

# FERRANTI ARGUS

Process-Control Computer System



*The Basic Argus Computer.*

*"Argus, the all-seeing, had a hundred eyes which slept in turns, so that he was at all times awake."*

# Ferranti ARGUS

## Process-Control Computer System

### Introduction

Recent years have seen fundamental changes both in the characteristics of industrial plants and processes, and in the function of those who operate them. Plants have become larger and processes enormously more complex, so that very difficult control problems are constantly arising. The functions of the operators of many processes are now largely supervisory, the actual plant operation being governed directly by some form of automatic – often analogue – control system.

Meanwhile, the digital computer has come into use as a research tool in designing new plant, and digital techniques are being widely used for data logging and alarm systems. The next stage in the evolution of process control has been made possible by the latest advances in electronics, thermionic valves and electro-mechanical storage elements being replaced by much more reliable devices such as transistors and ferrite cores. The application of these techniques permits the supervision of plant control systems by means of computers, or even the direct control of the plant, with a consequent improvement in the efficiency of the process and in the flexibility of control.

With an already well-established reputation in the field of digital computers generally, and with a long background in the design of specialised control systems, Ferranti Ltd. have now produced Argus, a digital computer designed specifically for process control. The special features necessary in a computer intended for work of this kind have been included from the start – it should be emphasised that Argus is not just an adaptation of a general purpose computer.

### Applications

The use of a digital computer for process control shows immediate advantages in flexibility and in the degree of sophistication of control that can be achieved. Flexibility arises naturally from the fact that the computer is a programmed instrument; that is, a programme of instructions detailing the computation and logical operations to be carried out is stored within the computer and can easily be altered or expanded. Sophistication is linked with this in that there is no necessity for restriction to linear control systems, so that simple forms of control can be chosen initially and improved upon later in the light of experience.

The range of applications for the Argus computer is very wide. Where existing controllers are already in use – as in all kinds of chemical processes, whether continuous or batch production – significant improvements in consistency of control can often be achieved. There are also applications in fields where no adequate form of control has yet been developed, in which the handling of logic is as important as the manipulation of numerical information. Traffic control and the starting up of otherwise continuous processes are typical examples.

The degree of control that is exercised by the computer in any application can be varied to suit the process. Even when there are no immediate control requirements, it is often possible to justify financially an Argus installation for on-line data reduction, producing a small amount of really meaningful information, or operating guides for manual control. At a more advanced level, the operating guides can be applied directly to adjust the set points of existing analogue controllers. Ultimately, the controllers themselves may be eliminated, Argus forming a direct link between measuring elements and controlling devices, and providing control transfer functions of any desired form.

Argus can be used at any of these levels, or can progress from one to another without the expense of replacing each successive installation. By suitable time-sharing techniques, a single computer can often serve a number of independent processes.

### **Background**

For those who are unfamiliar with digital computers, a word of explanation may be helpful. Any computation that has to be performed must be broken down into a programme, which is a sequence of elementary instructions, each of which is usually an arithmetical operation such as addition or multiplication. The programme is stored within the computer, which can therefore repeat (or alternatively ignore) some sections as required; the decision for such a repetition is made, for example, on the result of a computation. Writing programmes for any computer is necessarily a specialised skill, and Ferranti Ltd. employ a large staff of programmers to advise and help customers at any level.

Numbers within the computer are held in binary form, so that successive digits represent powers of 2, as opposed to powers of 10 in decimal notation. Each binary digit or "bit" can only have two values, 0 or 1 (contrast 0 to 9), which is convenient for purposes of design. Temporary storage is provided for intermediate results of computations, one number occupying one location in the store. The greater the number of bits, the more accurate the computer: one decimal place corresponds to about three bits. Each instruction is stored in the computer as a set of binary numbers which are decoded before the instruction is obeyed.

### **The Argus Computer**

In designing a computer for process control, the following requirements have to be considered:

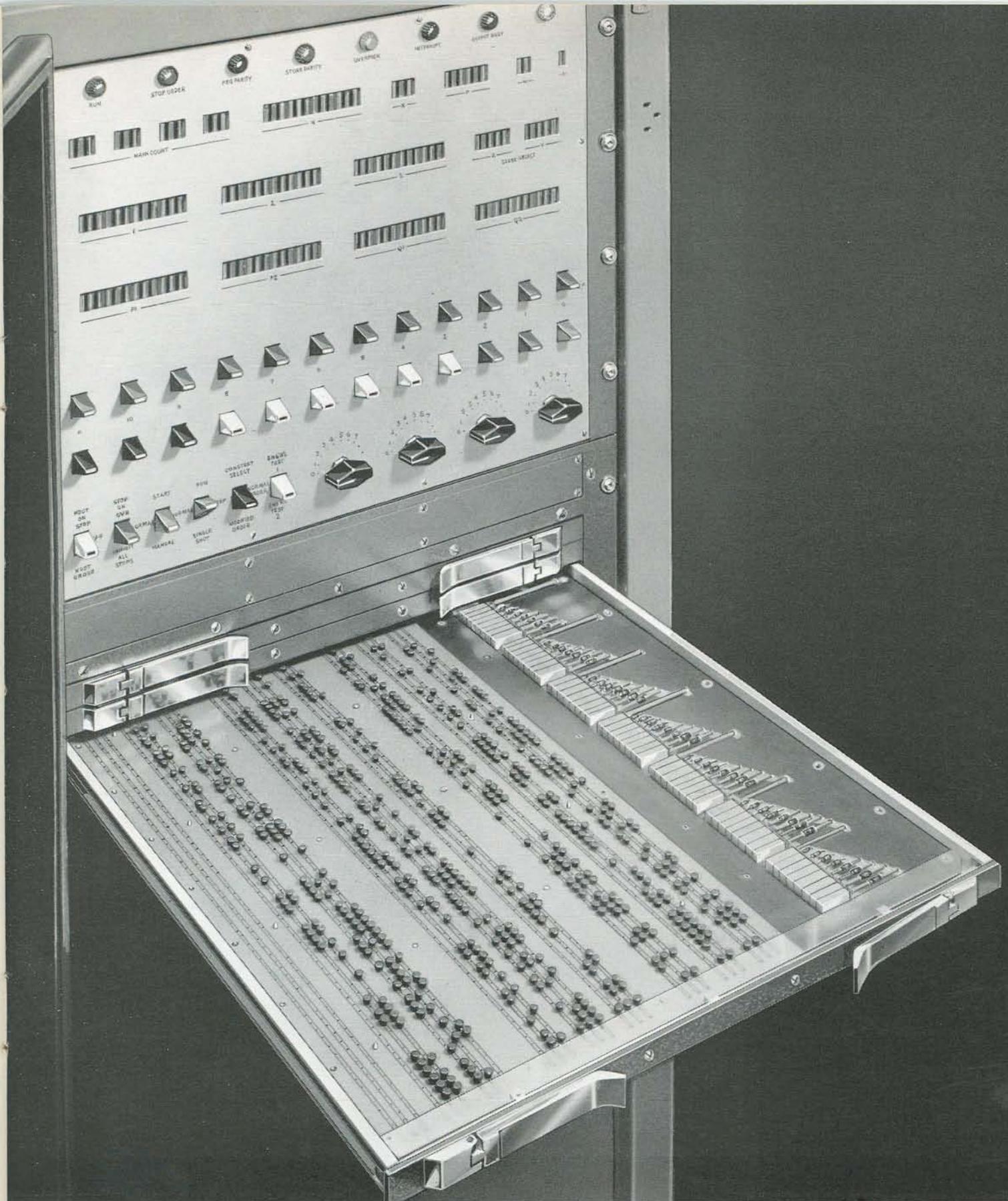
- (1) Reliability. It has already been remarked above that very high reliability is an essential feature in process control.
- (2) Ease of programming. This is a desirable feature in any digital computer, and depends upon the order code.
- (3) Timing. Some method of ensuring the exact timing of operations within the computer is vital, since many external devices depend on this, and also because the theory of control requires regular sampling of inputs.
- (4) Suitable accuracy and speed. These two conflict with one another to some extent and a compromise must be carefully chosen.
- (5) Input and output facilities. One of the principal differences between a general-purpose computer and a process control computer is that the former uses comparatively few input and output devices, whereas the latter must also expect to take readings from many instruments and send signals to many control points.

Argus has been designed to satisfy all these conditions. A complete specification is given at the end of this pamphlet, and is discussed in greater detail under the separate headings below.

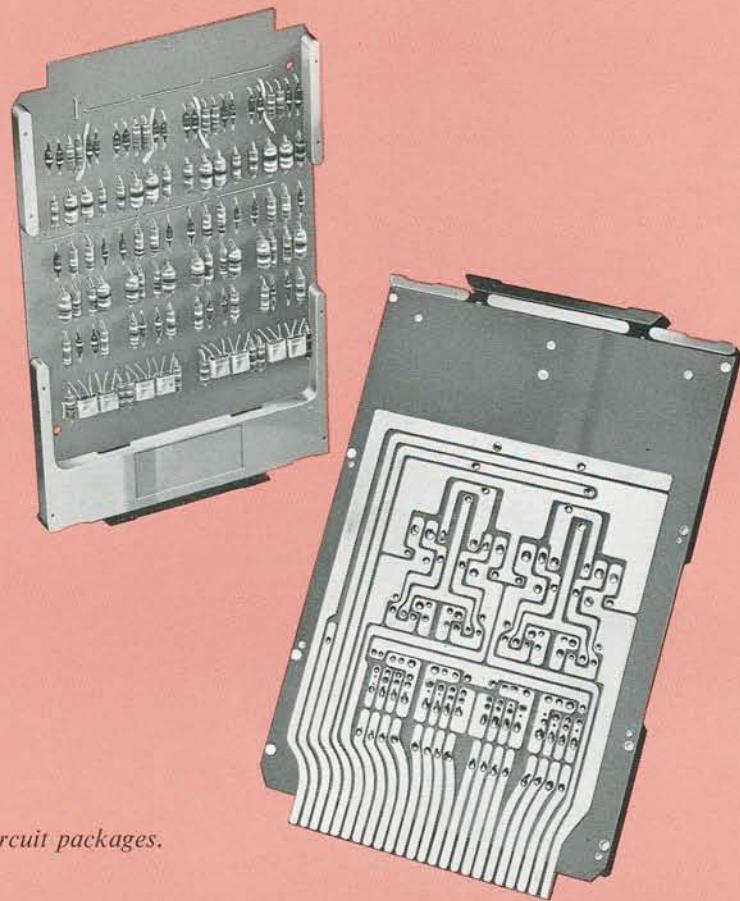
### **Reliability**

In order to achieve the reliability required, the standards of design and engineering are very high. The circuitry employs only solid state devices and is based on a small number of elementary functions. Unit construction, employing printed circuit cards of a few standard types, forms the basis of the construction; these are plugged into connectors in boxes, whose own interconnections are made by wrapped joints and wire loom held in special channel sections. The entire computer conforms with AID standards, to satisfy the exacting demands of certain military applications.

The standard of reliability is so high that preventive maintenance is not envisaged for most applications. When failure does occur, a diagnostic programme can be used to assist in locating the fault, and the relevant package can be replaced from a small stock of spares. Measures can also be taken to guard against breakdown during operation; this is arranged by performing a special test computation at regular intervals and automatically taking alarm action if it is not completed satisfactorily.



The programme storage is of peg-board construction. A typical tray is illustrated.  
The monitor panel can be seen above.



*Typical Argus printed circuit packages.*

*Box containing 36 packages.*



A particular feature that distinguishes the design of Argus from that of other digital computers is the provision of separate storage for programme and numerical information. The reliability of the core store is adequate for information which changes in the course of the calculation so that errors which might conceivably arise do not normally have any far reaching effects. A further safeguard is the provision of an additional "parity" bit on each storage location which gives an indication when failure occurs. For the programme, which is not changed during operation and requires an even higher standard of reliability, a pegboard system is provided. Instructions are pegged up in their binary form in trays with ferrite pegs, giving the necessary response by means of currents induced in wiring printed on the trays. Orders can only be altered, therefore, by inserting or removing pegs from the trays; accidental alteration is impossible, as the pegs are locked in position by putting the trays into their rack in the computer. A parity check is provided on the pegboards also.

### Programming

The order code for Argus, given at the end of the pamphlet, has been based on that for Pegasus, which has proved itself convenient in use over many years. The main differences are the simplifications introduced by using entirely fast-access core storage for numbers, and the extra provisions for working with double-length numbers where greater accuracy is needed (see below). The amount of core storage provided makes the organisation of programmes particularly easy, but where very large amounts of data have to be handled a magnetic drum can also be supplied. Constants of a permanent nature can be stored interchangeably with programme on the pegboards.

The pegging of programmes on to trays can be made easier by use of an interpretative routine prepared for Pegasus, which will assemble a complete programme from sections, and print a pegging layout. Punched cards can also be used to cover the trays, pegs being inserted in exposed holes.

### Timing

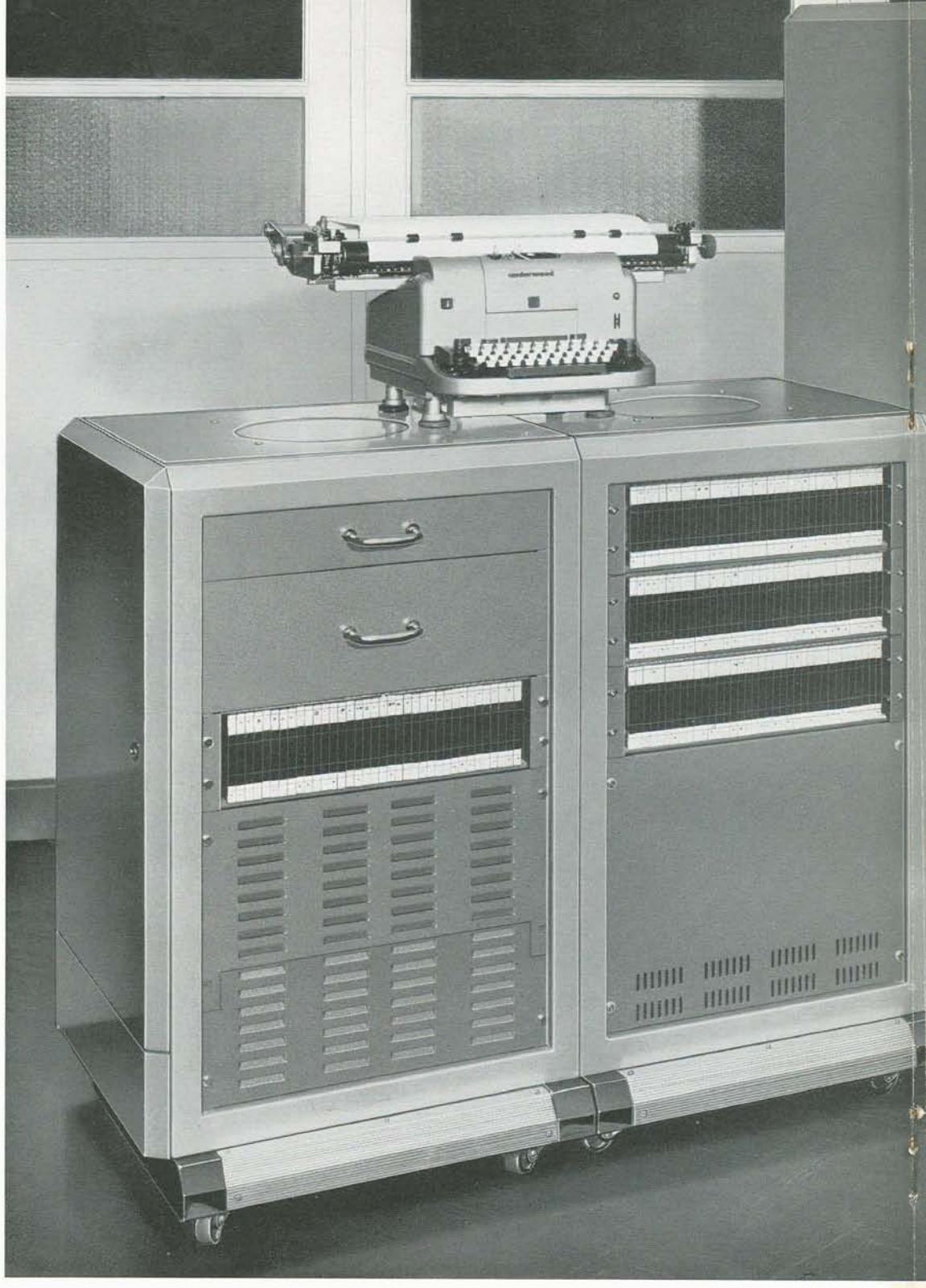
The fact that certain orders take a variable length of time (particularly multiplication and division), and the uncertainty introduced by conditional jumps, make it imperative to include an independent device for the exact control of timing, particularly of input and output. In preference to a clock, treated as a special input, Argus is provided with a timer interrupt scheme as being more flexible and convenient in use. A special register holds an integer which is reduced by 1 every two computer word times (40 microseconds) independently of the other operations being carried out by the programme. When it becomes zero, the programme sequence is broken and returned to the orders at the beginning, which may be terminated by a return order, taking control back to the main programme at the point of interruption.

The integer can be reset by programme to any value between 1 and  $2^{12}$ , giving interrupt periods of up to 160 milliseconds. Several different interrupt periods operating simultaneously can be arranged by suitable programming, using counters held in the core store; this technique also permits the extension of interrupt periods to any desired length.

### Accuracy and Speed

The basic word length of Argus gives an accuracy of 1 part in 2048. This is sufficient to allow for an overall accuracy for simple computations which compares favourably with industrial instrumentation. Where longer computations are involved, round-off errors may build up and cause deterioration in overall accuracy; provision is therefore made for working with equal ease to an accuracy of about 1 part in  $8 \times 10^6$ . Higher accuracies can be obtained by means of special programming techniques.

The times taken for the various functions in the order code are detailed in the specification at the end of the pamphlet (Table 3). They have been chosen to meet the most exacting requirements, found mainly in military applications, and the overall speed is ample for control purposes in industrial processes, where the shortest reaction times normally encountered are of the order of seconds.



A TYPICAL ARGUS



*A typical Argus installation comprising input/output and logging equipment (left) and the basic computer (right).*

## S INSTALLATION

### **Input and Output Facilities**

It is not possible to read more than one input at a time by programme, so that only one input channel is necessary for this purpose. This channel accepts information in standard binary form. Associated with it is an address register sending out an address for a demanded input. The equipment connecting the two, which has to decode the address, operate the switching for the appropriate input, and perform any necessary conversion into digital form, will be custom-built from standard units to suit each application, thus keeping expense to a minimum. Time for selection, switching and conversion to take place can be allowed for in the programme, by separating the input demand order ( $F=30$  to  $37$ ), and the order reading from the input register, by the appropriate amount. Outputs are similarly handled, with an output register and associated address register, but only one programme order is necessary.

The equipment involved in input and output will be housed in one or more separate cabinets, using the same unit construction as the computer itself (and, to a considerable extent, the actual Argus packages), so that the same standard of reliability can be expected. The range of standard input and output equipment available is shown in Table 2; special equipment to meet unusual requirements can also be provided.

In addition to the above method of input and output on programmed demand, there is also a facility for direct access to the core store, intended for use where external requirements cannot conveniently be synchronised with the programme. This facility is known as C.S.I. (Core Store Interrupt), and has its own input and output registers, to which conversion equipment similar to that for programmed input may be connected. Information is taken direct to or from the core store on external demand, using sequential store addresses. The timing is independent of the running of the programme, but an order of priorities has to be decided between C.S.I. and programme when both require access to the core store simultaneously.

**TABLE 1****Computer Specification** (Excluding Input–Output Cabinet)**Type**

Number system: Fixed-point binary.

Mode of operation: Serial-parallel.

Word length: Numbers, 12 or 24 bits including sign; instructions 24 bits; 1 parity checking bit on numbers and orders.

Addressing: Modified single address; multiple accumulators.

**Storage**

Numbers: Immediate access ferrite core matrix, capacity up to 3072 words (12 bits+parity) in units of 1024.

Optional extra MD5 magnetic drum, capacity 50,000 words.

Instructions: Immediate access induction-type pegboard, capacity up to 4096 words (24 bits+parity)  
in units of 1024.

**Special Features**

Timer interrupt, breaking control sequence at times determined by programme.

Input and output by programmed random selection or direct sequential access to store (C.S.I.).

**Physical Construction**

Cabinet: Dust-sealed, 72"×60"×24".

Components: Approximately 4000 transistors and 20,000 diodes depending on capacity; no thermionic valves.

Wiring: Printed circuit plates (15 types) mounted in boxes; interconnections by wrapped joint and wire loom.

**Power Consumption**

Approximately 2 kVA. Normal supplies are 240 V, 50 c/s, 3-phase, but other supplies may be used.

**TABLE 2****Input and Output Equipment****Address Decoding**

This is similar for both input and output, and uses standard Argus logical packages. Any number of input or output addresses can be decoded, up to the capacity of the computer. In many cases, the final stage of the decoding is combined with providing drive to the switches.

**Digital Inputs**

Switching systems for binary information from digitisers and single bit inputs from on-off switches.

Decimal switches, for manual alteration of constants.

Decimal-to-binary converter, for use where programmed conversion is inconvenient.

Paper tape reader.

Magnetic tape reader.

**Analogue Inputs**

Solid-state switching, giving maximum speed and reliability.

Relay switching, for very low-level signals.

Analogue-to-digital converter, giving 10 bits (0.1%) accuracy in 44 microseconds.

D.C. amplifier, for amplifying low-level signals prior to conversion, and smoothing where necessary.

**Digital Outputs**

Drive gates for relays and control solenoids.

Decimal display, giving visual operating guides.

Binary-to-decimal converter, for use where programmed conversion is inconvenient.

Paper tape punch.

Magnetic tape output.

Electric typewriter or other printing device.

**Analogue Outputs**

Digital-to-analogue converter, producing analogue control signals or drive for chart recorders.

TABLE 3

## Programming Information

The 24 bits constituting an instruction (apart from the parity bit) are allocated as follows:

Address <i>N</i> 12 bits	Accumulator <i>X</i> 3 bits	Function <i>F</i> 6 bits	Modifier <i>M</i> 2 bits	Constant <i>C</i> 1 bit
--------------------------------	-----------------------------------	--------------------------------	--------------------------------	-------------------------------

*N*, *X* and *F* are normally written in octal notation, each digit corresponding to 3 bits.

Addresses *N*, *C*

The meaning given to *N* depends on the first of the two *F* digits (function group), and also on the *C* bit.

When *C*=0, the interpretation of *N* is as follows:

Function groups 0, 1, 2: *N* refers to an address in the core store, a special register, or an output address.

Function group 3: *N* refers to an input address.

Function groups 4, 5: *N* is treated as a binary number; in group 4, *N* is a constant used directly in computation, and in group 5, *N* is regarded as a signed integer controlling shift orders.

Function groups 6, 7: *N* refers to an instruction address in the peg store.

When *C*=1, the interpretation in groups 2 and 4 is changed slightly, *N* being the address of a 12-bit constant in the peg store. Since two such constants occupy one instruction location, the constant address is twice that of the corresponding instruction address, the appropriate half being indicated by adding 0 (for the m.s. half) or 1 (l.s. half).

Constants may be stored in the first 2048 instruction locations (where these are supplied), giving a maximum of 4096 constants. The *C* bit may be used with functions 6 and 7 as an optional stop, for development work.

Normal allocation of *N* addresses for *C*=0 and function groups 0–2 is as follows:

0000 – 0007	Accumulators
0010	Link register
0011	Hoot
0013	Timer
0014 – 0015	Handswitches
0016 – 0017	C.S.I. Limits
0021	Input
0025	Carry (l.s. bit)
0026	Overflow (m.s. bit)
0030 – 0031	Isolation register (l.s. bit)
2000 – 7777	C.S.I. Control
	Outputs and Core Store.

24-bit numbers occupy two locations, an even address followed by an odd address, the latter being used to call the number into a 24-bit accumulator.

Accumulator *X*

There are eight accumulators, numbered 0–7, of which 0 contains zero permanently.

Accumulators 0–5 are numbered according to their *N* addresses, and are 12 bits long. Accumulators 6 and 7 are 24 bits long, 6 consisting of registers 0006 and 0007, and 7 of 0004 and 0005. The letters *P* and *Q* are also used to denote accumulators 7 and 6 respectively, *P* being further subdivided into *P*<sub>1</sub> and *P*<sub>2</sub> (accumulators 4 and 5).

<i>N</i> address	<i>X</i> address
0000	0
0001	1
0002	2
0003	3
0004	4 ( <i>P</i> <sub>1</sub> )
0005	5 ( <i>P</i> <sub>2</sub> )
0006	7 ( <i>P</i> )
0007	6 ( <i>Q</i> )

Function *F*

The complete function code is appended in the table below.

Modifier *M*

Accumulators 0–3 can be used for modification in the *M* position; the number in *M* is added to the address *N* before the order is obeyed.

### Function Code

Lower case letters indicate contents of the address or accumulator, denoted by corresponding capital letters before the order is obeyed. Values after the order is obeyed are indicated by a prime (').  $N$  and  $n$  refer to contents of  $N$  address locations after modification. In orders of group 5,  $N$  is regarded as an integer.

00 $x' = n$	10 $n' = x$
01 $x' = x + n$	11 $n' = n + x$
02 $x' = -n$	12 $n' = -x$
03 $x' = x - n$	13 $n' = x - n$
04 $x' = n - x$	14 $n' = n - x$
05 $x' = x \& n$	15 $n' = n \& x$
06 $x' = x \# n$	16 $n' = n \# x$
07	17
12-bit numbers ( $X=0$ to 5)	24-bit numbers ( $X=6$ or 7)
20 $p' = n.x$	$(pq)' = n.x$
21 $p' = n.x + 2^{-12}$	$(pq)' = n.x + 2^{-24}$
22	
23 $p'_2 + 2^{-11} \frac{p'_1}{n} = \frac{(xp_2)}{n}, 0 \leq \frac{p'_1}{n} < 1$	$q' + 2^{-23} \frac{p'}{n} = \frac{(pq)}{n}, 0 \leq \frac{p'}{n} < 1$
24 $p'_2 + 2^{-11} \frac{p'_1}{n} = \frac{(xp_2)}{n}, -\frac{1}{2} \leq \frac{p'_1}{n} < \frac{1}{2}$	$q' + 2^{-23} \frac{p'}{n} = \frac{(pq)}{n}, -\frac{1}{2} \leq \frac{p'}{n} < \frac{1}{2}$
25 As 24 but with zero for $p_2$	As 25 but with zero for $q$ .
26	
27	
Optional input instructions	
30 Call $n$ to 0021 (single bits)	40 $x' = N$
31 " " " (c.p. digitisers)	41 $x' = x + N$
32 " " " (analogues)	42 $x' = -N$
33 " " "	43 $x' = x - N$
34 " " "	44 $x' = N - x$
35 " " "	45 $x' = x \& N$
36 " " "	46 $x' = x \# N$
37 " " "	47
50 $x' = x.2^N$ ( $x'$ rounded when $X=5$ to 7, $-64 \leq N < 64$ )	
51 Shift $x$ logically $N$ places (up if $N > 0$ , down if $N < 0$ ).	
52 $p' = p.2^N$ , $X=4$ , or $(pq)' = (pq).2^N$ , if $X=7$ , unrounded.	
53 Shift $p$ , if $X=4$ , or $(pq)$ if $X=7$ , logically $N$ places.	
54 $p' = p.2^\mu, x' = x - \mu$	Normalise:
55 $(pq)' = (pq).2^\mu, x' = x - \mu$	Either (i) $\mu=0$ , and $p=0$ Or (ii) $-1 \leq \mu \leq 46$ , and $-\frac{1}{2} \leq p' < \frac{1}{2}$ or $\frac{1}{4} \leq p' < \frac{1}{2}$
56	
57	
60 Jump to $N$ if $x=0$	70 Jump to $N$ and store link
61 Jump to $N$ if $x \neq 0$	71 Jump to link+1
62 Jump to $N$ if $x \geq 0$	72 Jump to link (after interrupt)
63 Jump to $N$ if $x < 0$	73 Jump if $X$ busy
64 Jump to $N$ if OVR clear	74 Jump if $X$ not busy
65 Jump to $N$ if OVR set	75
66 $x' = x - 2, m' = m + 2$	76 Obey the contents of $N$
67 $x' = x - 1, m' = m + 1$	77 STOP.

### Speed

Groups 0, 1, 3, 4: 20  $\mu$ sec. short, 40  $\mu$ sec. long.

Functions 60 – 65, 70 – 74: 20  $\mu$ sec.

Functions 66 – 67: 40  $\mu$ sec.

Function 76: 40  $\mu$ sec. + execution time of order.

Function 20: maximum 120  $\mu$ sec. short, 400  $\mu$ sec. long.

Function 21: maximum 140  $\mu$ sec. short, 440  $\mu$ sec. long.

Function 24: maximum 180  $\mu$ sec. short, 640  $\mu$ sec. long.

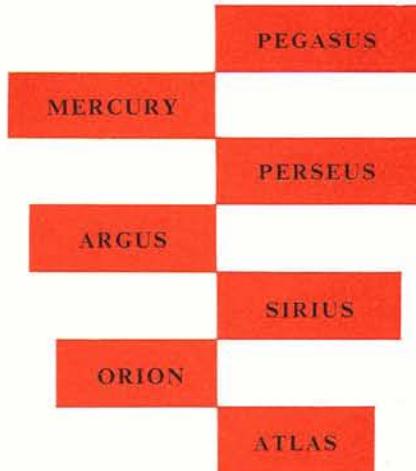
Functions 25, 26: maximum 200  $\mu$ sec. short, 680  $\mu$ sec. long.

Functions 50–53: maximum 140  $\mu$ sec.

Functions 54, 55: maximum 120  $\mu$ sec.

Modification on any order adds a further 20  $\mu$ sec. short, 40  $\mu$ sec. long.

*Ferranti  
range of  
Computer  
systems*



*Enquiries to  
FERRANTI LTD  
LONDON COMPUTER CENTRE  
68-71 NEWMAN STREET, LONDON W1  
Telephone MUSeum 5040*

*or WYTHENSHAWE, MANCHESTER 22  
Telephone MERcury 5291*

*or 21 PORTLAND PLACE, LONDON W1  
Telephone LANgham 9211*

*Works WEST GORTON, MANCHESTER 12  
Telephone EASt 1301*

**Ferranti**

**Ferranti**