LINC III ORDER CODE

Mary Allen Wilkes

4 June 1963

I. MSC CLASS INSTRUCTIONS

HLT 0000 HLT

HALT. The computer halts: the Run light on the control console goes off, and the gong rings.

CLR 0011 8 μsec CLR

CLEAR. Clear the Accumulator and the Link bit. 0 → C (ACC); 0 → C(L).

MSC 13 0013 8 μsec MSC 13

WRITE GATE ON. The write-gate for marking tapes is turned on if, and only if, the Mark push button on the console has been depressed. The instruction is only used when generating LINC tapes.

ATR 0014 8 μsec ATR

ACCUMULATOR TO RELAY. The contents of the right half (bits 0-5) of the Accumulator replace the contents of the relay register. The Accumulator is left unchanged.

RTA 0015 8 μsec RTA

RELAY TO ACCUMULATOR. The contents of the relay register replace the contents of the right half (bits 0-5) of the Accumulator. The left half of the Accumulator is cleared. The relay register is left unchanged.

NOP 0016 8 μsec NOP

NO OPERATION. This instruction provides a delay of 8 μsecs. before proceeding to the next instruction. It does nothing.

COM 0017 8 μsec COM

COMPLEMENT. Complement the number in the Accumulator. C(ACC) → C(ACC).

II. SHIFT CLASS INSTRUCTIONS

ROL i n 240 + 20i + n 16 μsec + ROL

ROTATE LEFT. If i = 0, shift contents of the Accumulator n places to the left (n = 0, 1, ..., 17 octal), with bit 11 feeding bit 0. If i = 1, shift the Link - Accumulator combination n places to the left, with bit 11 feeding the Link bit and the Link bit feeding bit 0.

Time of execution: 16 μsec for n = 0, 1, 2, 3; 8 μsec additional for each additional 4 places or fraction thereof.

ROR i n 300 + 20i + n 16 μsec + ROR

ROTATE RIGHT. If i = 0, shift contents of the Accumulator n places to the right (n = 0, 1, ..., 17 octal), with bit 0 feeding bit 11. If i = 1, shift the Link - Accumulator combination n places to the right, with bit 0 feeding the Link bit and the Link bit feeding bit 11.

Time of execution: 16 μsec for n = 0, 1, 2, 3; 8 μsec additional for each additional 4 places or fraction thereof.

SCR i n 340 + 20i + n 16 μsec + SCR

SCALE RIGHT. If i = 0, shift the contents of the Accumulator n places to the right (n = 0, 1, ..., 17 octal), with bit 11 (the sign bit) unchanged and bits shifted out of bit 0 lost. If i = 1, shift the Accumulator as above; the last bit shifted out of bit 0 will be saved in the Link bit.

Time of execution: 16 μsec for n = 0, 1, 2, 3; 8 μsec additional for each additional 4 places or fraction thereof.

III. FULL-ADDRESS CLASS INSTURCTIONS

ADD X	2000 + X	16 μsec	ADD

ADD. Add the contents of memory register X ($0 \leq X \leq 1777$) to the contents of the Accumulator, leaving the result in the Accumulator, C(X) + C(ACC) → C(ACC), using 12-bit binary addition with end-around-carry. Register X is unchanged.

STC X	4000 + X	16 μsec	STC

STORE-CLEAR. Copy the contents of the Accumulator into memory register X($0 \leq X \leq 1777$) and then clear the Accumulator. C(ACC) → C(X), 0 → C(ACC).

JMP X	6000 + X	16 μsec*	JMP

JUMP. Take the next instruction from memory register X ($0 \leq X \leq 1777$) and continue to execute instructions in sequence starting with register X. The address X replaces the contents of the Program Counter and, unless X = 0, the original contents of the Program Counter increased by 1 and prefixed by the code for JMP replace the contents of memory register 0. If C(PC) = p, then X → C(PC) and JMP p+1 → C(0) unless X = 0. If X = 0, then 0 → C(PC).

*Execution time: For X = 0, 8 μsec; for X ≠ 0, 16 μsec.

IV. INDEX CLASS INSTURCTIONS

TABLE I. ADDRESSING IN INDEX CLASS INSTRUCTIONS

i	β	Y	t μsec	Comment
0	0	X(p + 1)	16	
1	0	p + 1	8	
0	$1 \leq \beta \leq 17$	X(β)	16	
1	$1 \leq \beta < 17$	X(β) + 1	16	X(β) + 1→X(β)

The time t μsec must be added to the execution time to get the total instruction time. Y is the address of the register which holds the operand used by the instruction. The instruction is assumed to be in register p.

LDA i β 1000 + 20i + β (8 + t) μsec* LDA

LOAD ACCUMULATOR. Copy the contents of memory register Y (*see Table I) into the Accumulator. C(Y) → C(ACC). Register Y is unchanged.

STA i β 1040 + 20i + β (8 + t) μsec* STA

STORE ACCUMULATOR. Copy the contents of the Accumulator into memory register Y (*see Table I). C(ACC) → C(Y). The Accumulator is unchanged.

ADA i β 1100 + 20i + β (8 + t) μsec* ADA

ADD TO ACCUMULATOR. Add the contents of memory register Y (*see Table I) to the contents of the Accumulator, leaving the result in the Accumulator. C(Y) + C(ACC) → C(ACC), using 12-bit binary addition with end-around-carry. Register Y is unchanged.

ADM i β 1140 + 20i + β (16 + t) μsec* ADM

ADD TO MEMORY. Add the contents of memory register Y (*see Table I) to the contents of the Accumulator, leaving the result in the Accumulator and in register Y. C(Y) + C(ACC) → C(ACC) and → C(Y), using 12-bit binary addition with end-around-carry.

LAM i β 1200 + 20i + β (16 + t) μsec* LAM

LINK-ADD TO MEMORY. First add the contents of the Link bit (the integer 0 or 1) to the contents of the Accumulator leaving the sum in the Accumulator, using 12-bit binary addition with the end-carry, if any, replacing the contents of the Link bit; (if no end-carry arises, clear the Link bit). Next add the contents of register Y (*see Table I) to the contents of the Accumulator with the end-carry, if any, replacing the contents of the Link bit (if no end-carry arises, the Link bit is unchanged), leaving the 12-bit result in the Accumulator and in register Y.

MUL i β 1240 + 20i + β (104 + t) μsec* MUL

MULTIPLY. Multiply the contents of register Y (*see Table I) by the contents of the Accumulator, and leave the result in the Accumulator. The values in Y and in the Accumulator are treated as signed, 11-bit ones' complement numbers, and are multiplied together to form a 22-bit product. They may be interpreted as either fractions or integers: if bit 11 (the h-bit) of the address Y is a one, they are treated as fractions whose binary points are between bit 11 (the sign bit) and bit 10. In this case the most significant 11 bits of the 22-bit product are left with the proper sign in the Accumulator. If bit 11 of the address Y is a zero, the values are treated as integers, and the least significant 11 bits of the 22-bit product are left with the proper sign in the Accumulator. When i = 1 and β = 0, bit 11 of the address Y is assumed to be zero, and the values are treated as integers. Register Y is unchanged.

SAE i β 1440 + 20i + β (8 + t) μsec* SAE

SKIP IF ACCUMULATOR EQUALS. If the contents of the Accumulator exactly match the contents of memory register Y (*see Table I) then skip the next instruction (actually, skip the first register of the next instruction). Otherwise, go on to the next instruction in sequence. C(ACC) and C(Y) are unchanged in either case.

SRO i β | 1500 + 20i + β | (8 + t) μsec* | SRO

SKIP AND ROTATE. If the rightmost bit of the contents of memory register Y (*see Table I) is a zero, then skip the next instruction (actually, skip the first register of the next instruction). Otherwise, go on to the next instruction in sequence. In either case, rotate the contents of register Y one place to the right and replace in register Y. The Accumulator is unaffected.

BCL i β | 1540 + 20i + β | (8 + t) μsec* | BCL

BIT CLEAR. For each bit of the contents of register Y (*see Table I) which is a one, clear the corresponding bit of the contents of the Accumulator. Register Y and other bits in the Accumulator are unaffected.

BCO i β | 1640 + 20i + β | (8 + t) μsec* | BCO

BIT COMPLEMENT. For each bit of the contents of register Y (*see Table I) which is a one, complement the corresponding bit of the contents of the Accumulator. Register Y and other bits in the Accumulator are unaffected.

BSE i β | 1600 + 20i + B | (8 + t) μsec* | BSE

BIT SET. For each bit of the contents of register Y (*see Table I) which is a one, set to one the corresponding bit of the contents of the Accumulator. Register Y and other bits in the Accumulator are unaffected.

DSC i β	1740 + 20i + β	(112 + t) μsec*	DSC

DISPLAY CHARACTER. Display, in a 2 x 6 grid, the pattern contained in register Y (*see Table I). The contents of register Y are examined from right to left beginning with bit zero, and for each bit found to be a one a point is displayed. The initial X-coordinate will be the contents of register i, plus 4; the display channel is selected by the leftmost bit of register i. The initial Y-coordinate will be the contents of the Accumulator with the rightmost 5 bits (bits 0-4) set to zero by the computer. The initial coordinates specify the lower left position of the display; the computer proceeds from lower left to upper right. For each bit of register Y which is examined, +4 is added to the Y-coordinate in the Accumulator. When the right 6 bits of register Y have been examined, the right 5 bits of the Accumulator are reset to zero and +4 is added to the X-coordinate in register i. The procedure is then repeated for the left 6 bits of register Y. At the conclusion of the instruction the contents of register i have been incremented by 10 (octal), and the right 5 bits of the Accumulator are left equal to 30 (octal). Register Y is unchanged.

V. HALF-WORD CLASS INSTRUCTIONS

Table II ADDRESSING IN HALF-WORD CLASS INSTRUCTIONS

i	β	Y	h	t μsec	Comment
0	0	X(p + 1)	h(p + 1)	16	If h=0, operand is LH(X) If h=1, operand is RH(X)
1	0	p + 1	0	8	Operand is always LH(p + 1)
0	1≤β≤17	X(β)	h(β)	16	If h=0, operand is LH(X) If h=1, operand is RH(X)
1	1≤β≤17	X(β) + h(β)	h(β)	16	If h=0, operand is LH(X+1) If h=1, operand is RH(X) h, j, X + h → C(β)

The time t μsec must be added to the execution time to get the total instruction time. Y is the address of the register holding the operand in the half designated by h. (h = 1: right half; h = 0: left half). The instruction is assumed to be in register p.

LDH i β 1300 + 20i + β (8 + t) μsec* LDH

LOAD HALF. Copy the contents of the designated half of register Y (*see Table II) into the right half of the Accumulator, clearing the left half of the Accumulator. Register Y is unchanged.

STH i β 1340 + 20i + β (8 + t) μsec* STH

STORE HALF. Copy the contents of the right half of the Accumulator into the designated half of register Y (*see Table II). The Accumulator and the unused half of register Y are unchanged.

SHD i β 1400 + 20i + β (8 + t) μsec* SHD

SKIP IF HALF DIFFERS. If the contents of the right half of the Accumulator differ from the contents of the designated half of register Y (*see Table II) then skip the next instruction (actually, skip the first register of the next instruction). Otherwise, go on to the next instruction in sequence. C(ACC) and C(Y) are unchanged in either case.

VI. MISCELLANEOUS INSTRUCTIONS

TABLE III. ADDRESSING IN SET INSTRUCTION

i	n	Y	t μsec
0	$0 \leq n \leq 17$	X(p + 1)	8
1	$0 \leq n \leq 17$	p + 1	0

SET i n — 40 + 20i + n — (24 + t) μsec* — SET

SET. Set register n equal to the contents of register Y (*see Table III). The Accumulator and register Y are unchanged. Take the next instruction from p + 2.

SAM i n — 100 + 20i + n — 8 μsec / 24 μsec — SAM

SAMPLE. Sample the signal on one of 16 input channels selected by n. Put its binary conversion, ±177, into the Accumulator, extending the sign through bit 11. $0 \leq n \leq 7$ selects one of the potentiometers; $10 \leq n \leq 17$ selects one of the high speed inputs. If i = 0, the instruction requires 24 μsec. If i = 1, the computer goes on to the next instruction after 8 μsec; the conversion process in the Accumulator continues, however, for an additional 14 μsec. Therefore, care should be taken when instructions which affect the Accumulator follow a SAM with i = 1.

DIS i n — 140 + 20i + n — 32 μsec — DIS

DISPLAY. When i = 1, index the address part of register n (n = 0,1,..., 17 octal by 1. Display one point whose X coordinate is specified by the rightmost 9 bits of register n, and whose Y coordinate, ±377 octal, is specified by the contents of the Accumulator. Display via the channel selected by the leftmost bit of register n. The Accumulator remains unchanged.

XSK i n — 200 + 20i + n — 16 μsec — XSK

INDEX AND SKIP. If i = 1, increment the address part of register n by 1. If i = 0, do not index register n. In either case, skip the next instruction (actually the first register of the next instruction) if the address part of register n equals 1777. If it does not equal 1777, go to the next instruction in sequence. The Accumulator is unchanged.

OPR i n	500 + 20i + n	16 μsec*	OPR

OPERATE. The Operate instruction is a multi-purpose input-output instruction wich is used to:

1) transfer information from the LINC keyboard and the control console toggle switches (Right Switches and Left Switches) to the LINC Accumulator.
2) control the transfer of digital information between the LINC and external digital devices.
3) provide pulses which may be used externally to synchronize or control special equipment.

Toggle Switch Input. When n = 16, the contents of the Right Switches replace the contents of the Accumulator. When n = 17, the contents of the Left Switches replace the contents of the Accumulator. In these cases the i-bit has no effect. *Time of Execution: 16 μsec.

Pausing. For $0 \leq n \leq 15$ the i-bit provides a timing control generally used to synchronize the LINC with external devices. When i = 1 the computer will pause. It will remain in an inactive state until it receives a "restart" signal (-3 volts) from the external equipment. The n-bits ($0 \leq n \leq 15$) of the instruction designate the external level input line to be used for restart. If i = 0, or if the restart signal is already present on line n, the computer will not pause. In this case it is assumed that the external equipment is ready for restart at the time the computer would normally pause.

Pulse Output. During the execution of an OPR instruction, four pulses of -3 volts are available to external equipment. The first of these is a long pulse which appears 4 μsec after the beginning of the instruction on one of 16 pulse output lines provided at the LINC's Data Terminal Box. The output line is designated by n ($0 \leq n \leq 17$); minimum duration of this pulse is 12 μsec. If the computer pauses, the pulse duration is extended by the length of the pause.

The other 3 pulses each .4 μsec duration, are associated with pause and restart. One is delivered to external equipment at pause time if, and only if, the computer actually pauses. The second occurs at the time when the computer would normally resume operation after a pause, regardless of whether

the computer has actually paused or not. If the computer has paused, the pulse, which indicates that the computer is now running, will appear not less than 2 μsec and not more than 4 μsec after the restart signal has been delivered by the external equipment. If the computer has not paused, this pulse will appear 2 μsec after pause time. The third pulse appears 2 μsec later.

Keyboard Input. When n = 15, the Accumulator is cleared and the 6-bit code number for a struck key is transferred into the right half of the Accumulator; the key is released. If i = 1, the computer waits for a key to be struck; if i = 0, or if a key was previously struck, the computer does not wait. When a key is struck, the keyboard locks until a KBD instruction is executed. *Time of Execution: 16 μsec. when no pause.

Digital Input-Output. The OPR instruction may be used to transfer 12-bit digital information between the LINC and external devices. The user may choose to transfer one word, into the LINC Accumulator each time the OPR instruction is executed, or he may use the instruction to transfer a group of words between the LINC memory and his external device. In this context the pause feature and the LINC's output pulses would be used to synchronize external equipment with the computer.

Digital Input to Accumulator. There are four 12-bit channels available for single-word input to the LINC Accumulator. Two, SN and TN, provide direct input to the Accumulator, and two, UN and VN, provide input via the B Register to the Accumulator. One word is transferred each time the Operate instruction is executed. When information is ready to be transferred, the external equipment must supply an enabling level for the appropriate channel (SNEL, TNEL, UNEL, or VNEL), followed, if the computer is paused, by the restart signal on line n. After restart, if the transfer is into the Accumulator via SN or TN, the Accumulator will be cleared automatically before the transfer takes place. Transfers into B via UN or VN are ORed (exclusive OR, partial add) with the contents of the Accumulator, and the result is left in the Accumulator. The Accumulator will not be cleared before UN and VN transfers unless a special clear enabling level (CLEL) is supplied along with the appropriate channel enabling level (UNEL or VNEL) before restart. *Time of Execution: 16 μsec. when no pause.

OPR

Digital Input to Memory. Information may be transferred from external equipment to the LINC memory via UN or VN. Information, transferred one word at a time, is stored in consecutive locations in the LINC memory. The number of words transferred each time the OPR instruction is executed is controlled by the external device. The computer will pause and transfer information repeatedly, until the external device indicated that no more transfers are to be made. Transfers are handled in the following way: the first word transferred must be a beginning address for storing subsequent transfers. This is transmitted over UN or VN, and the appropriate channel enabling level must be supplied (UNEL or VNEl). In addition, a begin transfer level (BEGT) and a memory input level (MINP) must be presented to the computer before restart. After restart, the computer transfers the word over UN or VN to the B register and to the memory address register. The Accumulator is cleared automatically, and the computer prepares to store subsequent transfers in the memory. The computer then pauses.

When the external device is ready with the first word of information, it must present enabling levels UNEL or VNEL and MINP before restart. The clear enabling level (CLEL) may be present. The begin transfer level (BEGT) may not be present. After restart the word is transferred to the B register over UN or VN, and stored in the LINC memory at the location specified by the memory address register. It is also ORed (exculsive OR, partial add) with the contents of the Accumulator and the result is left in the Accomulator. (The Accumulator will not have been cleared unless CLEL was present.) The contents of the memory address register are incremented by one in preparation for the next transfer, and the computer again pauses. The process is repeated as described above, until the last word is ready to be transferred. This time the external device presents only the channel enabling level (UNEL or VNEL) and the restart signal. MINP may not be present. After restart the computer completes the last transfer, and goes to p + 1 for the next instruction. The partial sum of all words transferred to the memory is left in the Accumulator.
*Time of Execution: 16 μsec. plus 8 μsec. for each word input to memory, exclusive of pauses.

External Output from Memory. Information may be transferred from the LINC memory to an external device directly from the B register. The operation is similar to memory input, except that a memory output level, MOUT, is presented instead of MINP. A beginning address must be supplied over UN or VN along with BEGT and MOUT, as described above; after restart the beginning address is transferred to the memory address register and the contents of the word at that location replace the contents of the B register. The computer then pauses. The external device completes the transfer from the B register. This time the MOUT level must be present before restart. CLEL is optional, and BEGT, UNEL, and VNEL may not be present. After restart the contents of the B register are ORed (exclusive or, partial add) with the contents of the Accumulator, and the result is left in the Accumulator. The contents of the memory address register are incremented by one, the next word in the memory replaces the contents of the B register, and the computer pauses. The process continues until the MOUT level is removed. The partial sum of all words transferred from the memory is left in the Accumulator, and the computer goes to p + 1 for the next instruction. The memory is left unchanged. *Time of Execution: 16 μsec. plus 8 μsec. for each word output, exclusive of pauses.

VII. SKIP CLASS INSTRUCTIONS

In these instructions the i-bit can be used to reverse the skip decision; that is, when i = 0 the computer will skip the next instruction (actually, the first register of the next instruction) only when the specified condition is met. However, when i = 1, the computer will skip only when the condition is not met; otherwise it will go on to the next instruction in sequence. The four situations which may arise are summarized in the following table in which p + n, n = 1 or 2, is the location of the next instruction.

TABLE IV. BRANCHING IN SKIP CLASS INSTRUCTIONS

i	condition	n	p + n
0	met +	2	p + 2
0	met −	1	p + 1
1	met +	1	p + 1
1	met −	2	p + 2

SNS i n — 440 + 20i + n — 8 μsec — SNS

SENSE SWITCH. Check to see if Sense Switch n (n = 0, 1,..., 5 octal) on the control console is up (set to one), and go to p + n (see Table IV) for the next instruction.

AZE i — 450 + 20i — 8 μsec — AZE

ACCUMULATOR ZERO. Check to see if the contents of the Accumulator equal positive or negative zero (all zeros or all ones) and go to p + n (see Table IV) for the next instruction. C(ACC) are unchanged.

APO i — 451 + 20i — 8 μsec — APO

ACCUMULATOR POSITIVE. Check to see if the sign bit (bit 11) of the Accumulator is positive (zero) and go to p + n (see Table IV) for the next instruction. C(ACC) are unchanged.

LZE i — 452 + 20i — 8 μsec — LZE

LINK ZERO. Check to see if the Link bit is zero and go to p + n (see Table IV) for the next instruction. C(ACC) and the Link bit are unchanged.

IBZ i	453 + 20i	8 μsec	IBZ

INTER BLOCK ZONE. Check to see whether either tape is in an Inter Block Zone and go to p + n (see Table IV) for the next instruction. A tape must be moving and up to speed for this condition to be met. The tapes are unaffected.

SXL i n	400 +20i + n	8 μsec	SXL

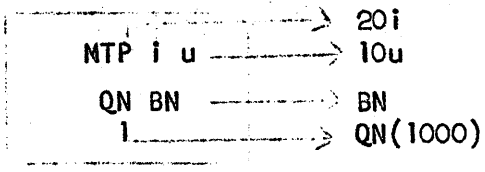
SKIP ON EXTERNAL LEVEL. Check to see if external level n (n = 0, 1,..., 14 octal) is present and go to p + n (see Table IV) for the next instruction.

KST i	415 + 20i	8 μsec	KST

KEY STRUCK. Check to see if a key has been struck on the Keyboard and go to p + n (see Table IV) for the next instruction. The keyboard is unaffected.

VIII. MAGNETIC TAPE INSTRUCTIONS

MAGNETIC TAPE INSTRUCTION SUMMARY



i: Motion Control

i = 0 Tape stops

i = 1 Tape moves

u: Unit Select

u = 0 ⁄ Unit 0

u = 1 ⁄ Unit 1

0 330

QN: Quarter Number $0 \leq QN \leq 7$

QN	Memory Registers
0	0 - 377
1	400 - 777
2	1000 - 1377
3	1400 - 1777
4	2000 - 2377
5	2400 - 2777
6	3000 - 3377
7	3400 - 3777

BN: Block Number $0 \leq BN \leq 777$ (octal)

1 Tape = 512 (decimal) Blocks 1000_8

1 Block = 256 (decimal) Words 400_8

1 Word = 12 (decimal) Bits

Data Sum = two's complement sum of 256 Words in Block

Check Sum = $\overline{\text{Data Sum}}$

Check Sum + Data Sum = Transfer Check

To Check: Transfer Check = -0

RDC i u 700 + 20i + 10u RDC

READ AND CHECK. The specified Block is read into the specified Memory Quarter and the transfer is checked. If it does not check, the Block is read and checked again. If it checks, -0 is left in the Accumulator and the computer goes to p + 2 for the next instruction. The information on the tape is unchanged.

RCG i u 701 + 20i + 10u RCG

READ AND CHECK GROUP. Consecutive Blocks are read into consecutive Memory Quarters and the transfers are checked. The BN bits in p + 1 specify the initial Block; bits 0-2 in p + 1 specify the initial Memory Quarter. Bits 9-11 in p + 1 (the QN bits) specify the number of additional Blocks to read after the initial Block. That is, C(bits 9-11) + 1 equal the total number of Blocks to read. If a Block does not check, it is read and checked again. When all Blocks have been read and checked, -0 is left in the Accumulator, and the computer goes to p + 2 for the next instruction. The information on the tape is unchanged.

RDE i u 702 + 20i + 10u RDE

READ TAPE. The specified Block from the tape is read into the specified Memory Quarter, the Transfer Check is formed and left in the Accumulator. The computer goes to p + 2 for the next instruction. The information on the tape is unchanged.

MTB i u 703 + 20i + 10u MTB

MOVE TOWARD BLOCK. The next Block Number, either forward or backward, is added to the BN bits (bits 0-8) of p + 1. (The QN bits are ignored.) The result is left in the Accumulator. If i = 1, the tape is left moving toward the Block specified by p + 1. If i = 0, the tape stops. The information on the tape and in memory is unchanged. The computer goes to p + 2 for the next instruction.

WRC i u 704 + 20i + 10u WRC

WRITE AND CHECK. The specified Memory Quarter is written in the specified Block on tape. The Check Sum is written on the tape. The tape reverses, finds the specified Block again, and checks the transfer. If it does not check, the instruction is repeated. If it checks, =0 is left in the Accumulator and the computer goes to p + 2 for the next instruction. The memory is unchanged.

WCG i u 705 + 20i + 10u WCG

WRITE AND CHECK GROUP. Consecutive Memory Quarters are written, with their Check Sums, in consecutive Blocks on the tape. The BN bits in p + 1 specify the initial Block; bits 0-2 in p + 1 specify the initial Memory Quarter. Bits 9-11 in p + 1 (the QN bits) specify the number of additional Blocks to write after the initial Block. That is, C(bits 9-11) + 1 equal the total number of Blocks to write. After all Blocks and their Check Sums are written, the tape reverses, finds the initial Block again, and checks the transfers. If a Block does not check, the instruction is repeated beginning with the Block which failed. When all Blocks have been written and checked, -0 is left in the Accumulator, and the computer goes to p + 2 for the next instruction. The memory is unchanged.

WRI i u 706 + 20i + 10u WRI

WRITE TAPE. The specified Memory Quarter is written in the specified Block on the tape. The Check Sum, formed and left in the Accumulator, is written on the tape. The computer goes to p + 2 for the next instruction. The memory is unchanged.

CHK i u 707 + 20i + 10u CHK

CHECK TAPE. The contents of the specified Block are added together in the Accumulator to form the Data Sum. The Check Sum from the tape is added to the Data Sum and the resulting Transfer Check is left in the Accumulator. The information on the tape and in memory is unchanged. The computer goes to p + 2 for the next instruction.