LEO III instruction set.

Program actions on the *LEO III* computer.
The LEO III instruction set showed a marked contrast to those of LEO I and LEO II. Although familiar, basic actions were included, i.e. Add, Subtract, Shift, Change Sequence and Halt, the extra bits in the half-word (19 instead of 17) and the power of the microcode concept allowed for:

1. Actions which were omitted from LEO I, for example Divide, Set-up and Step-on Modification Registers, Enter and Leave Sub-routines and Double Length Arithmetic.

2. New categories of action required by LEO III's more advanced facilities, for example storage allocation/protection, setting radix for arithmetic and tape deck control.

3. Complex logic to be carried out by a single action, for example Merge, Table Look Up and Floating Point Arithmetic.

The format of the instruction was:

5	1	2	13
Action	d	m	Address

```
13 12----------7 6------------1
N3 <----N2---> <----N1---->
<--------------N---------------->
```

The notation applied in the action list (see later) is:

A	Register A.
B	Register B.
C	Excess Contents Register.
SC	Sequence Control Register.
T	Tag Register
A*	Floating Point Arithmetic Accumulator (Registers A and B).
N	The least significant 13 bits of the instruction address.
N1	The least significant 6 bits of the instruction address.
N2	Bits 7 to 12 of the instruction address.
N3	Bit 13 of the instruction address.
N	The instruction address plus the division number.
N'	As for N except that the address is for a long location.
d	The discriminant of the action. In many actions this specifies the type of word transfer to and from the store (d = 0 = short, d = 1 = long) but in some cases it specifies a variant of the action.

m	The modifier of the action. In many actions this specifies the Modification Register within the Modification Register Group to be used to modify the instruction address but in some cases it specifies a variant of the action.

The use of brackets denotes the contents of the location or register specified, e.g. (A) means the contents of Register A. The use of vertical bars denotes the modulus of a location, e.g. |N| means the modulus of location N.

The data for the above explanation, and for the listing which follows, has been gathered from three sources:

1. LEO III Users Manual, Volume 1.

2. Peter J. Bird's book "LEO: The First Business Computer", see http://www.leocomputers.org.uk/newbooks.htm .

3. LEO III Maintenance Manual, Volume 1.

Tony Morgan Eastcote, May 2009.

LEO III instruction set.

Action/d/m	**name**	**Description of program action**
0/0/0	HALT	Stop machine and light stop lamp.
0/d/2	REPLACE	Replace (B) by (N).
0/0/3	SET RADIX	Copy (N) to C.
0/1/0	COPY REGISTERS	Copy (A), (B), (C) to N', N+2', N + 4'.
0/1/1	REPLACE REGISTERS	Replace (A), (B), (C) by (N'), (N+2'), (N+ 4').
1/d/0	TABLE LOOK UP	Search locations N onwards for a literal L in a compartment where in binary (L) >= (A) then place L in B.

1/0/1	PREPARE FOR DIGIT COLLATION	For each of the least significant 10 bits of N (literal) which is 1 set the corresponding quartet of B as ‘1111’ and for each which is 0 set the corresponding quartet of B as ‘0000’. Copy bit 11 of N to sign bit of B.
1/0/2	ROUND OFF	Add 1 to (A) if Q10 of B >= N and clear B.
1/0/3	INTERCHANGE AREA ADDRESSES	Interchange (64 + N1) and (64 + N2). Used for non-Fast Channel input/output.
1/1/1	ADD LITERAL ADDRESS	Add N to (A).
1/1/2	SUBTRACT LITERAL ADDRESS	Subtract N from (A).
1/1/3	SELECT LITERAL ADDRESS	Select N into A.
2/d/m	TRANSFER	Transfer (A) to N and clear A.
3/d/m	COPY	Copy (A) to N.
4/d/m	ADD	Add (N) to (A).
5/d/m	SUBTRACT	Subtract (N) from (A).
6/d/m	SELECT	Select (N) into A.
7/d/m	AUGMENT	Augment (N) by (A).
8/0/m	MERGE CONSTANT LENGTH	Merge strings of constant length data, specified by modification registers 1 & 2 into a block specified by modification register 3, comparing items and selecting the lower, until either of the two input blocks becomes exhausted or the output block becomes full. Item length is (N) in binary.
8/1/m	MERGE VARIABLE LENGTH	Merge strings of variable length data, specified by modification registers 1 & 2 into a block specified by modification

		register 3, comparing items and selecting the lower, until either of the two input blocks becomes exhausted or the output block becomes full. Item length is specified in binary by a single word entry preceding each data item.
9/d/m	MULTIPLY UNIFORM RADIX	Multiply (A) by (N) and place in AB.
10/d/m	MULTIPLY AND ADD	Multiply (B) by (N) and add to (A).
11/d/m	MULTIPLY AND SUBTRACT	Multiply (B) by (N) and subtract from (A).
12/d/m	CONVERT	Convert (N) to the radix (C) and place In A using table held in (A) onwards.
13/d/m	DIVIDE UNIFORM RADIX	Divide (AB) by (N) leaving quotient in A and remainder in B.
14/d/m	REPLACE SELECTED BITS	Replace bits of (N) specified by (B) with corresponding bits of the sign and modulus form of (A).
15/d/m	COLLATE AND ADD	Collate (N) with (B) and add to (A).
16/1/0	COMPARE	Compare strings S1 and S2 of alpha data where S1 starts at N and S2 at (A). Successive pairs of words are compared until a difference is found or the end of the strings is reached. The next instruction is obeyed if S2 is > S1, one instruction is skipped if S2 = S1, and two instructions are skipped is S2 is < S1. (Only available on LEO 360 and 326).
17/0/0	SPECIAL SELECT	Select (N’) into A if N is even, otherwise select (N+1’) into A. No change is made between sign and modulus and sign and complement form.
17/1/0	SPECIAL COPY	Copy (A) to N’ if N is even, otherwise copy (A) to N+1’. No change is made

		between sign and modulus and sign and complement form.
18/0/0	SHIFT LOGICAL	Shift (A) logically.
18/0/1	SHIFT ARITHMETIC	Shift (A) arithmetically.
18/0/2	SCALE NUMERATOR	Shift (AB) so that the most significant non-zero digit of (AB) is in Q9 of A.
18/0/3	SHIFT BINARY	Binary left shift (A).
18/1/0	SHIFT LOGICAL	Shift (AB) logically.
18/1/1	SHIFT ARITHMETIC	Shift (AB) arithmetically.
18/1/2	SCALE DENOMINATOR	Shift (AB) so that the most significant non-zero digit of (AB) is in Q9 of A.
18/1/3	SHIFT BINARY	Binary left shift (AB).
19/0/0	OUTPUT	Output one block to route N1.
19/0/1	INPUT ONE BLOCK/RESET TIMER	Input one block from route N1./ Reset to zero. (Timer only)
19/0/2	RUN BACK *	Run back to last block mark on route N1.
19/0/3	RUN FORWARD *	Run forward to next mark on route N1.
19/1/0	STEP BACK *	Step back one block on route N1.
19/1/1	REWIND *	Rewind route N1 ready to read/write first block.
19/1/2	UNLOAD *	Unload route N1 and set route to manual.
19/1/3	INPUT FIRST WORD */SET ROUTE N1 TO MANUAL	Input first word of block and run forward to next block end without further transfer of information/Set route N1 to manual (non-magnetic tape media).

20/0/m	ADD FLOATING POINT	Add (N) to (A*).
20/1/m	SUBTRACT FLOATING POINT	Subtract (N) from (A*).
21/0/m	TRANSFER FLOATING POINT	Transfer (A*) to N.
21/1/m	COPY FLOATING POINT	Copy (A*) to N.
22/0/m	MULTIPLY FLOATING POINT	Multiply (A*) by (N) and place in A*.
22/1/m	DIVIDE FLOATING POINT	Divide (A*) by (N) and place in A*.
23/0/0	STEP ON AND TEST	Step on and test indirect modifier by <u>N</u>. If it equals end value skip two instructions, if not skip one.
23/0/2	ENTER MASTER ROUTINE	Place SC + 1 in bits 1 to 15 of N and (T) in bits 17 to 20 set and T to 14. Change sequence to N + 1.
23/0/3	SELECT TAG	Select tag of N into A clearing rest of A.
23/1/m	COPY INTO TAG	Copy (A) into tag of N.
24/0/0	MODIFY ADDRESS OF NEXT INSTRUCTION WITH A POSITIVE MODIFIER AND CURRENT DIVISION NUMBER	Search locations N, (|N|), (|(|N|)|), etc. until a location with positive contents is found. Modify address of next instruction by bits 1 to 15 of that location after adding the division number from SC. Modify address of next instruction by (N) after adding the division number from SC.
24/0/1	MODIFY ADDRESS OF NEXT INSTRUCTION	Modify address of next instruction by (N) after adding the division number from SC.
24/0/2	SELECT LITERAL AND DIVISION NUMBER	Select N and division number from SC into A.
24/0/3	MODIFY ADDRESS OF NEXT INSTRUCTION SUPPRESSING DIVISION NUMBER	Modify address of next instruction by (N) suppressing current division number
24/1/0	UNCONDITIONAL SEQUENCE	Change sequence to N.

	CHANGE	
24/1/1	SET MODIFICATION GROUP	Set bits 14 and 15 of indicator register to bits 14 and 15 of (N).
24/1/2 0	MODIFY ADDRESS OF NEXT INSTRUCTION WITH A POSITIVE MODIFIER SUPPRESSING DIVISION NUMBER	Search locations N, (|N|), (|(|N|)|), etc. until a location with positive contents is found. Modify address of next instruction by bits of that location suppressing the division number from SC.
25/0/m	STEP ON AND TEST MODIFICATION REGISTER	Step on and test modification register by N. If it equals end value skip one Instruction.
25/1/0	SET INDICATORS	Where bits of N are ‘1’ set corresponding bits of indicator register.
25/1/1	CLEAR INDICATORS	Where bits of N are ‘1’ reset corresponding bits of indicator register.
25/1/2	INTERROGATE INDICATORS	Collate indicator register with N and copy result to A.
25/1/3	CONDITIONAL HALT	Collate indicator register with N and Halt the computer if result non-zero.
26/0/0	ENTER SUBROUTINE	Place (SC) in N and change sequence To N + 1.
26/0/1	LEAVE SUBROUTINE	Change sequence to (N).
26/0/2	ENTER PRIORITY CONTROL	Place (SC) in N bits 1 to 15 and (T) in N bits 17 to 20 and change sequence to N + 1.
26/0/3	LEAVE MASTER ROUTINE	Replace (T) by bits 17 to 20 of (N). Set bit 13 of I and change sequence to bits 1 to 15 of (N).
26/1/0	TEST ROUTE	Test route specified by N literal.
26/1/m	SET MODIFICATION REGISTER	Copy (N) to modification register.
27/0/0	TEST SHORT ACCUMULATOR = 0	Test (A) and change sequence to N if

		(A) equal to 0.
27/0/1	TEST ACCUMULATOR non 0	Test (A) and change sequence to N if (A) not equal to 0.
27/0/2	TEST SHORT ACCUMULATOR +ve/0	Test (A) and change sequence to N if (A) positive or 0.
27/0/3	TEST LONG ACCUMULATOR –ve	Test (A) and change sequence to N if (A) negative.
27/1/0	TEST LONG ACCUMULATOR = 0	Test (AB) and change sequence to N if (AB) equal to 0.
27/1/1	TEST LONG ACCUMULATOR non 0	Test (AB) and change sequence to N if (AB) not equal 0.
27/1/2	TEST LONG ACCUMULATOR +ve/0	Test (AB) and change sequence to N if (AB) positive or 0.
27/1/3	TEST LONG ACCUMULATOR –ve	Test (AB) and change sequence to N if (AB) negative.
28/0/0	BULK COPY SHORT NUMERIC	If bit 38 of A = 0 copy short numeric words to N onwards the number of items specified by table entry (A).
28/0/0	BULK CLEAR SHORT	If bit 38 of A = 1 clear short compartments starting at N.
28/0/1	BULK COPY SHORT NUMERIC TO ALPHA	If bit 38 of A = 0 copy short numeric words into alpha form to N onwards the number of items specified by table entry (A). N must be D + 1 where D is the first long destination.
28/0/2	UNPACK FIXED FIELD DATA	Unpack data in alpha octet form in a fixed field layout in long compartments starting at N according to table entry starting at (A).
28/0/3	UNPACK VARIABLE FIELD DATA	Unpack data in alpha octet form in a variable field layout in long compartments starting at N according to the number of items specified by table entry (A).

28/1/0	BULK COPY ALPHA TO SHORT NUMERIC	If bit 38 of A = 0 copy alpha words to short numeric form words to N onwards the number of items specified by table entry (A).
28/1/1	BULK COPY LONG NUMERIC OR ALPHA	If bit 38 of A = 0 copy long numeric or alpha words to N onwards the number of items specified by table entry (A).
28/1/1	BULK CLEAR LONG	If bit 38 of A = 1 clear long compartment
28/1/2	EDIT FIXED FIELD FORMATS	Edit items into fixed field layout in alpha sextet form in compartments starting at N according to table entry starting at (A).
28/1/3	CONDENSE	Condense items into variable field layout in alpha sextet form in compartments starting at N according to table entry starting at (A).
29/0/0	EDIT FOR HOLLERITH OUTPUT	After action 28/1/2 edit up to 80 characters for card punch output by General Purpose Output Assembler into N onwards.
29/0/1	EDIT FOR ANELEX OUTPUT	After action 28/1/2 edit up to 160 characters for line printer output by General Purpose Output Assembler into N onwards.
30/0/m	TRANSFER DOUBLE LENGTH	Transfer (AB) to N + 2 and N and clear A and B.
30/1/m	COPY DOUBLE LENGTH	Copy (AB) to N + 2 and N.
31/0/m	ADD DOUBLE LENGTH	Add (N + 2) and (N) to (AB).
31/1/m	SUBTRACT DOUBLE LENGTH	Subtract (N + 2) and (N) from (AB).

** Magnetic tape only.*