The Evolution of Design in a series of Computers, LEO I–III

By J. M. M. Pinkerton

This article surveys the evolution of the functional and systematic aspects of design in three generations of LEO computers, indicating how the earlier ideas have been extended or generalized. Changes have been made to meet actual needs and are the direct result of experience of designing, building, programming, operating, and maintaining computer equipment over more than a decade.

Introduction

Readers of this journal will be aware that there are now three generations of LEO Computers. LEO I, a large valve machine, was designed and built by J. Lyons & Co. Ltd.; starting in 1949 it was completed in 1953, and is still heavily used by them. It was intended to demonstrate the feasibility of business procedures on a computer. LEO II, of which eleven are now working, for various users, is a more advanced design but still mainly using valves. All the LEO II models belong to a common series but differ from one another to a varying extent; later models have large transistorized core stores, and magnetic tape. LEO III is a new, all-transistorized system of great power and flexibility, embodying multi-programming (interrupt) facilities and a microprogrammed control unit.

Along with these developments in engineering practice there have been corresponding changes in approach to programming and use of the computer. On the one hand, the programmer of a LEO III enjoys a wider range of actions than with the earlier machines; on the other, by means of automatic programming procedures it is becoming much easier to command those actions effectively. The operator, too, now has far greater control of the job (or jobs) in hand yet is able to rely far more on the machine itself to help him. In a sense the machine has become more self-conscious. This article aims to trace the evolution of ideas underlying these changes. It is not possible in a short article to deal fully with this subject, so only the more fundamental or interesting aspects are referred to.

Evolution of the whole System

LEO I was always intended to perform useful clerical work; until such work had been done the real needs of a business computer would not be known. Nevertheless, something had to be built. As a start, therefore, the logical structure and engineering techniques of EDSAC I were adopted. These were seen to provide only the nucleus of an effective system. The input/output demands of typical business problems had not been satisfied in EDSAC I. A system of multiple input/output channels with a small buffer store in each was therefore evolved for LEO I. In LEO II these buffers, in the various input channels, were retained. In LEO III, however, they have been incorporated in the main store; the automatic and independent functioning of the input/output channels has, however, been retained.

It was recognized in LEO I also that conversion of numbers to and from binary could be very time-consuming in business jobs; special instructions were therefore included for converting automatically between binary and either decimal or sterling. These instructions were continued in LEO II. In LEO III provision is made for carrying out arithmetic directly in any system of units as well as in binary; conversion instructions are also available.

It was always felt that the high-speed direct-access store needed to be large, and the storage capacity of LEO I was made twice that of EDSAC I which, when LEO I was commissioned, represented a very large high-speed store. The later LEO II computers were fitted with magnetic-core stores as large as 8,192 long words (of 39 bits). Supplementary magnetic-drum storage was also used with some LEO II computers, transfers being made in blocks of 16 words. LEO III provides core storage systems with direct access holding up to 32,768 long words.

The linkage of very diverse input/output mechanisms with the computer is a notable feature of the LEO series. At different times paper-tape and punched-card readers, magnetic-tape transports, Hollerith, Bull, Samastronic, and Anelex line-printers, paper-tape and punched-card punches, and teleprinters have been used, in a suitable combination with any given machine. With LEO II provision for the particular mechanism required must be made when the machine is built. To get away from this the unit system concept embodied in LEO III was evolved; separate classes of input or output devices are grouped and coupled to the indispensable storage and arithmetic units of the system by means of the so-called *Store Access Control* unit. The transformation of the information from the form used in sensing or recording the input or output mechanism to that used within the store and arithmetic unit, and the detailed control of the mechanisms themselves is done in special-purpose *Assemblers*. All units of the system are self-contained and linked to one another by plugged-in cables. A very flexible installation layout is then possible which grows with the needs of the work load.

Development of the Function Code

A short article cannot mention all the ideas successively incorporated in the function codes of the three systems reviewed; nevertheless, general characteristics can be outlined. The LEO I function code was based on that of EDSAC I but with the addition of such actions as Convert and Reconvert (to or from binary), and block input and block output. The code used a single address, and operands could be *long* or *short* numbers of 17 or 35 binary digits respectively. The single-address feature and use of long and short words have been found efficient and have been continued in LEO II and LEO III. The function code in LEO I had 17 bits, whereas in LEO II it has 19, and in LEO III 21.

LEO I possessed no modification or index register; in LEO II three modification registers are fitted. Since for modern programming techniques these are sometimes inadequate LEO III provides for twelve in four groups of three. In addition a facility is available whereby any storage location can be used to hold a modifier number. To simplify programming a special counting and testing instruction was introduced into LEO II whereby the modifier number was automatically augmented and a test made to see if the end of the count had been reached; if not the usual sequence jump took place, resulting in the required repetition of the subroutine. An instruction of this type also appears in LEO III.

Other instructions in the LEO II code cater for arithmetic to be performed directly between the 16 registers of the arithmetic unit, and for block transfers to or from magnetic-drum storage or to or from magnetic tape.

Extra flexibility in allocating function codes in LEO II and III is secured by dividing functions into two classes, *address-modifiable* and *address-unmodifiable*. In the latter class are actions whose address digits cannot usefully be modified, e.g. shifting and inter-register actions. Two bits are used in LEO II and III either to specify the modifier register or to vary the function itself.

To cater for mixed-radix arithmetic, a function in LEO III sets up constants in the arithmetic register appropriate to the type of arithmetic to be carried out.

Further functions in LEO III cater for compressing alpha-numeric information for output, say to magnetic tape on which the recording of non-significant information wastes time and space, or for automatically decompressing such information on input, so segregating digits of different numbers into separate storage locations. Related actions cater for setting out information in the form required to be printed, and for recording separately stored items as one continuous block on magnetic tape, without having first to gather them together in the store. Finally, LEO III possesses functions for floating-point arithmetic, and for sorting. The range of functions available in LEO II has thus been greatly extended already, and by use of the in-built microprogramming feature, future needs can also be catered for. Indeed, additional microprogram planes to incorporate additional actions can be added comparatively easily on site.

Arithmetic Circuits

In LEO I there were three registers only, the accumulator, the multiplicand, and the multiplier registers. As a consequence numbers had frequently to be transferred back to store, which, owing to the comparatively long access time in that machine, slowed down work appreciably. In LEO II there are up to 16 registers, two of which form the main accumulator while others are used for multiplier and multiplicand. Direct addition and subtraction between any pair of registers is possible. Furthermore, multiplication may be performed in a subsidiary accumulator independently of the main accumulator. These arrangements make possible a significant speeding up of work.

In LEO III the store access time ($\sim$6 microseconds) is so much less than in LEO II ($\sim$150 microseconds) that subsidiary registers can be dispensed with: in a parallel machine their provision would have been expensive in any case. By embracing, in the structure of the registers available for general use, the sequence register and address portion of the instruction register, additional flexibility is obtained; this is made full use of in the more elaborate actions such as table look-up and sorting. There are in LEO III five full-length registers and four shorter ones.

In LEO I four separate half-adders were provided: for adding "1" to the sequence number, for counting the successive locations circulating in the delay lines of the store, and for adding into the accumulator. In LEO II arithmetic and control registers become merged and share a total of three full adder/subtractors. By inserting two separate adder/subtractors in the accumulator loop, faster multiplication is obtained than in LEO I. In LEO III there is one 41-bit parallel binary adder for all arithmetic operations. By virtue of its high speed its use can efficiently be shared by both arithmetic and control functions.

As mentioned above, conversion and reconversion actions are incorporated in LEO I and II. These are based on the provision of wired-in constants; the binary equivalents of 10, 100, 1,000, etc., can be selected by switching circuits and multiplying by 2, 4, and 8 by means of delays of 1, 2, or 3 pulse positions. Thus to convert into binary, the constants equivalent to the "1" bits in the binary coded decimal number are added successively into the accumulator. The process is, of course, similar to multiplication, takes the same time, and, like multiplication, stops automatically when all significant digits are converted.

In LEO III conversion and reconversion are less used because arithmetic is performed directly in the radix in which the number enters the computer. This is achieved by the use of a set of 10 correction constants appropriate to the system of units in use. The value of the constant is the difference between the radix for that position and

16. For decimal arithmetic these constants are all equal to 6. For a sterling number, however, they would be 6, 6, 6, 6, 6, 6, 6, 14, 6, 4. The tens of shillings is allowed a separate digit position, but one position only is allowed for pence. Mixed-radix arithmetic is thus possible, including, for example, decimal multiplication of sterling quantities. By setting all constants equal to zero, arithmetic is executed in binary. Provision of these facilities is greatly assisted by the microprogrammed system of control already mentioned.

Form of Information held in the Systems

In LEO I and II negative numbers are held as complements, which involves programmed conversion to or from sign and modulus form in the input/output procedures. In LEO III negative numbers are held as sign and modulus in the store and converted automatically to complementary form during arithmetic, thus avoiding unnecessary program steps.

To cater for decimal or sterling input and output in LEO I and II, groups of four binary digits were associated and allowed to represent a single digit up to 15. Normally values above 11 did not occur; 10 and 11 as single digits arise in the sterling pence position. LEO I did not handle alphabetic information; in LEO II this is provided for by associating two quartets in the store. One, the basic quartet, is combined with two bits of the other quartet to give a total of 64 possible characters. The usual 39 punched-card codes were thus provided for, with ample spares for special characters.

A variant of this system is used in LEO III. Six-bit alpha-numeric characters are recorded on paper tape and magnetic tape, and the usual one or two hole per column arrangements is used with punched cards. Inside the store, however, the 40-bit word represents five pairs of quartets and, therefore, five alpha-numeric characters. For more efficient packing of purely numerical information in the store, actions are available for discarding the two bits of each character which are relevant only to non-numeric information, thus allowing ten decimal or other digits to be stored as a long word.

Input/Output Channels

The concept of multiple channels for data and results all operating simultaneously was first introduced in LEO I and continues in LEO II and III. An essential part of the concept is that any input/output process, once initiated by the program, is completed automatically. Blocks of data read relatively slowly by the input mechanism are built up gradually—in LEO I and II in separate buffers, in LEO III in areas of the main store assigned by the program in advance—and are made available for instant use when complete. In LEO I and II the completed block has to be transferred into the main store. Important advantages accrue in LEO III from the formation of input blocks in the main store itself.

(*a*) The size of the blocks can be varied by the program to suit the work, during the job if necessary.
(*b*) The blocks can be used for computing as soon as assembled, without unnecessary transfers.
(*c*) The direct interchange of storage areas between input and computing, computing and output, or input and output becomes possible. In LEO III special program facilities provide for this.

The output process is, of course, an exact inverse of the input process.

Time Sharing

In LEO III, since the input and output functions demand direct access to the store as well as the arithmetic and control processes, an automatic means for deciding priority is provided. Were the *access priority* to be decided by program it would place an intolerable burden on the programmer. Priority for store access is to be carefully distinguished from priority for different programs where several are to be held in the store at once. Program priority in LEO III is fully at the disposal of the programmer. In LEO III *Interruptions* to the execution of a program may result from the state of some peripheral device, e.g. a tape transport. A master program decides, in the event of two or more interruptions being attempted at once, which is to be dealt with first. Interruptions are not allowed merely because a peripheral device is available, but are restricted to occasions when it is worth while breaking into current operations. Once in the interrupt condition, further interruptions are forbidden. Much time in dealing with interruptions can thus be saved.

The store access control unit handles all traffic to and from the store; it accepts the address from the unit of highest priority currently claiming access, and affords an information path to or from that address. Up to eight input or output channels can be connected to one store access control unit, in addition to the main frame and a monitor display. Alteration to access priority is made by plugging the assembler cables into different sockets.

Any given channel is fed from or feeds to a number of *routes*, not greater than 8. In general one route corresponds to an individual reading or recording device, e.g. a card reader or magnetic-tape transport. In some cases, two different route numbers allow the same device to be operated under different regimes, e.g. cards to be read punched either in standard or non-standard (e.g. binary) codes. Full performance details of all peripheral devices used with LEO computers would take too much space, but a summary of the peripheral equipment currently available with LEO III is given in Table 1.

Checks and Interlocks

The scope and power of the various built-in checks has, in the light of experience, been gradually increased. In LEO I the only check provided was on card punching.

Table 1

Schedule of Different Input/Output Media and Devices available with LEO III

IN/OUT	MEDIUM	READER/RECORDER	CODE	MAXIMUM SPEED OF READING AND RECORDING	NOTE
In	7-hole perforated tape (fast)	Photo-electric reader	Alpha-numeric plus parity	1,000 chs/sec	
In	80-column punched cards	Photo-electric reader	Alpha-numeric	400 cards/min	
In	80-column punched cards	Photo-electric reader	Alpha-numeric	600 cards/min	
In or Out	Half-inch magnetic tape	High-speed tape transport	Alpha-numeric plus parity	45,000 chs/sec	Effective rate 30,000 chs/sec for blocks of 1,000 chs
In or Out	One-inch magnetic tape	High-speed tape transport	Alpha-numeric plus parity	90,000 chs/sec	Effective rate 45,000 chs/sec for blocks of 1,000 chs
Out	80-column punched cards	Card punch	Alpha-numeric	100 cards/min	
Out	Printed Result (medium speed)	92-column printer	Alpha-numeric	150 lines/min	
Out	Printed Result (high speed)	120-column printer	Alpha-numeric	850 lines/min	For single spacing. The rate drops to 500 lines per min. for 1-in. spacing.
Out	Printed Result (slow)	Electric type-writer	Alpha-numeric	10 chs/sec	
Out	Paper Tape	Paper-tape punch	Alpha-numeric	110 chs/sec	

It had always been felt that programmed checks could and should play a major part in ensuring that errors were brought to light as soon as possible after they occurred, so that remedial action could be quick and effective.

In LEO II, besides the card-punch check, checking is provided in the magnetic-tape system, with the magnetic-core store, and with the magnetic-drum store. The writing check on magnetic tape compares every digit written with the corresponding digit held in the tape buffer before it is cleared.

In LEO III more comprehensive checking is provided than in LEO II, with the aim of giving a close indication of the area of trouble. Thus all sensing processes and most of the recording processes are checked by an appropriate form of parity. All transfers between assemblers and store *and vice versa* are checked, and all information retrieved from store is also checked. Parity checking is also applied to the stepping of the micro-program sequence. Particularly stringent checks are applied to magnetic tape, both in recording and sensing. There is a parity bit on every row and, at the end of each block, a 6-bit block sum-check is formed by an adder having end-around carry. Further, to reduce the probability of an undetected error from a dropout, the sequence, across the tape, of bits forming one character is not the same as that in which the same bits are fed to the check adder.

Experience with LEO has never demonstrated the value of checking in arithmetic circuits. With faults in the arithmetic circuits the whole operation of the program would rapidly be thrown into disorder; the risk of wrong but plausible-seeming results from this cause has been found to be very slight. Thus no checking in the arithmetic circuits themselves has been fitted.

Operation of several programs at once has introduced at least the possibility of new types of trouble. A perfectly correct program could be upset because of an error in another program or in the data for another program. Again, with variable-length block working, input data could overrun the area of store assigned to it, and perhaps encroach on an entirely different program area.

A system of tags is available as an optional extra in LEO III, allowing any storage location to be reserved for the program or programs for which access is to be permitted. Four extra bits are associated with every long word and define the program number of that word. The corresponding key number has to be presented to store to gain access to any word from an input or output channel or the arithmetic unit. If the two numbers do not correspond, interruption occurs handing control to the master routine, which itself has overriding access to all parts of the store. Further, locations having zero program number are accessible to all programs. Another program number is allocated to guard words which prevent input areas spilling over into program or data areas of the same program. Interlocks which enable the programmer to prevent access to incompletely formed blocks of input data, or incompletely recorded output blocks, are provided in the form of special instructions. In LEO II an attempt to use an input or output channel before it is ready results in the system "marking time." In LEO III, however, the same test, being divorced from input/output instruction proper, thus becomes a jump instruction. Other useful work can then be performed if an input/output channel is unready.

Operating Controls and Monitoring Arrangements

It has always been the aim to make the LEO computers easy to operate, but the emphasis has considerably shifted in recent years. In LEO I it was to provide the operator with the controls and monitor indications he needed, conveniently grouped and easy to use. A similar approach is found in LEO II, but with the further increase in numbers of controls and types of peripheral device, and more especially in speed, a new approach is clearly demanded. This can be described as making *the computer more self-conscious*. The objective is to allow a master program to take many of those decisions previously left to the operators, to give the operators detailed warning in advance of the things they have to do, and, to a large extent, to let the computer organize its own work.

If a computer is engaged in doing several jobs at once, one of them might get held up for a significant time without the operators being aware of it, if the computer did not issue a reminder. Means are therefore needed for the computer to communicate with its operators *and vice versa*. The former need is met by an electric typewriter, a variety of special alarm and signal lamps, and illuminated number indicators set up by output instructions; the latter need is met by eight independent indicator keys interrogated by program, as well as the more usual start and stop buttons, and peripheral device controls, e.g. for rewinding or unloading magnetic tape. In addition, comprehensive monitoring of the store and registers is provided, in all machines of the series, for use by engineers, though its use in routine operation is being discouraged because it is too slow. Techniques of trying out programs without excessive use of machine time become possible with the interrupt feature incorporated in LEO III.

In conclusion, it is emphasized that the LEO III system represents, with all its improvements, yet another step in the evolution of computer design. When experience of its use has been accumulated, further steps will undoubtedly be seen to be necessary. As in the past, these steps will reflect all aspects of experience in installing, operating, and maintaining computers.

THE COMPUTER JOURNAL

Published Quarterly by

The British Computer Society Limited, Finsbury Court, Finsbury Pavement, LONDON, E.C.2, England.

Subscription price per volume £2 10s. 0d. (U.S. $7.00). Single Copies 15s. 0d.

All inquiries should be sent to the Assistant Secretary at the above address.