are set equal to bit 40.

FIELD 2. The first triad, interpreted octally, determines the Class of operation as follows:

0,	Compare or test, and skip, jump, or transfer.
1,	Arithmetic operations.
2,	Substitute, set tags.
3,	Not used.
4,	Short registers, shifts,
5,	Logical operations.
6,	Input - output.
7,	Not used.

<u>Class 0.</u> OP1 specifies the type of transfer to be made if certain conditions are satisfied. OP2, OP3 and OP4 specify a set of zero, one, two, or three conditions. HALT and TRANSFER means stop; then when "continue" button is pressed, $(M) \rightarrow CC$

TRANSFER means $(M) \rightarrow CC$

SKIP means $(CC) + 1 \rightarrow CC$

JUMP means $(CC) + (X) \rightarrow CC$

(all) means execute transfer only if all tests are satisfied

(any) means execute transfer if any of tests are satisfied

OP1. 0, if (any) halt and transfer.

1, if (any) transfer.

2, if (any) skip by 1.

3, if (any) jump by (X).

4, if (all) halt and transfer.

5, if (all) transfer.

6, if (all) skip by 1.

7, if (all) jump by (X).

OP2. 0, no test.

1, is U mantissa sign positive (0).

2, is mantissa overflow indicator light on.

3, is exponent overflow indicator light on.

4, no test.

5, is U mantissa sign negative (1).

6, is mantissa overflow indicator light off.

7, is exponent overflow indicator light off.

OP3. 0, no test.

1, is U mantissa equal to zero.

2, is bit 54 of U equal to zero.

3, are all the sense lights corresponding to ones in M on.

4, is every bit in U zero.

5, is U mantissa different from zero.

6, is bit 54 of U equal to one.

7, are all the sense lights corresponding to ones in M off.

OP4. 0, no test.

1, is tag 1 indicator light on.

2, is tag 2 indicator light on.

3, is tag 3 indicator light on.

4, are all tag indicator lights off.

5, is tag 1 indicator light off.

6, is tag 2 indicator light off.

7, is tag 3 indicator light off.

NOTES.

Tag and overflow indicator lights are turned off when tested. Sense lights are not altered when tested.

In SKIP or JUMP (OP1 = 2, 3, 6 or 7) the contents of U and S are combined in a way determined by the test called for in OP3:

If OP3 = 0, 1, 2, 5 or 6, the arithmetic difference U - S is formed in U, fixed point or floating point according to the nature of the two numbers. See the discussion in class 1.

If OP3 = 4, U and S are compared logically through R as a mask. If U and S are identical in the bits where R is zero, U ends up all zeros; otherwise U will not be null. The "if null" test is then performed.

If OP3 = 3 or 7 (sense light test), U is not changed.

Class 1. Bits not mentioned are not used. Bits mentioned individually have no effect when equal to zero. OP2 and OP4 are interpreted octally. Operations are carried out in the order listed.

OP1. Bit 1. 1, interchange (U) and (S).

Bit 2. 1, clear R mantissa to sign of U mantissa.

Bit 3. 1, interchange (U) and (R).

OP2.

0,	fixed point add.	$(U)+(S) \rightarrow U$.
1,	fixed point subtract.	$(U)-(S) \rightarrow U$.
2,	fixed point multiply.	$(U)X(S) \rightarrow U,R$.
3,	fixed point divide.	$(U,R)\div(S) \rightarrow U,R$.
4,	floating point add.	$(U)+(S) \rightarrow U$.
5,	floating point subtract.	$(U)-(S) \rightarrow U$.
6,	floating point multiply.	$(U)X(S) \rightarrow U,R$.
7,	floating point divide.	$(U)\div(S) \rightarrow U,R$.

OP3. Bit 1. 1, do not normalize after a floating operation.

Bit 2. 1, then if OP2 = 0 or 1, the exponents of U and S are added or subtracted (as integers) in addition to the mantissa addition or subtraction.

If OP2 = 2, 3 or 7, U and R are interchanged after the arithmetic operation.

If OP2 = 4 or 5, U and R are adjusted so that the two represent a double precision floating point sum or difference.

OP4. 0, not used.

1, store (U) → M.

2, not used.

3, store (U) → M + (B6).

4, thru 7, not used.

NOTES.

Fixed point numbers normally have an exponent of 00. Fixed point operations on such numbers give a result in the same form. If either or both numbers have any other exponent, the exponent of the result is usually useless.

A floating point operation on two floating point numbers (U, S) normally gives a proper, normalized floating point result. If one of the two numbers is fixed point (exponent = 00), it is treated as if it were zero. The most important result of this convention is that an all zero word, null, behaves as a true floating point zero. However, if both U and S are fixed point, they are combined by a floating point operation properly as fixed point numbers.

After a multiplication, the high-order bits of the product are in U, the low-order bits in R. Before a division, the high-order bits of the dividend are in U and the low-order bits in R. After a division, the quotient is in U and the remainder in R.

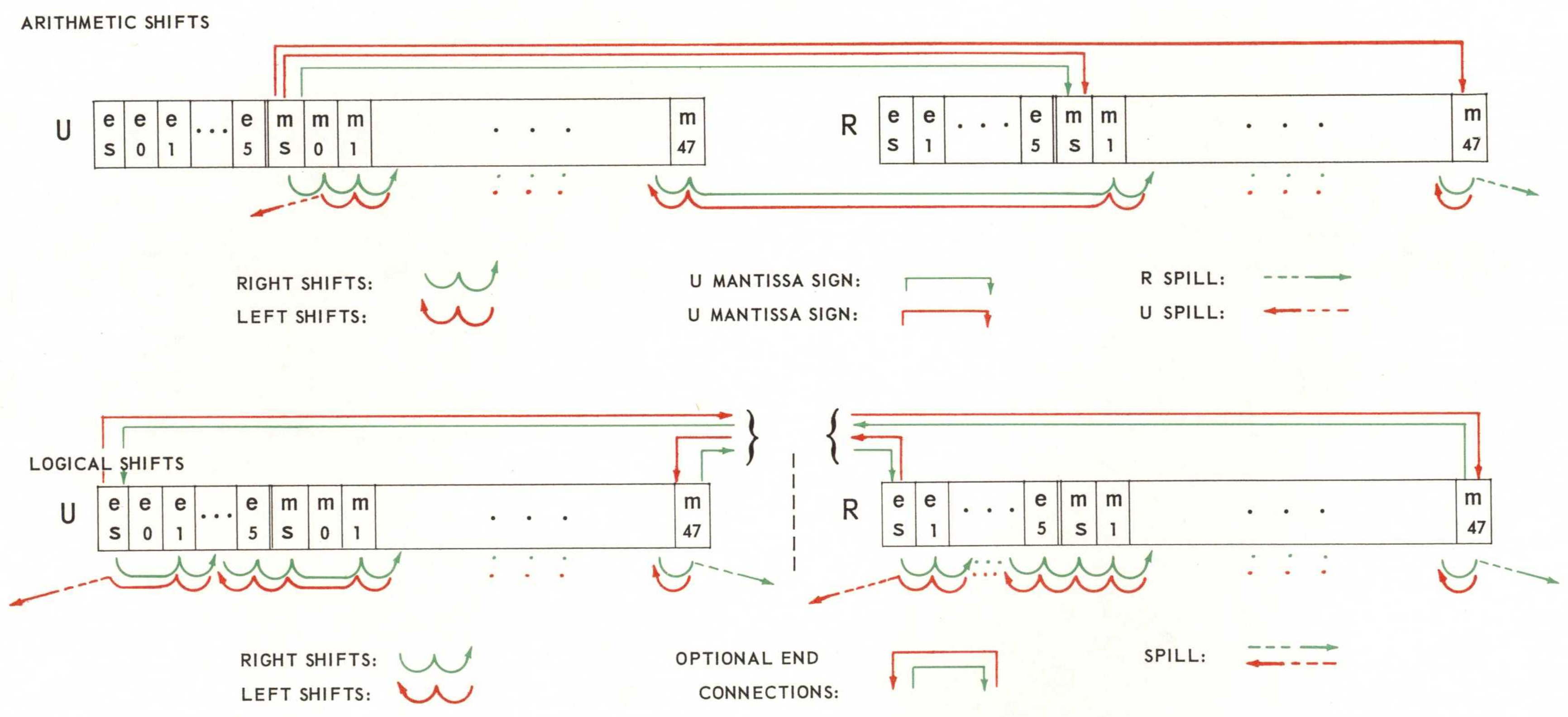
After either floating point multiplication or division, the exponent of R is set equal to the final exponent of U.

Examples:

10300	(U,R) $\div$ (S), fixed point
12300	(U) $\div$ (S) , fixed point
13300	integer divide
16300	(S) $\div$ (U) , fixed point
10401	floating point add to memory
10700	(U,R) $\div$ (S), floating point
12700	(U) $\div$ (S) , floating point

RICE COMPUTER

SHIFTING OPERATIONS



AFTER EACH LOGICAL SHIFT OF U, THE EXPONENT AND MANTISSA OVERFLOW BITS ARE SET EQUAL TO THE CORRESPONDING SIGN BITS. ZEROS FILL IN WHERE END CONNECTIONS ARE NOT SPECIFIED.

FIG. 4

Class 2. Bits mentioned individually have no effect when equal to zero. OP3 and OP4 are interpreted octally.

OP1.	Bit 3.	1, bits 1-6 of S replace bits 1-6 of U.
OP2.	Bit 1.	1, bits 7-27 of S replace bits 7-27 of U.
	Bit 2.	1, bits 28-39 of S replace bits 28-39 of U.
	Bit 3.	1, bits 40-54 of S replace bits 40-54 of U.

OP3. 0, clear ATR (Arithmetic Tag Register).

1, set tag 1 in ATR.

2, set tag 2 in ATR.

3, set tag 3 in ATR.

4, do not change ATR.

5, not used.

6, not used.

7, not used.

OP4. 0, not used.

1, store (U) in location M.

2, not used.

3, store (U) in location M+(B6).

4, not used.

5, not used.

6, not used.

7, not used.

<u>Class 3</u>. There are no orders in Class 3. Any Field 2 code of the form 3XXXX will be ignored.

<u>Class 4</u>. OP1 is interpreted octally, and its value determines the meaning of the other bits of the field. In each case, bits not mentioned are not used, and bits mentioned individually have no effect when equal to zero.

OP1. 0, set the B-register designated octally by OP4 to the value given by M.

1, add the number given by M to the B-register designated octally by OP4.

2, turn on or off the lights corresponding to ones in M, as indicated octally by OP4.

OP4. 0, turn on designated sense lights.

1, turn on designated indicator lights.

2, turn on designated mode lights.

3, turn on designated trapping lights.

4, turn off designated sense lights.

5, turn off designated indicator lights.

6, turn off designated mode lights.

7, turn off designated trapping lights.

3, set the special purpose register designated octally by OP4 to the value given by M.

OP4. 0, not used.

1, not used.

2, not used.

3, not used.

4, P2, pathfinder 2.

5, X, increment register.

6, TT, words-to-tape.

7, FT, words-from-tape.

OP1. 4, arithmetically shift the mantissa of U and R, connecting bit 47 of U mantissa to Bit 1 of R mantissa. Ignore exponent bits. Send U mantissa sign to R mantissa sign, and to vacated bits.

OP3. Bit 3. 0, shift right M places.

1, shift left M places.

5, logically shift all 54 bits of U and/or R, ignoring overflow bits in U.

OP3. Bit 1. 0, fill vacated bits of U with zeros.

1, fill vacated bits of U with spill from R.

Bit 2. 1, shift U left M places.

Bit 3. 1, shift U right M places.

OP4. Bit 1. 0, fill vacated bits of R with zeros.

1, fill vacated bits of R with spill from U.

Bit 2. 1, shift R left M places.

Bit 3. 1, shift R right M places.

6, bit count. Clear U, shift R right M places, and add each 1 which spills from R one at a time into U.

7, not used.

NOTES.

Shifts and bit count use the integer in M modulus 128 (decimally). If M is negative (bit 40 = 1), a shift will be carried out along the specified path but in the reverse direction.

Class 5. All 54 bits are treated on the same basis. OP1, 2, 3 and 4 are decoded and executed in succession. NOP means no operation. Hang up means a machine stop. Some meaning may be attached to these cases at a later time. A bar above a register symbol means logical complement. Address M is the final address formed in field 4.

AND and OR have the usual logical meanings, i.e., AND: if both registers have a 1, the result is 1, otherwise a zero; OR: if either or both registers have a 1, the result is 1, otherwise zero. EXTRACT: wherever a bit of R is equal to one, the bit in that position of S replaces the corresponding bit in U. Other bits of U are unchanged.

OP1. 0, NOP.

1, Hang up.

2, Hang up.

3, $U \rightarrow R$, $S \rightarrow U$.

4, $U \rightleftharpoons R$ (including overflow bits).

5, Hang up.

6, Hang up.

7, Hang up.

OP2. 0, NOP.

1, $\overline{U} \rightarrow U$.

2, $\overline{S} \rightarrow S$.

3, $\overline{U} \rightarrow U$, $\overline{S} \rightarrow S$.

4, $S \rightarrow R$.

5, Hang up.

6, Hang up.

7, Hang up.

OP3. 0, NOP.

1, S OR U → U.

2, EXTRACT S thru R → U (thru 1's of R).

3, Hang up.

4, Hang up.

5, Hang up.

6, Hang up.

7, Hang up.

OP4. 0, NOP.

1, U → M (store).

2, NOP.

3, U → M + (B6).

4, $\overline{U} \rightarrow U$.

5, $\overline{U} \rightarrow U \rightarrow M$.

6, $\overline{U} \rightarrow U$.

7, $\overline{U} \rightarrow U \rightarrow M$ + (B6).

Examples:

AND	50314	S AND U → U
ORU	50010	S OR U → U

SYM	53220	S,U symmetric difference
SYD	53220	S,U symmetric difference
SYS	53120	S,U symmetric sum
LDR	50400	Load R from S
ORU→	50011	(M) OR U → M

Class 6. Input-Output Orders

OP1. 0, paper tape and typewriter control*

OP2. 0, read triads: 3 bits (from channels 1,2,3 of the tape frame currently in the optical reader) are stored in the first 3 bits of R. A logical shift of UR left 3 places is carried out. If a '1' bit is spilled from the high order (exponent sign) bit of U before the last shift takes place, OPN is complete. Otherwise, a further triad is read.

1, read hexads: 6 bits (from channels 1,2,3,4,5,6 of the tape frame currently in the optical reader) are stored in the first 6 bits of R. A logical shift of UR left 6 places is carried out. OPN is completed by a spill from U as in the case of reading triads.

2, not used.

3, not used.

* See Table 1 for complete paper tape codes.

OP2. 4, punch hexads. A frame of six bits is punched from the exponent position of S, and the code 02 is inserted in the last 6 bits of U. U and S are then shifted left logically 6 places. If a spill from U is detected in the first 5 shifts, OPN is complete. Otherwise, a further hexad is punched.

5, punch hexads with 7th hole. As for OP2=4, except that a hole is punched in channel 7 of the paper tape.

6, punch triads. A frame of 3 bits is punched from the high order triad of S, and the code 2 is inserted in the last triad of U. U and S are then shifted left logically 3 places. If a spill from U is detected on the first two shifts, OPN is complete. Otherwise, a further triad is punched.

7, type. An eighteen digit octal number is typed from S by the console typewriter.

OP1. 1, printer control**.

OP2. 0, print nothing.

1, print character codes 00-37 only.

2, print character codes 00-77.

OP3. 0, do not space after printing.

1, space one line (1/66 page).

2, advance paper to first 1/22 page mark.

3, " " " " 1/11 " " .

4, " " " " 1/6 " " .

5, " " " " 1/3 " " .

6, " " " " 1/2 " " .

7, " " " new page .

** Printing takes place from a <u>print</u> <u>matrix</u> of either 100 or 200 words in memory, whose first word address is given by the final address, M, formed in I. Each pair of words in the matrix correspond to a line of a particular character code of 108 characters. Thus, words 1, 2 give the code for character '0'; a '1' in either of these words causes a '0' to be printed in the appropriate type position on the line. Words 3, 4 correspond to character '1', 5, 6 to character '2', etc. See Table 2 for complete printer character codes.

TABLE 1

FLEXOWRITER CODES

key		code	
L-case	U-case	7654	321
a	A	0100	000
b	B	0100	001
c	C	0100	010
d	D	0100	011
e	E	0100	100
f	F	0100	101
g	G	0100	110
h	H	0100	111
i	I	0101	000
j	J	0101	001
k	K	0101	010
l	L	0101	011
m	M	0101	100
n	N	0101	101
o	O	0101	110
p	P	0101	111
q	Q	0110	000
r	R	0110	001
s	S	0110	010
t	T	0110	011
u	U	0110	100
v	V	0110	101
w	W	0110	110
x	X	0110	111
y	Y	0111	000
z	Z	0111	001
0	(	0000	000
1	)	0000	001
2	*	0000	010
3	#	0000	011
4	λ	0000	100
5	σ	0000	101
6	α	0000	110
7	β	0000	111
8	γ	0001	000
9	Δ	0001	001
π	Π	0001	010
$\times$	Σ	0011	101
+	/	0010	000
-	$\rightarrow$	0010	001
<	$\leqslant$	0111	010
.	,	0011	111
=	|	0011	110
carriage return		0010	100

key		code		key		code	
L-case	U-case	7654	321	L-case	U-case	7654	321
upper case		0111	100	1/2 space down		0010	011
lower case		0111	110	1/2 space up		0010	010
tab		0010	101	space		0001	100
tape feed		1011	000	code stop		1010	111
back space		0010	110	code delete		1111	111

Control punches

"stop code" - turns off repeat mode light when reading

7th hole - frame is <u>not</u> read into U

TABLE 2

PRINTER CODES

symbol	octal code	binary code	symbol	octal code	binary code
0	00	00 0000	A	40	10 0000
1	01	00 0001	B	41	10 0001
2	02	00 0010	C	42	10 0010
3	03	00 0011	D	43	10 0011
4	04	00 0100	E	44	10 0100
5	05	00 0101	F	45	10 0101
6	06	00 0110	G	46	10 0110
7	07	00 0111	H	47	10 0111
8	10	00 1000	I	50	10 1000
9	11	00 1001	J	51	10 1001
a	12	00 1010	K	52	10 1010
b	13	00 1011	L	53	10 1011
c	14	00 1100	M	54	10 1100
d	15	00 1101	N	55	10 1101
e	16	00 1110	O	56	10 1110
f	17	00 1111	P	57	10 1111
+	20	01 0000	Q	60	11 0000
-	21	01 0001	R	61	11 0001
/	22	01 0010	S	62	11 0010
.	23	01 0011	T	63	11 0011
.	24	01 0100	U	64	11 0100
x	25	01 0101	V	65	11 0101
x	26	01 0110	W	66	11 0110

TABLE 2 cont'd

symbol	octal code	binary code	symbol	octal code	binary code
Δ	27	01 0111	X	67	11 0111
Δ	30	01 1000	Y	70	11 1000
*	31	01 1001	Z	71	11 1001
|	32	01 1010	<	72	11 1010
(	33	01 1011	≤	73	11 1011
)	34	01 1100	↑	74	11 1100
×	35	01 1101	←	75	11 1101
=	36	01 1110	→	76	11 1110
,	37	01 1111	↓	77	11 1111

FIELD 3. OF is interpreted octally as follows. F is interpreted octally as an A or B series address.

OF.
- 0, store U in the A-series register designated by F.
- 1, store R in the A-series register designated by F.
- 2, add 1 to the contents of the B-series register designated by F.
- 3, add (X) to the contents of the B-series register designated by F. X is the increment register.
- 4, store the last 15 bits of U in the B-series register designated by F.
- 5, store the last 15 bits of R in the B-series register designated by F.
- 6, subtract 1 from the contents of the B-series register designated by F.
- 7, store the M portion of I in the B-series register designated by F.

VIII.

EXAMPLES OF SINGLE INSTRUCTIONS

In order to illustrate the procedure of composing single instructions, the following list is presented. The instructions are grouped according to operation class: for each, its numerical code is given, together with an equivalent symbolic representation. No explanation of the latter is offered, but full details of the symbolic representation of orders may be obtained from programming memoranda.

Following each symbolic code is a brief interpretation of the order. The reader is strongly advised to check these numbers against the outline of the instruction codes given in Section VII.

Class 0

01 03030 00 4000 30002

symbolically: IF(SLN)JMP a 30002

Interpretation: if sense lights 2, 3 and 14 are on, (CC) + (X)$\rightarrow$CC; otherwise proceed to next instruction. When 3 or 7 in Op 3 of OPN(SLN or SLF, respectively) is used, the address portion of the instruction always denotes the lights to be tested; thus, only SKP or JMP options may be elected with these tests.

01 00000 00 4000 55555

symbolically: HTR a55555

The program halts unconditionally with this instruction displayed in I. Such an instruction may provide a stop at the end of a program; an appropriate address field will then allow re-entry when the 'continue' switch is depressed.

04 02110 21 0002 01000

symbolically: T4 IF(POS)SKP 1000+B1,B1+1

Interpretation: if (1000+(B1))$\leq$(T4), next execute instruction two beyond one given; if (1000+(B1))> (T4), next execute instruction immediately following one given; always increment (B1) by 1.

01 05552 00 4000 00134

symbolically: IF(NNZXTG2)TRA a134

Transfer is effected if all tests are satisfied; no subtraction takes place. The presence of the numeric bit saves the time of a fetch from memory in SETS since (134) does not pertain to the execution of this instruction. Interpretation: if (U)<0 and indicator light 2 (arithmetic tag 2 indicator) on, transfer to location 134.

Class 1

06 10401 00 0000 03742

symbolically: T6 FAD→ 3742

Interpretation: floating point (T6)+(3742)→3742. With this instruction a cumulative sum may be collected at location 3742, successive terms having been computed and stored in T6.

41 13300 53 4004 00000

symbolically: B1 IDV aB2, R→B3

Interpretation: (B1)→U, then (B2)→S, then U_{M_s}→R_M, then interchange (U)←→(R), then fixed point division (U,R)/(S)→U (quotient) and R (remainder), then (R)→B3 in AUXILIARY; hence, (B1) modulo (B2)→B3.

Class 2

01 21700 07 0040 00032

symbolically: CLA 32+B5, U→T7

Interpretation: (32+(B5))→S in SETS; $(S)^{1,54}$→ $U^{1,54}$ in OPERATION, then (U)→T7 in F3; hence, (32+(B5))→T7.

44 21641 00 0000 05120

symbolically: B4 RPA,WTG 5120

Interpretation: (B4)→U; (5120)→S; then $(S)^{1,39}$→ $U^{1,39}$, and (U) with old tag →5120; hence, (B4) replaces address at location 5120 and tags are unchanged.

Class 4

Note: all instructions of this class operate with the last 15 bits of the instruction register so that (S) never pertains to execution.

46 40006 45 4000 00001

symbolically: B6 SB6 a1,U→B5

Interpretation: (B6)→U in SETU, 1→B6 in OPERATION, then (U)→B5 in AUX; hence, B6 is set to 1 and old (B6) is retained in B5. The U register may be used for transfer of data since it is not involved in the execution of OPERATION.

15 42002 05 4000 20000

symbolically: -T5 MLN a20000,U→T5

Interpretation: -(T5)→U in SETU, then ML#2 turned on in OPERATION, then (U)→T5 in AUX. The U register is used for implementation of -(T5)→T5. Only lights designated by 1's in the address of the instruction are affected; all others are left unchanged.

04 43005 02 4000 77774

symbolically: T4 STX a-3,U→R

Interpretation: (T4)→U in SETU, 77774→X in OPERATION, then (U)→R in AUX.

07 44000 07 4000 00001

symbolically: T7 DMR a1,U→T7

Interpretation: (T7)→U in SETU, then arithmetic shift of connected U and R (double mantissa) right 1 in OPERATION, then (U)→T7 in AUX. Effectively, (T7)/2→T7. Similarly, multiplication by a power of 2 can be accomplished by a double mantissa left shift, provided (R) is appropriately pre-set and overflow is observed.

04 45066 51 4000 00005

symbolically: T4 CRL a5,R→B1

Interpretation: (T4)→U in SETU, then logical circular shift of U and R left 5 places, then (R)→B1 in AUX. If (R) is all zeros before this instruction is executed, the first 5 bits of (T4) are sent to B1 as an integer.

Class 5

05 50314 00 0000 00061

symbolically: T5 AND 61

Interpretation: (T5) 'and' (61)→U, logical conjunctive operation. If T5 contains a mask, those bits of (61) corresponding to 1's in the mask are retained in U; all other bits of (U) will be zero.

Class 6

00 60001 00 4002 00100

symbolically: Z RTR$\rightarrow$ a100+B1

Interpretation: read successive triads from paper tape into R_e and shift U and R connected logically left 3 places after each read; when spill from high order U is detected, (U)$\rightarrow$100+(B1) to terminate OPERATION. Thus, the first non-zero triad read from paper tape is that which is detected as spill when it is shifted off the high order end of U, and the following 18 triads read from paper tape are stored at 100+(B1). If each absolute numeric word (18 triads) punched on paper tape is preceded by a carriage return or tabulate punch, the carriage control characters provide spill, and successive numeric words will be stored correspondingly in memory by such an instruction provided B1 is incremented appropriately.

IX.

SPECIAL REGISTERS AND TAGS

SPECIAL REGISTERS

The eight machine addresses 77770 through 77777 refer to eight 15-bit registers in the control section of the machine. Four of the registers are displayed at the console and in each of these individual bits may be set to '0' or '1' states by means of toggle switches: these are the Sense, Indicator, Mode and Trapping registers. Individual bits in these registers are numbered 1 - 15 from left to right in the word, i.e., the high order bit is number 1. A brief summary of the function of each special purpose register follows.

Sense Lights (SL) Address 77770

By means of Class 4 operations, individual sense lights may be turned on or off. By means of Class 0 operations, these may be interrogated one or more at a time, and followed by a SKP or JMP type transfer. The status of the lights is not affected by a Class 0 interrogation.

Indicator Lights (IL) Address 77771

By means of Class 4 operations, individual indicator lights may be turned on or off. The first five lights will also be turned on by particular conditions which arise in the machine, i.e.,

IL#1 By a word carrying Tag 1 (2 or 3) being
IL#2 brought from memory to S after the
IL#3 decoding of the SET S field.
IL#4 By a mantissa overflow in U.
IL#5 By an exponent overflow in U.

These lights may be interrogated by any Class 0 order. They remain on until tested, or until turned off manually or by a Class 4 order.

Mode Lights (ML) Address 77772

By means of Class 4 operations, individual mode lights may be turned on or off. Only mode lights nos. 1, 2, 3, and 4 have particular significance at present.

ML#1: Non-rounding Mode.

When ML #1 is on, a rounding bit (bit 18 or octal 1 in OP3) in a Class 1 instruction is ignored.

ML #2: Repeat Mode.

When ML #2 is turned on, the normal sequencing of instructions is halted after the next command is fetched into I: the machine is then said to be in the 'Repeat Mode.' In this situation, the command in I will be obeyed repeatedly until one of four conditions arises:

(a) A B-register is counted to zero in AUX.

(b) A tagged word enters the arithmetic unit and a tag indicator light is turned on.

(c) A test in Class 0 is satisfied.

(d) [If the command is a 'read' order] A 'stop code' is sensed on paper tape.

When one of these conditions is satisfied, the current executive sequence is completed and then the next instruction is fetched to I in the 'normal' way. It is important to note that if I is subject to B-modification, the last 15 bits of I will change at each decoding of the SETS field.

EXAMPLE: Let (B1) = 1, (B2) = 200. Then the order I = 00 20001 62 4002 00777, when executed in the repeat mode, will clear to zero the words in addresses 1000 to 1177.

ML #3: Trapping Mode.

When ML #3 is turned on, the normal sequencing of instructions may be changed if certain conditions (selected by the status of the Trapping Lights) arise in the machine. The point at which normal sequencing is disturbed is also determined by the status of TL. If ML #3 is off, no trapping will occur, irrespective of the condition of TL. When a 'trap transfer' takes place, ML #3 is automatically turned off, and transfer is made through a 'trap address' by means of the following instruction, which is placed automatically in I:

```
01 01000 00 4400 AAAAA
```

Here 'AAAAA' stands for one of eight possible trap addresses, which is dependent on the selected trap condition.

ML #4: Suppress Tag Mode.

When ML #4 is on, indicator lights 1, 2, and 3 (the tag indicators) are not turned on when a tagged number is brought into the arithmetic section. This mode is primarily useful in avoiding exiting from the repeat mode when a tagged number is brought into S.

Trapping Lights (TL) Address 77773

By means of Class 4 operations, individual trapping lights may be turned on or off. When the machine is running in the trapping mode, trap transfers will take place if the lights corresponding to the conditions listed below are turned on: in each case the corresponding trap address and the point of trapping are given, and it is a matter for the coder to ensure that this completes a transfer to a sequence of coding to be obeyed if a trap condition is satisfied.

LIGHT	CONDITION	TRAP TIME	ADDRESS (OCTAL)
#1	Arith. Tag 1	After SETS,	15
#2	Arith. Tag 2	before OPN	16
#3	Arith. Tag 3		17

LIGHT	CONDITION	TRAP TIME	ADDRESS (OCTAL)
#4	Control Tag 1	After fetch-	11
#5	Control Tag 2	ing I, before	12
#6	Control Tag 3	SETU	13
#7	Control Tag 1	After	11
#8	Control Tag 2	AUXILIARY	12
#9	Control Tag 3		13
#10	(U) positive	After AUXIL-	14
#11	(U) negative	IARY, and after	14
#12	MOV Light on	tag tests in-	14
#13	EOV Light on	dicated by TL	14
		#7, #8 and #9	
#14	Not used.		
#15			

Second Pathfinder (P2) Address 77774

On all modifications to CC except for the initial increment by 1 at the beginning of an instruction sequence, the initial contents of CC are stored in P2. This register is mainly used for engineering purposes.

Increment Register (X) Address 77775

The contents of this register are used in Class 0 transfers of the 'JMP' variety (03XXX or 07XXX), and by the AUXILIARY operation 'Bi+X', i.e., the code 3i. It may be loaded by a Class 4 order.

To Tape (TT) Address 77776

From Tape (FT) Address 77777

These registers are used in control of magnetic tape input and output by the machine and are not generally used by the coder.

Storage in Special Purpose Registers

A special purpose register may be treated partly, but not entirely, as any other storage register in the machine. It contains only 15 bits, and hence on fetching it to S it is stored in bits 40-54 of S, while bits 1-6 of S are set to 0, and bits 7-39 are set equal to bit 40. A fifteen bit number may be stored in P2, X, TT or FT from bits 40-54 of U. Upon storing to SL, TL, ML or IL however, a logical 'or' takes place with the bit pattern already in that register. It should be noted for a bit in one of the latter four registers that unless the corresponding toggle switch is in the 'neutral' position, its status is invariant with respect to store operations.

TAGS

Each general location in electrostatic memory contains, in addition to a 54-bit instruction or data word, two tag bits. It has already been indicated that when such a word is brought to S, the condition of the tag bits may cause IL #1, #2 or #3 to be turned on. Additionally, the tag bits are stored temporarily in a non-addressable two-bit Arithmetic Tag Register (ATR). The status of ATR is only changed when a new word is brought from electrostatic memory (by substitution of the new tag bits). When a fast register, or one of the special purpose registers, is brought to S, ATR is unchanged, and so are IL #1, #2 and #3. Arithmetic tag trapping takes place on the basis of the bits in ATR. The operations in Class 2 provide for ATR to be cleared, or set to any particular value after loading S.

When a store sequence takes place, the tag bits are taken from ATR.

In the control unit, corresponding to ATR is a control tag register, CTR, of two bits, which is reset and loaded when any instruction is brought to I. Control trap transfers take place on the basis of the bits in CTR. There is no means of directly altering CTR.

Figure 2 illustrates the role of the tag bits in data transfers in the machine.

X.

MAGNETIC TAPES

The magnetic tape facilities are not yet finished. All of the machine users await their completion hopefully and impatiently.

XI.

BINARY POINT LOCATION AND FLOATING POINT ARITHMETIC

Fixed point

In fixed point operations the exponent bits are not used. U, R and S should be thought of as 47 bit (mantissa) registers, each supplied with a sign bit (bit 7). These registers obey rules very much the same as those obeyed by the registers of a desk calculator. Any of the usual desk calculator conventions may be used in regard to the location of the binary point in each register. There are two simple and common conventions, however, that are used more than any others:

Fractional fixed point: The binary point in all registers is assumed to be between bits 7 and 8, just to the left of the mantissa. The number range is therefore from $-(1 - 2^{-47})$ to $+(1 - 2^{-47})$. The result of any arithmetic operation must also fall in this range. In the case of addition or subtraction, it is possible to exceed the permissible number range (indicated by the mantissa overflow indicator, IL#4, turning on). It is not possible to develop overflow in multiplication. After multiplication, the upper half of the product is found in U (with the standard binary point convention); the mantissa of R is the continuation of this product. The sign bit of R will agree with the sign bit of U. After two fixed point numbers, A and B, are multiplied (with the operation code 10200), we have

$$A \times B = (U) + (R)2^{-47}$$

After a division of A by B (code 10300), we have

$$(U) \times B + (R)\times 2^{-47} = A.$$

Note that the remainder, in R, may be interpreted in two ways (1) using the standard binary point of R, the true remainder is $(R)2^{-47}$; (2) alternatively, the binary point of the remainder in R is actually 47 places to the left of the usual point in R. These remarks are also true of a desk calculator. Ordinary division (code 10300) uses the double length register U, R as the dividend. The code 12300 will clear R to zero (of appropriate sign) and thus accomplish the single length division, $(U) \div (S)$. In case the quotient exceeds the permissible number range (i.e., magnitude $\geqslant 1$), the divide check light (IL#6) will turn on.

<u>Integer fixed point</u>: The standard binary point convention in this system is that the point lies at the right end of all registers, i.e., all numbers are signed integers. The operation codes 10000 and 10100, of course, serve as integer add and subtract orders. Multiplication

may be accomplished by 10200; however, the integer product will be found in R. If one is sure the product will not be as large in magnitude as 2^{47}, the code 10220 will introduce a final U, R interchange, leaving the product in U. Integer divide is accomplished by 13300. The integral quotient will be found in U and the remainder in R. For some purposes, one may want to use 13320, thus leaving the remainder in U. Note that addresses and the contents of B registers are brought into the arithmetic unit as integers according to these conventions. However, the integer address range is effectively $-(2^{14} - 1)$ to $+(2^{14} - 1)$, since the left-most bit of an address acts as a sign bit when a short register is brought into a long register.

Floating Point

As has been explained previously, floating point numbers use for a fractional part the same mantissa as do fixed point numbers. The binary point convention is the same as in fractional fixed point. The left-most six bits of the word are used to hold an integral exponent. The highest order bit is used as a sign bit (in the usual one's-complement system) so the exponent range is -31 to +31 (decimally) or -37 to +37 (octally). The assumed base is $2^8 = 256$.

The floating point arithmetical orders are primarily designed to accept normalized floating point numbers and produce normalized floating point answers. A normalized floating point number has no more than seven lead zeros in the mantissa of a positive number (or no more than seven lead ones in a negative number) so the mantissa is in the range

$$2^{-8} < |\text{mantissa}| < 1 - 2^{-47}$$

The operation codes 10400 and 10500 will form (U) + (S) in U and (U) - (S) in U, respectively. Code 10600 will form (U) X (S) in U, with the lower (less significant) portion of the product in R. R will have the same exponent and mantissa sign as U, so the exact product is (U) + (R) 2^{-47}. The code 10700 will divide (U,R) by (S), leaving the quotient in U and the remainder in R. Just as in fixed point division, if A is divided by B,

$$(U) \times B + (R) \times 2^{-47} = A.$$

The code 12700 clears R before performing the division; the result is then that the single length number in U is divided by the number in S.

In floating point arithmetic it is not possible to get mantissa overflow. If during the process of the floating operation and subsequent normalization, an exponent larger than decimal 31 appears, the exponent overflow light (IL#5) turns on. On the other hand, if the exponent should count down to less than -31, we have an exponent underflow condition. No indicator light is provided; instead the U and R registers are both cleared to null, i.e., to plus zero. The philosophy is that any number requiring an exponent less than -31 is to be treated as zero.

After addition or subtraction, a maximum of five normalization left shifts (of eight bits each) is allowed. The difference between two identical floating point numbers, for example, will appear with mantissa equal to zero (minus zero, as is the rule in one's-complement systems) and with an exponent decreased by five.

After multiplication, a maximum of twelve normalization left shifts is allowed. No provision is made for normalization after division since it is not needed if the dividend and divisor are normalized.

If one of the two input numbers to a floating operation is fixed point (distinguished by a plus zero (octal 00) exponent) and the other is floating point, the fixed point number is treated as a true zero. This feature is primarily useful in making the null number, plus zero, obtained as an underflow or from the address zero, behave as a true floating point zero:

$$A \pm 0 = A$$

$$A \times 0 = 0$$

$$0 \div A = 0$$

In addition, floating point addition of a (possibly unnormalized) number A to the contents of address zero serves to normalize A.

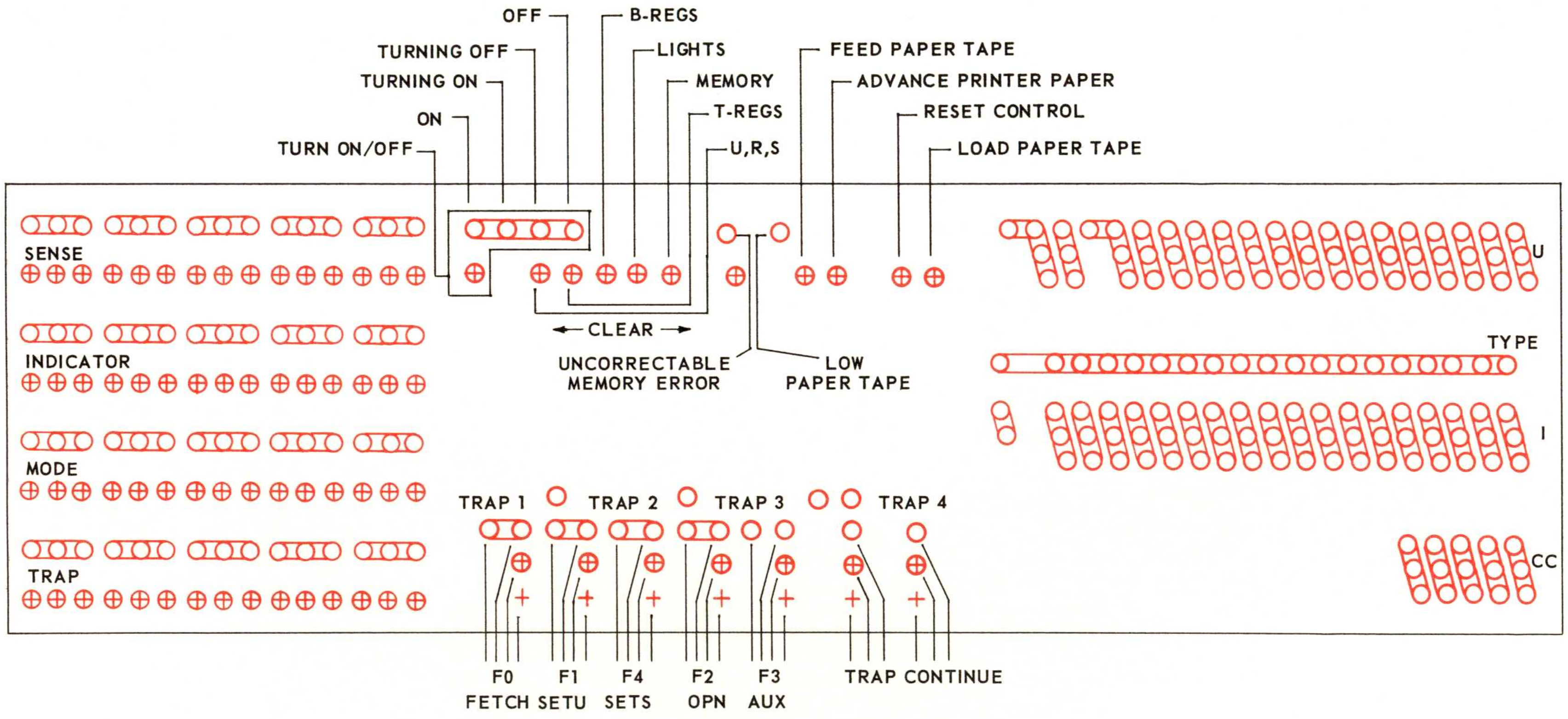
If both of the two input numbers are fixed point, a fixed point operation is carried out.

Zero. In all floating point systems, the number zero always requires special treatment. We have given some of the properties of zero in the preceding paragraphs. It may be well to summarize them here.

If the mathematical result of an arithmetic operation is zero, in our one's-complement system it normally appears as minus zero (all ones). The only exceptions are such cases as (+0) + (+0), (+0) X (+0), etc. In exponents, a plus zero is taken as an indication of a fixed point number. Null (Z, +0 or the contents of the Z register (address 0)) behaves as a proper fixed point or floating point zero.

If two equal floating point numbers are subtracted (and exponent underflow does not occur during the subsequent normalization), the result is an approximate zero. The exponent will be different from 00 (it will be exactly five less than the exponents of the two numbers) but the mantissa will be minus zero. This number will behave as a true zero when added to another floating point number with equal or greater exponent; when added to a floating point number with an exponent smaller by six or more, all of the mantissa of the number with the smaller exponent will be shifted off and lost when it is unnormalized in preparation for the addition. The result will be another approximate zero. If the difference in exponents is less than six, some of the low order bits in the non-zero mantissa will be lost.

RICE COMPUTER
CONSOLE DISPLAY AND CONTROLS



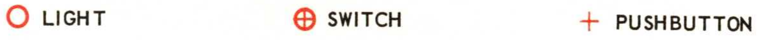


FIG. 5

XII.

MANUAL CONTROLS

A modified IBM typewriter and a set of switches and lights for controlling the machine and observing the status of particular registers are located at the operator's console. In addition, other units on the machine have controls which are of interest to the operator.

Typewriter

This is used for input and output of octal numbers only. On the typed form, all input information appears in red and output in black. Apart from its own keys, the typewriter is controlled by a set of push-buttons situated on the console immediately to the left of the typewriter. In order to type into a register, the following procedure is sufficient:

1. Depress the red 'Read-in' button. At this point the red light on the typewriter control panel will be turned on, indicating that it is ready to read information to the machine.
2. Depress the white button corresponding to the register which is to be loaded. Any fast or special purpose register may be selected, together with the instruction register and a number of others (corresponding to the unlabelled buttons) which are used

for engineering purposes. At this point the typewriter carriage will return, and a one or two letter mnemonic corresponding to the selected register will be typed at the beginning of the line. If a short register has been selected, the carriage will 'tab' across 13 positions. The selected register will be cleared.

3. Type the octal number to be loaded into the register (either 5 or 18 digits). After completing the loading, the carriage automatically 'tabs' across five positions. At this point on the line, notes may be typed without affecting the status of the machine.
4. To type into other registers, repeat from step (2).

In order to type out the contents of a register, the following procedure is sufficient:

1. Depress the green 'Read-out' button.
2. Depress the white button corresponding to the register which is to be read out. At this point, the carriage automatically returns, and a one or two letter mnemonic is typed out, followed by the content of the selected register.
3. To read out of the other registers, repeat step (2).

Portable Keyboard

Situated on the console is a small keyboard which may be used in a manner analogous to the typewriter for the input of numbers to U, I and CC. Depressing one of these three keys is equivalent to performing step 1 of the typewriter read-in sequence, followed by the selection of U, I or CC in step 2. Step 3 may be executed from this keyboard, or from the typewriter.

Console Switches

At the left side of the console display panel is a bank of four registers (Figure 5), each containing 15 bits grouped into five triads. These registers are (reading downwards) the Sense, Indicator, Mode and Trapping Light displays, together with their associated toggle switches. Each toggle switch has three positions: up ("on", "1"), down ("off", "0") and neutral. Only when a toggle switch is neutral may the corresponding light be turned on and off under machine control.

Console Controls

At the center of the console display are push-buttons and switches which are used to control the execution of programs. Starting at the top left-hand corner are the following sets of controls:

Machine On-Off Switch and Lights: The on/off switch may be used to turn the machine on whenever the (white) 'off' light is on. The sequence of turning on various sections of the machine is automatically controlled and is completed after about 4 minutes, when the (green) 'on' light will be turned on. This switch also controls input-output equipment, and it is therefore advisable to run out printed paper or punched tape before turning off the machine.

Clear Switches: Depressing one of these five switches causes the registers in the corresponding section of the machine to be cleared to zero, viz:

1. U, R and S
2. T4, T5, T6, T7
3. All eight B-registers
4. All special purpose registers, I, ATR, and CTR.
5. Memory (including tags). To clear memory completely, this switch must be held down for about a second.

Many-One Switch: This is used for engineering tests only and should normally be in the neutral position.

Feed Paper Tape: Depressing this switch causes paper tape to be punched at the tape punch with 'tape feed' characters for as long as the switch is held down.

Advance Printer Control: Depressing this switch causes paper to be passed through the printer until a new page is positioned for printing, or until the switch is released. Raising the switch effects continual paper advance until the switch is released.

Reset Control: Depressing this switch turns off all control flip-flops.

Load Paper Tape: Depressing this switch has the following effect, assuming paper tape has been placed in the reader:

(1) U, I and CC are cleared. The number 00 60020 00 0000 00000 is stored in I. (A read triads order).

(2) An execution sequence is started in OPN (Field 2). The effect of this is that triads are read into U until a spill occurs at the high order end of U. At this point Field 2 terminates, Field 3 (no operation) is executed, and the next instruction is brought to I from U.

(3) Since the machine is in the repeat mode, it will execute the order in I until one of the

four necessary conditions arises for leaving the repeat mode. During execution of this order, (CC) = 1.

Field Control: Corresponding to each field in an instruction are two lights, a switch and a push button on the Console Controls. A single instruction cycle is executed in five separate steps, during each of which a certain field is decoded, and at the end of which the following step is initiated or 'primed'. In order to initiate a program, therefore, it is sufficient to prime manually the execution of one field of the instruction cycle. An instruction cycle is considered to begin with the 'FETCH' (Field 0) in which the next instruction to be executed is brought to I. On the console, the field control lights are arranged from left to right according to their order of execution: FETCH, SETU, SETS, OPN, AUX (See Section VII).

In order to prime a field, the appropriate switch should be raised and then lowered to neutral. In this state, the left one of its two lights will be on.

In order to execute a primed field, the appropriate switch should be depressed. During execution, the right one of its two lights will be on. At the end of execution, the following field (FETCH following AUX) will be primed. An alternative method of executing a primed field when the switch is neutral is to press the appropriate push-button.

Each field control switch may be locked in any of its three positions. The 'normal' state during program execution is to have all five switches down, permitting fully automatic execution once the cycle has been entered.

<u>Continue Light, Switch and Push Button</u>: The continue switch is effective only when a 'halt and transfer' order (Class 0) is about to alter (CC). If the switch is locked down, the 'halt' is not observed, and the transfer will be made in the same way as a 'transfer' code. If the switch is neutral, the machine will halt in executing OPN (Field 2) and the continue light will be turned on. The normal cycle is re-entered when the continue button is pressed. If the switch is locked up, the machine will halt and sound a bell.

<u>Trap Lights</u>: The four trap lights and the trapping light, switch and push button are effective only when a trap transfer is about to take place. If the trap switch is locked down, the transfer will be made immediately. If the trap switch is neutral, a halt will occur with the

trapping light and one of the four trap lights on; at this point, pressing the trap button will cause calculation to proceed. The trap light which is on indicates the point of the instruction cycle at which trapping has occurred, i.e.,

Trap 1. After loading I (Control Tag Trap)

Trap 2. After loading S (Arithmetic Tag Trap)

Trap 3. After executing AUXILIARY (Control Tag Trap)

Trap 4. After executing AUXILIARY (Arithmetic Status Trap).

Console Displays

At the right side of the control panel are three groups of lights which display the contents of the registers U, I and CC. Associated with I is the two bit control tag register (CTR) display.

A fourth display of nineteen lights indicates the position of the typewriter carriage relative to a machine register during a type-in or type-out operation.

Printer

There are two push-buttons on the printer. The upper one, when held down, will restore paper to a new page. The lower one will feed paper as long as it is held down. All printer operations stop automatically when the end of the paper is sensed.

Punch

There is a single push-button on the punch, which causes tape to be punched continuously with 'tape feed' characters as long as it is held down.