

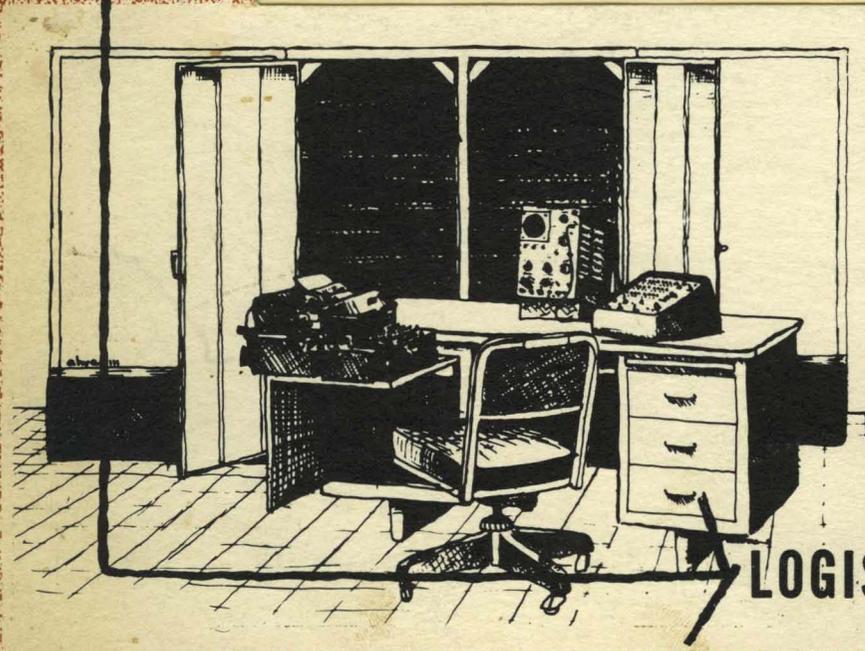
alwac III-

DATA PROCESSING SYSTEMS

DESCRIPTION OF OPERATIONS

ALWAC III

ELECTRONIC DIGITAL COMPUTER



LOGISTICS RESEARCH INC.

141 so. pacific ave. redondo beach, california

ALWAC III

GENERAL PURPOSE COMPUTER



LOGISTICS RESEARCH, INC.
REDONDO BEACH, CALIF.

DESCRIPTION OF OPERATIONS

ALWAC III

ELECTRONIC DIGITAL COMPUTER

DESCRIPTION OF ALWAC III OPERATIONS

The arithmetic registers of the ALWAC computer are circulating loops of information one word in length. The information is changed when the magnetic writing head is switched to an input device or to some reading head other than the one which makes up the loop with it. In addition to four one-word registers, the drum has four 32-word channels of working storage from which instructions are picked up to be carried out and data picked up to be operated on in the arithmetic registers. The working storage, through what is essentially a loop arrangement, has faster access than the remainder of the drum, the main storage.

Main storage consists of 128 channels, 32 words in each. Any main-storage channel may be copied in a block to one of the working-storage channels, and a working-storage channel may be copied in a block to any main channel. Working-storage channels are designated by roman numerals and main-storage channels by a pair of hexadecimal characters. As notation for the integers ten through fifteen, the addressing system of the ALWAC borrows the first six letters of the alphabet, and the digits used are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, a, b, c, d, e, and f. The decimal value of any hexadecimal pair is obtained by adding the second digit to the product of the first digit and sixteen. The value of c9, for example, is 201 decimal, or 9 plus the product of 12 (or c) and 16, which is 192. The integer following c9 is ca and the integer following 7f is 80. The diagram below shows the entire array of two-digit hexadecimal integers, some of which are used in ALWAC in more than one way.

00	01	02	03	04	05	06	07	40	41	42	43	44	45	46	47
08	09	0a	0b	0c	0d	0e	0f	48	49	4a	4b	4c	4d	4e	4f
10	11	12	13	14	15	16	17	50	51	52	53	54	55	56	57
18	19	1a	1b	1c	1d	1e	1f	58	59	5a	5b	5c	5d	5e	5f
20	21	22	23	24	25	26	27	60	61	62	63	64	65	66	67
28	29	2a	2b	2c	2d	2e	2f	68	69	6a	6b	6c	6d	6e	6f
30	31	32	33	34	35	36	37	70	71	72	73	74	75	76	77
38	39	3a	3b	3c	3d	3e	3f	78	79	7a	7b	7c	7d	7e	7f
80	81	82	83	84	85	86	87	c0	c1	c2	c3	c4	c5	c6	c7
88	89	8a	8b	8c	8d	8e	8f	c8	c9	ca	cb	cc	cd	ce	cf
90	91	92	93	94	95	96	97	d0	d1	d2	d3	d4	d5	d6	d7
98	99	9a	9b	9c	9d	9e	9f	d8	d9	da	db	dc	dd	de	df
a0	a1	a2	a3	a4	a5	a6	a7	e0	e1	e2	e3	e4	e5	e6	e7
a8	a9	aa	ab	ac	ad	ae	af	e8	e9	ea	eb	ec	ed	ee	ef
b0	b1	b2	b3	b4	b5	b6	b7	f0	f1	f2	f3	f4	f5	f6	f7
b8	b9	ba	bb	bc	bd	be	bf	f8	f9	fa	fb	fc	fd	fe	ff

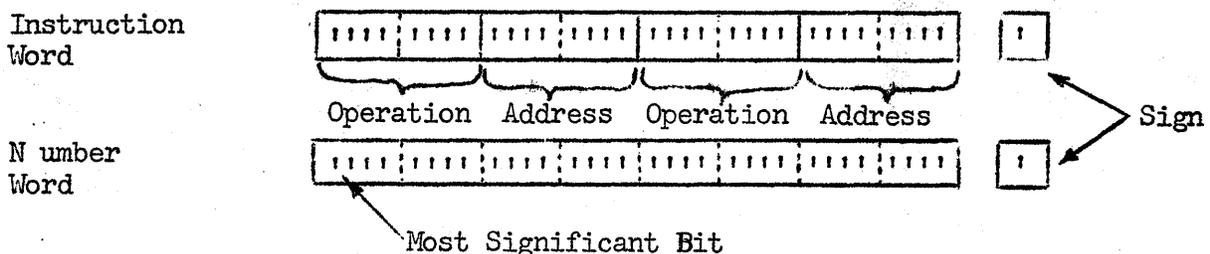
Since block-copy operations always transfer 32 words between working storage and main storage, it is not necessary to address individual words in main storage, and the hundred and twenty-eight channels are numbered 00 through 7f. In working storage, however, it is necessary to address not only words, but half-words, since each instruction occupies a half-word. In each working-storage channel the sixty-four half-words are numbered consecutively, 00 through 3f for channel I, 40 through 7f for channel II, 80 through bf for channel III, and c0 through ff for channel IV.

Instructions are picked up to be carried out in the order shown by reading down the half-word address columns of the diagram. For channel I, 00, 08, 10, 18, 20, 28, 30, 38, 01, 09, etc. For channel IV, c0, c8, d0, d8, e0, e8, f0, f8, c1, c9, etc.

The ALWAC III word may be put into the computer or typed out as eight hexadecimal digits. From left to right the normal arrangement is operation, address, operation, address. The pairs of digits are referred to as syllables. Since the operation codes are two digits, it would be possible with the present logic to wire two hundred and fifty-six different operations into the computer. Some eighty two-digit operation codes are actually used.

The normal instruction, consisting of operation and address, is four digits, but since some operations do not require addresses, they may be doubled up to yield three or four instructions per word. To do this it is necessary to modify the operation codes appearing in the first and third syllables. The codes which can be so modified are listed with an asterisk in the following description. They are increased by 1 whenever the second or fourth syllable is to be used for an operation code. For example, the operations 22, 3e, 2c, and 30 can be packed into one word, in which case they are written 23, 3e, 2d, and 30. The code which is increased must be one of those with an asterisk. The second code can be any of those for the no-address operations, whether marked with an asterisk or not (for example, e0).

As displayed on the oscilloscope the hexadecimal digits in registers appear, of course, as groups of four binary digits. The ALWAC word has a magnitude of 32 bits, and one bit is used for algebraic sign, which is 0 for minus, 1 for plus.



In the following description the sign bit is referred to as the sign bit or sign and the most significant bit is referred to as the left-hand bit of the word. The word "digit" is used for groups of four bits translated by the typewriter to or from hexadecimal notation, or for the decimal digits of input and output.

The A register and B register together are a double-length accumulator. B is the multiplicand register and the D register holds the multiplier or the divisor. AB holds the double-length product. AB or B is the dividend. B is the quotient register, A the remainder. E is a tally and test-for-jump register. The symbol W is used for any word drawn from working-storage. W and the addresses of main storage channels always occupy the address syllable of the instruction that uses them. This syllable, however, is not always a true address. In a0, a2, a4, a6, and a8, the second and fourth syllables are used to determine how many binary places the register is to be shifted. The address syllable has similar uses as a control indicator or tally in input-output operations.

- 22* ROUND OFF Round off the double length number AB to a single length number in A: if the left-hand bit in B is not zero, add 1 to A. B is left unchanged.

- 28* CLEAR A Change the number in A to zero and make the sign bit positive (1).

- 2c* POSITIVE A Make the sign bit of A positive (1).

- 2e* REVERSE A SIGN Reverse the sign of the number in A.

- 3e* COMPLEMENT A Complement the number in A, including the sign bit. (Change all 1's to 0, all 0's to 1).

The six shift operations in the ALWAC shift the number in A or AB, and the address digits indicate how many binary places the number is to be shifted, up to thirty-one. If the address is thirty-two, which is 20 hexadecimal, the result is zero. Of any digits used in the address, only the right-hand five bits will have any effect. This is an example of the "modular" interpretation of numbers. Of N binary digits, if only the right-hand six are effective, the "modulus" is sixty-four (value of the seventh bit from the right) and the N bits are said to be interpreted "modulo sixty-four."

a0 DOUBLE SHIFT RIGHT

Shift AB, excluding signs, N bits to the right, where N is the address syllable. Bits leaving A enter the left end of B. If overflow is on, the N left bits of A are filled with 1's; if it is off, they are filled with 0's. N bits are dropped from the right end of B.

a2 DOUBLE SHIFT LEFT

Shift AB, excluding signs, N bits to the left, where N is the address syllable. Bits leaving B enter the right end of A. The right N bits of B are filled with 0's and N bits are dropped from the left end of A.

a4 SHIFT RIGHT

Shift A, excluding the sign, N bits to the right, where N is the address syllable. If overflow is on, the left N bits of A are filled with 1's; if it is off, they are filled with 0's, and the right N bits are dropped.

a6 SHIFT LEFT

Shift A, excluding the sign, N bits to the left, where N is the address syllable. The right N bits of A are filled with 0's and the left N bits are dropped.

a8, a9 SCALE

a8: Shift A, excluding the sign, N bits to the right, where N is the address following a8 (which was copied from the A register by a 4c instruction). The left-hand N bits of A are filled with 0's regardless of whether the overflow is on or off. Make A sign equal to B sign.

a9: Shift A, excluding the sign, N bits to the left, where N is the address following a9. The right-hand N bits of A are filled with 0's. Make A sign equal to B sign.

The storage address of an a8 operation usually appears as the address syllable with a preceding 4c instruction. The 4c instruction copies the sign of the A register into the right-hand bit of the a8 syllable and alters the following address syllable to agree with corresponding digits in the A register.

If A is negative (sign-bit 0) in the 4c operation the a8 instruction is left unchanged in storage. If A is positive, the a8 instruction is changed to a9.

aa FLOAT

If A is not zero, shift AB to the right until A is zero and the left bit of B is 1. If A is zero, shift AB to the left until the left bit of B is a 1. Count in the D register the number of bits AB was shifted, and give D a minus sign if the shift was left, a plus sign if it was right. If A and B are both zero, turn on the overflow and clear the count from the D register. Make the sign of A equal to the sign of B.

In the description of all arithmetic and information-transfer operations, W refers to the word or half-word in the working-storage cell specified by the address syllable of the instruction.

- | | | |
|----|----------------|--|
| 60 | ADD | Add W to A and put the sum in A. If the sum exceeds the capacity of A, turn on the overflow. |
| 62 | MINUS ADD | Add W to A, put the sum in A, and reverse the sign of A. If the sum exceeds the capacity of A, turn on the overflow. |
| 64 | MINUS SUBTRACT | Subtract W from A, put the difference in A, and reverse the sign of A. If the difference exceeds the capacity of A, turn on the overflow. |
| 66 | SUBTRACT | Subtract W from A and put the difference in A. If the difference exceeds the capacity of A, turn on the overflow. |
| bc | LONG ADD | Make the sign of A equal to the sign of B. Add W to AB and put the sum in AB, with the sign of the sum in both A and B. |
| be | LONG SUBTRACT | Make the sign of A equal to the sign of B. Subtract W from AB and put the difference in AB, with the sign of the difference in both A and B. |

Certain operations cannot be carried out while the overflow is on, and the appearance of their codes in a routine will cause the computer to stop and give an alarm, leaving registers unchanged. They are OVERFLOW IF A SMALLER and the following add, subtract, multiply, and divide operations: 60, 62, 64, 66, e0, e2, e4, e6, e8, ea, ec, ee.

- | | | |
|----|-------------------|--|
| e0 | ADD MULTIPLY BY D | Give A and B the sign of the product, multiply B and D, add A to the less significant half of the product, and put the result in AB. D is unchanged. |
| e2 | ADD MULTIPLY | Give A and B the sign of the product, multiply B and W, add A to the less significant half of the product, and put the result in AB. D now contains W. |
| e4 | MULTIPLY BY D | Give A and B the sign of the product, multiply B and D, and put the product in AB. D is unchanged. |
| e6 | MULTIPLY | Give A and B the sign of the product, multiply B and W, and put the product in AB. D now contains W. |

In the four divide operations, the computer makes a pre-comparison of dividend and divisor to see whether the quotient will exceed the capacity of the registers. If it will, the division is not performed and the overflow is turned on. The sign of B is used for the sign of the dividend.

- | | | |
|----|------------------|---|
| e8 | LONG DIVIDE BY D | Divide AB by D. Put the quotient, with the proper sign, in B, and the remainder in A with the sign of the dividend. D is unchanged. |
| ea | LONG DIVIDE | Divide AB by W. Put the quotient, with the proper sign, in B, and the remainder in A with the sign of the dividend. D now contains W. |
| ec | DIVIDE BY D | Divide B by D. Put the quotient, with the proper sign, in B, and the remainder in A with the sign of the dividend. D is unchanged. |
| ee | DIVIDE | Divide B by W. Put the quotient, with the proper sign, in B, and the remainder in A with the sign of the dividend. D now contains W. |

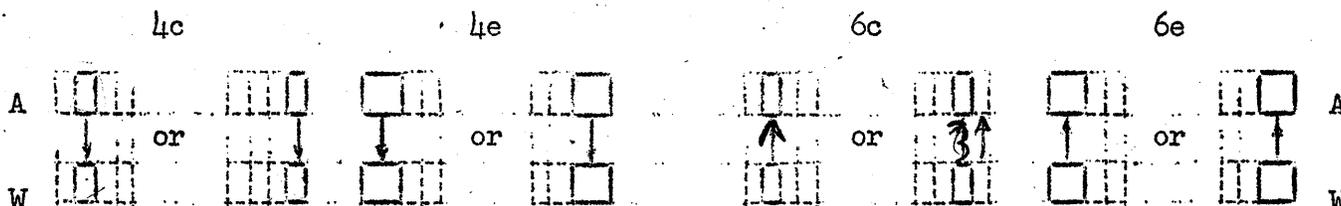
In the description of the seven jump operations, W refers to the half-word (operation or operation and address) in the working-storage cell specified by the address syllable in the jump instruction.

10	JUMP	Carry out W as the next operation.
12	CONTROL JUMP 1	If jump switch 1 on the control panel is ON, carry out W next; if it is OFF, continue in standard sequence of instructions.
14	CONTROL JUMP 2	If jump switch 2 on the control panel is ON, carry out W next; if it is OFF, continue in the standard sequence of instructions.
16	COUNT DOWN	Subtract 1 from the absolute value of E. If the result is not zero, carry out W next; if the result is zero, continue in the standard sequence of instructions, and leave $E = 2^{32} - 1$ with its original sign.
18	NON-ZERO JUMP	If A is not zero, carry out W next; if A is zero, continue in the standard sequence of instructions.
1c	LESS THAN ZERO JUMP	If A is less than zero, carry out W next; if A is minus zero, plus zero, or a positive quantity greater than zero, continue in the standard sequence of instructions.
1e	OVERFLOW JUMP	If the overflow is on, carry out W next and turn off overflow; if it is off, continue in the standard sequence of instructions.
50	OVERFLOW IF <u>A</u> SMALLER	If the overflow is on, stop and give an alarm. If the overflow is off, and A is less in absolute value than W, turn on the overflow.
02*	REVERSE OVERFLOW	If the overflow is on, turn it off; if it is off, turn it on.

In all copy operations the information remains in the location from which it was copied.

30*	EXCHANGE A AND B	40	COPY W to B
36*	EXCHANGE A AND E	48	COPY A to W
3a*	EXCHANGE A AND D	4c	COPY ADDRESS to W
68	EXCHANGE A AND W	4e	COPY HALF to W
08*	COPY E to B	56	COPY W to E
0a*	COPY E to D	5a	COPY W to D
32*	COPY B to A	6c	COPY ADDRESS to A
34*	COPY E to A	6e	COPY HALF to A
38*	COPY D to A	b4	COPY MAIN to A
		78	COPY W to A

In 4c, 4e, 6c, and 6e, W is a syllable (address) or two syllables (operation and address) in a working storage channel. Copying proceeds between W and the corresponding syllable or syllables of A.



In all cases, if W is even, the first half of A is involved, if W is odd, the second half of A. Besides copying the address, 4c copies the sign of A into the last bit of the operation-code in W.

In b4, the address syllable of the instruction refers to a word in channel 00 of main storage. This channel is reserved for fast access to data words. The address syllable with b4 may consist of any one of the sixty-four two-digit hexadecimal integers, but any odd address will draw out the word whose first half is addressed by the next lower even number.

At least two drum rotations, or 34 milliseconds, must be allowed between the b4 operation and any preceding BLOCK COPY operation except one which loads channel 00 (8800, 8a00, 8c00, or 8e00). This allows time for head-switching relays to drop out.

70 CONTROLLED COPY

```

A 11001100
D 10101010
W 00110001

A 01100100

```

Wherever the D register contains a 1 bit, change the corresponding bit of A to agree with the bit in W. Wherever the D register contains a 0 bit, do not change the corresponding bit in A.

74 **EXTRACT**

```

A 10110010
W 10100101
A 10100000

```

Wherever W contains a 0 bit, make the corresponding bit in A 0. Wherever W contains the 1 bit, do not change the corresponding bit in A.

80 **COPY to I**
 82 **COPY to II**
 84 **COPY to III**
 86 **COPY to IV**

88 **COPY FROM I**
 8a **COPY FROM II**
 8c **COPY FROM III**
 8e **COPY FROM IV**

In 80 through 8e, each of which transfers a block of 32 words between a working channel and one of the main-storage channels, the address syllable refers to one of the main channels. The second digit of the operation code determines whether the transfer is into or out of a working channel.

Operation code 00 is a blank which will simply be ignored during instruction pickup. Therefore, data words which have zeros in the operation-code positions may be inserted anywhere in a routine without the necessity of instructions to skip them.

f0 **CODES IN**

Shift A left four binary places and copy into the four right-hand bits the digit received from tape or keyboard. Perform this operation N times, where N is the address syllable following f0 (N= 0 is the signal for eight-digit input on this operation).

f2 **LETTERS IN**

Shift A left six binary places and copy into the seventh through second right-hand bits the code symbol received from tape or typewriter. Perform this operation N times, where N is the address syllable following f2. (0 < N ≤ 5.)

f4 **CODES OUT**

Transfer the four left-hand bits of A to tape or keyboard and shift A left four binary places. Perform this operation N times, where N is the address syllable following f4 (N=0 is the signal for eight-digit output on this operation).

f6	LETTERS OUT	Transfer the second through seventh left-hand bits of A to tape or keyboard and shift A left six binary places. Perform this operation N times, where N is the address syllable following f6. ($0 < N \leq 5$).
d8	SIGN OUT	If the sign of A is 1 (+), transmit a space symbol to tape or keyboard; if the sign of A is 0 (=), transmit a symbol for minus sign to tape or keyboard.
dc	NUMBER OUT	Transfer the four right-hand bits of A to tape or keyboard. Used with binary to decimal conversion, this operation may type out as many as 10 decimal digits for one computer word.
9a	TYPE	As d and f series operations occur, type out unless overruled by typewriter switch settings.
9c	PUNCH	As d and f series operations occur, punch out unless overruled by typewriter switch settings.
9e	BOTH	As d and f series operations occur, type and punch out unless overruled by typewriter switch settings.
98	NEITHER	As d and f series operations occur, do nothing unless some action is called for by typewriter switch settings.

Several functional routines are put into the ALWAC during its test and inspection phase. They are the FILL, COPY, CORRECTION, AND START routine, the DECIMAL INPUT routine and the DECIMAL OUTPUT routine. All make use of the concept of "key" words, in which typewriter inputs in a certain order are interpreted as different kinds of data -- for example, as a main-channel address for BLOCK COPY, as the address for a JUMP operation, etc.

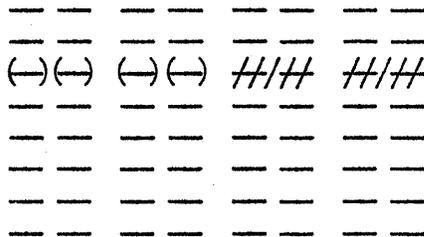
As its name indicates, the FILL, COPY, CORRECTION, AND START routine takes care of the operations of filling a routine which has been punched on paper tape, copying it out of the memory for checking, making any corrections found to be necessary, and performing the JUMP or START operation necessary to run the routine.

The DECIMAL INPUT and OUTPUT routines provide for inserting decimal numbers and transferring them in binary form to their proper places for computation, and printing out binary results in decimal form, with provisions in both cases for setting the number of digits following the decimal point and for certain programmed accuracy checks.

ALMAC III
OPTIMUM LOCATION OF OPERANDS FOR FAST ACCESS

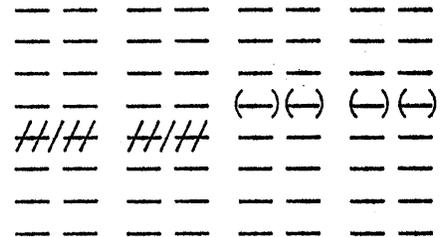
(←)(←) = Instructions

= Data Word



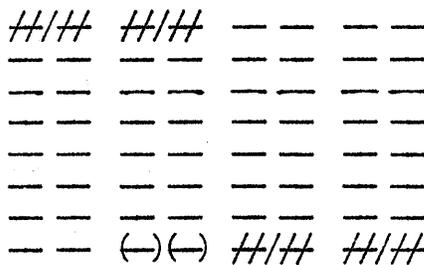
General Case, Instruction
At Left

For an instruction on the left half of the coding sheet, the optimum data locations are those on the same line on the right half of the sheet, and the two corresponding locations in each of the three other working channels.



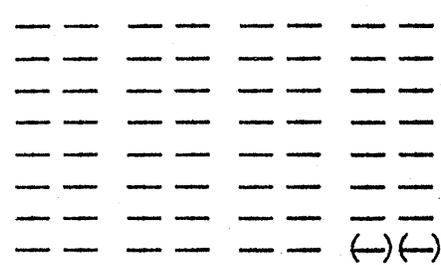
General Case, Instruction
At Right

For an instruction on the right half of the coding sheet, the optimum data locations are those on the next lower line on the left half of the sheet, and the corresponding locations in each of the three other working channels.



Special Case, Last Instruc-
tion on Left Half

The last instruction on the left half of the coding sheet has four optimum data locations, those on the same line on the right half and those on the top line on the left half of the sheet.



Special Case, Last Instruc-
tion on Right Half

The last instruction on the right half of the coding sheet has no optimum data locations.

CHANNEL II. PROGRAMMING FORM

LOGISTICS RESEARCH INC.

ROUTINE _____ CHANNEL _____ BY _____ DATE _____ CHK. _____ NO. _____

NOTES: _____

40	41	42	43	44	45	46	47
48	49	4a	4b	4c	4d	4e	4f
50	51	52	53	54	55	56	57
58	59	5a	5b	5c	5d	5e	5f
60	61	62	63	64	65	66	67
68	69	6a	6b	6c	6d	6e	6f
70	71	72	73	74	75	76	77
78	79	7a	7b	7c	7d	7e	7f

ROUTINE _____ CHANNEL _____ BY _____ DATE _____ CHK. _____ NO. _____

NOTES: _____

80	81

82	83

84	85

86	87

88	89

8a	8b

8c	8d

8e	8f

90	91

92	93

94	95

96	97

98	99

9a	9b

9c	9d

9e	9f

a0	a1

a2	a3

a4	a5

a6	a7

a8	a9

aa	ab

ac	ad

ae	af

b0	b1

b2	b3

b4	b5

b6	b7

b8	b9

ba	bb

bc	bd

be	bf

LOGISTICS RESEARCH INC. REDONDO BEACH, CALIF.

LOGISTICS APPLICATION ENGINEERING

Experienced Logistics Application Engineers are at your service . . . to survey your data-handling and computational requirements . . . and to show you exactly how ALWAC can serve you. Whether your field is science, business, government, or engineering . . . Logistics can help.