

PERQ

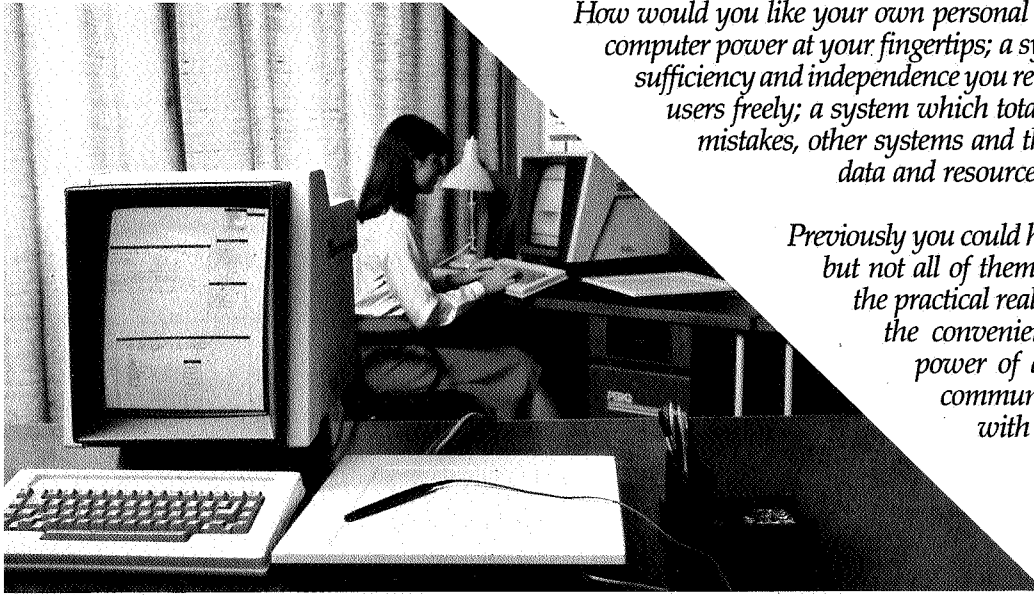


ICL helps shape the future

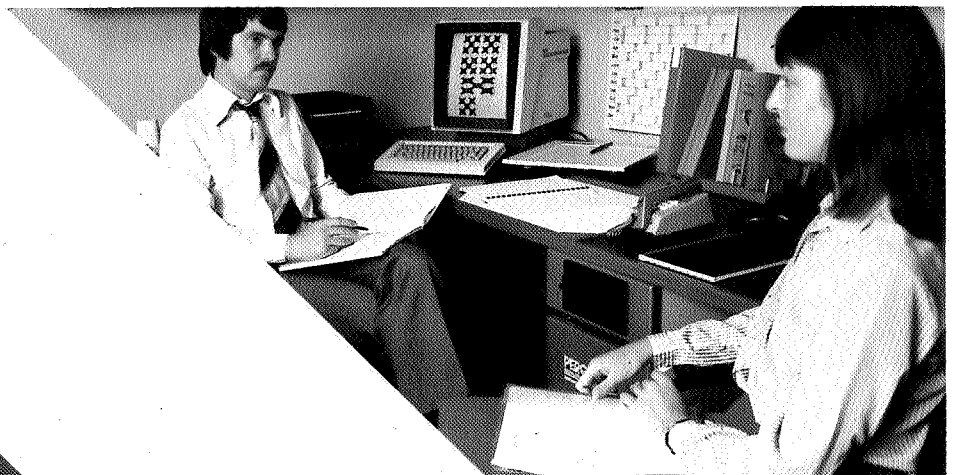
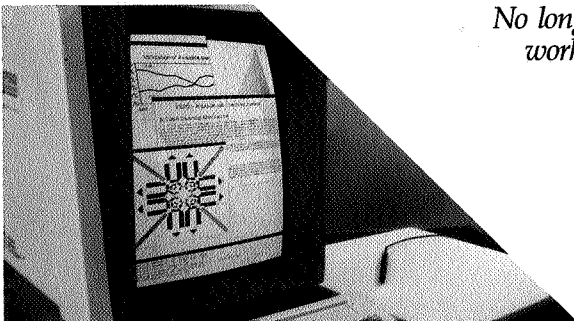
The advent of the personal workstation

How would you like your own personal computer system at your desk; exceptional computer power at your fingertips; a system powerful enough to give you the self-sufficiency and independence you require; the ability to communicate with other users freely; a system which totally protects you from other users and their mistakes, other systems and their failures; the option to share programs, data and resources whenever applicable; instant computing.

Previously you could have had some of these features individually, but not all of them together. However, ICL now brings you the practical reality – a computer resource that combines all the convenience of a desk-top microcomputer, all the power of a dedicated minicomputer system, all the communications advantages of a time-shared system with all the quality and facilities of an interactive graphics terminal.

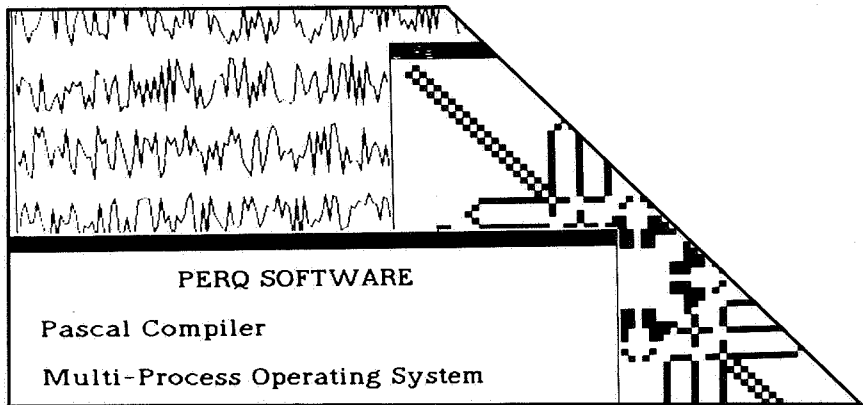
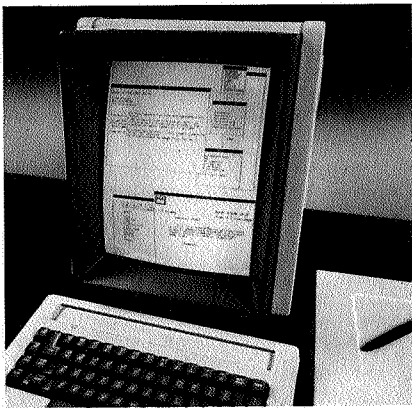


No longer will you be denied machine time when you need it most; the personal workstation provides you with all you want from a computer system – when you need it, how you need it – to meet your present and future requirements.



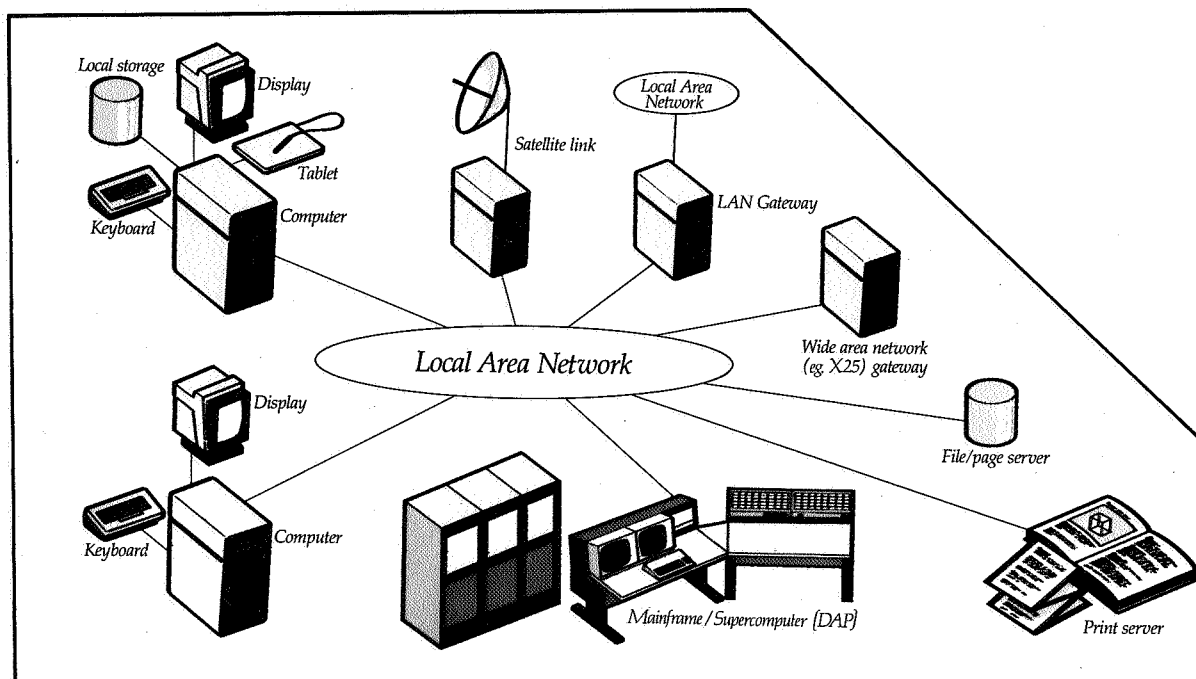
The PERQ professional workstation

PERQ is an extremely powerful, single user workstation which exploits the latest developments in technology to bring unprecedented levels of computing power to you at your desk. The workstation provides ultra high performance raster graphics of superb quality, and each PERQ, whilst entirely self-sufficient, provides connection to a high speed Local Area Network for communication with other workstations and access to shared resources.



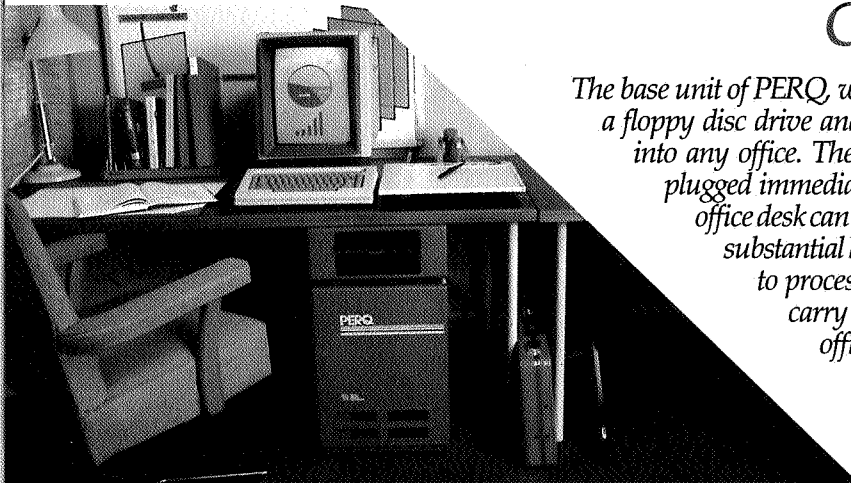
PERQ enables you to solve problems more quickly and more effectively by giving you:

- Instant access to your machine and your data
- Uninterrupted use of your own computing facilities
- Faster turnround by virtue of the dedicated power of a single user computer
- Unparalleled quality of interaction in the form of both hardware and software tools



Computing power of your own

The base unit of PERQ, which contains the powerful CPU, up to 1 megabyte of store, a floppy disc drive and a 24 megabyte fixed disc, is designed to fit unobtrusively into any office. The unit can easily be housed beneath a desk top and can be plugged immediately into a standard mains outlet. This means that a typical office desk can be transformed into a powerful computing work-station with substantial local storage. Self sufficiency becomes a reality. With an ability to process up to 1 million Q-codes per second PERQ enables you to carry out the computing you require when you like in your own office. If you wish to tailor PERQ to a particular application, a valuable feature of the system – a writable control store – enables you to enhance the existing instruction set or implement an entirely different one to match your specific needs.

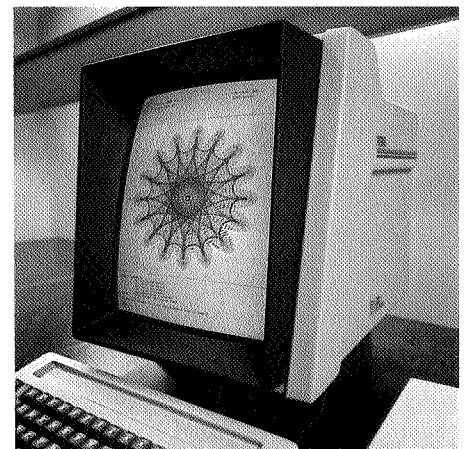
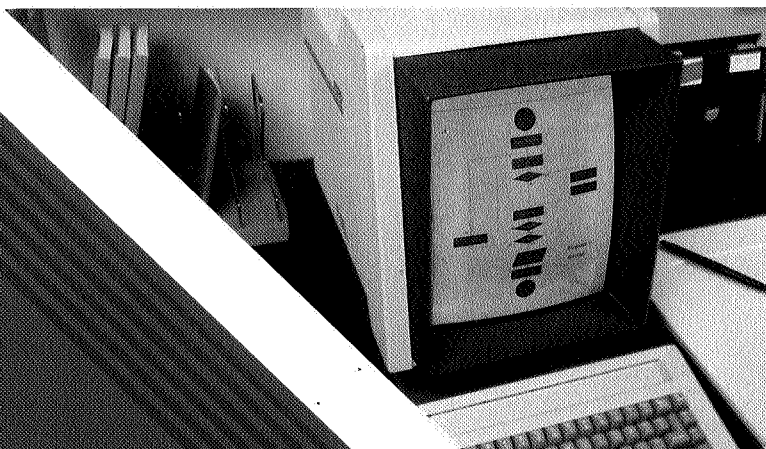
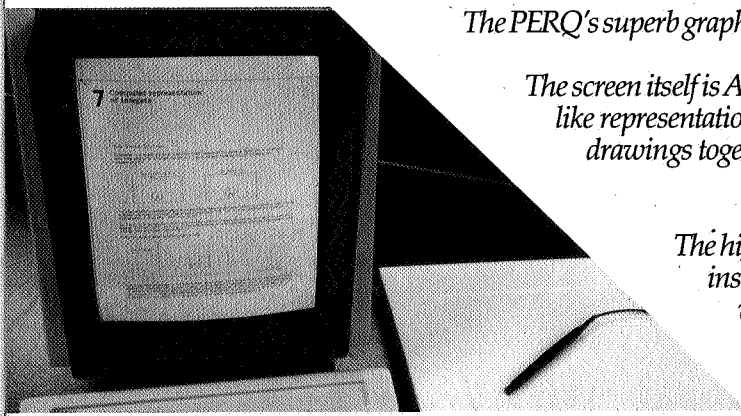


Living graphics

The PERQ's superb graphics features are supported by special and extremely powerful hardware.

The screen itself is A4 in size which is the natural shape to display your documents with life-like representation. In addition, just as in the real world documents contain a mixture of drawings together with characters of different fonts, so too can the PERQ display an accurate reproduction of that information.

The high performance of the graphics is brought about by additional hardware instructions ('RasterOp') which eliminate the delays usually associated with Raster-scan displays. These enable all or part of the display image to be changed by a single instruction. This, together with the flicker-free, high refresh rate display means that real animation is now possible.



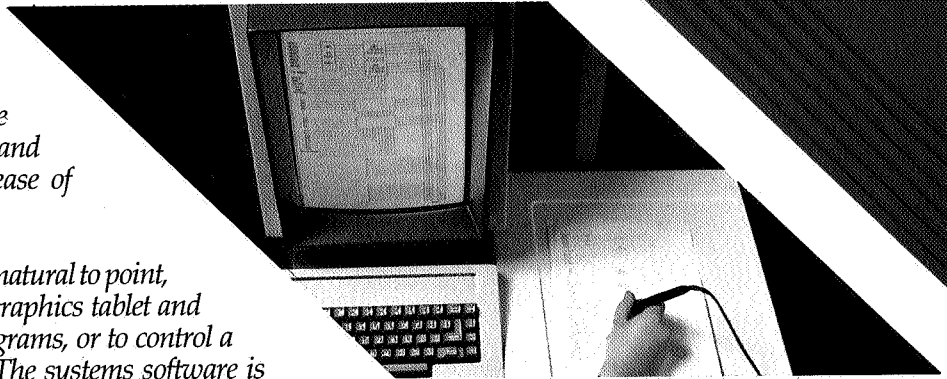
A natural facility

Even for the experienced user, interacting with a computer via a keyboard can be a time-consuming and unnatural function. The PERQ keyboard is therefore enhanced with extra keys, such as 'OOPS' and 'HELP', which are used to improve the ease of interaction.

To interact with the display it is much more natural to point, and this is achieved by the provision of a graphics tablet and stylus. This can either be used to input diagrams, or to control a hardware cursor displayed on the screen. The systems software is designed to exploit this more natural method of interaction. The editor, for example, enables you to select areas of text, to scroll up or down, or position the display within the file, all by a simple depression of the stylus.

Similarly, menus of commands can be displayed and selected in this manner.

The special graphics features mean that the screen may be divided into separate areas, or windows, of any size, each of which may be regarded as a virtual screen.



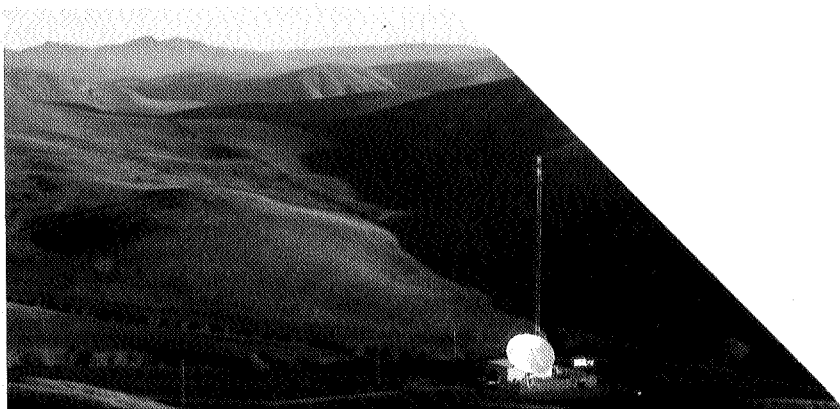
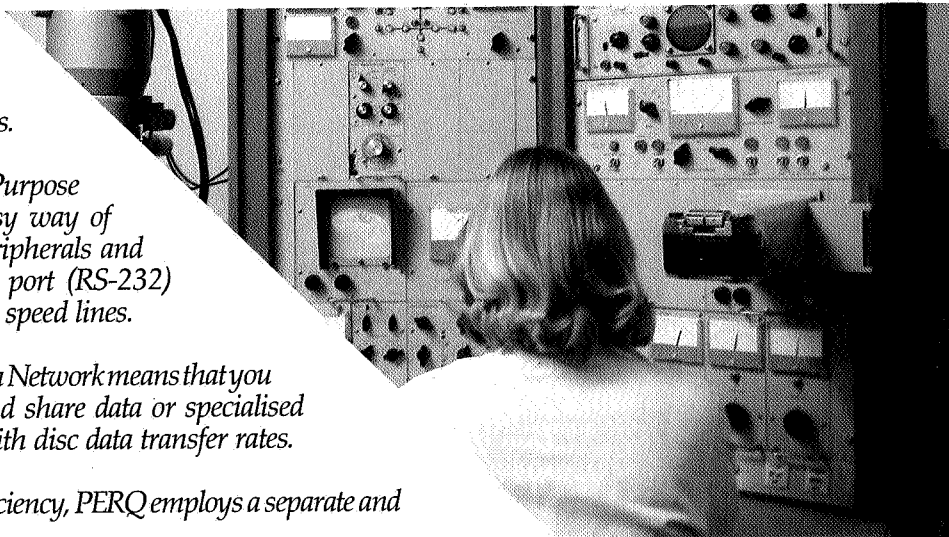
The right connections

PERQ not only offers stand-alone features. In addition, PERQ provides you with standard interfaces and the ability to communicate and share resources with other PERQ users.

A full standard implementation of the General Purpose Instrumentation Bus (GPIB) provides an easy way of interfacing a wide range of medium speed peripherals and laboratory equipment, while a standard serial port (RS-232) enables communications over slow to medium speed lines.

Moreover, the option of connecting to a Local Area Network means that you can communicate with other PERQ users, and share data or specialised resources with them, at a speed comparable with disc data transfer rates.

To handle these interfaces without impairing efficiency, PERQ employs a separate and dedicated processor for input/output.



PERQ Specification

General Information

Power requirements

Source: 220/240V AC

Line frequency: 50 Hz

Power consumption: 720 watts

Environmental requirements

Normal office environment

Physical dimensions

Base unit – height: 670mm width: 360mm depth: 670mm weight: approx. 50Kg

Display – height: 480mm width: 330mm depth: 480mm weight: approx. 15Kg

Hardware Specification

The basic PERQ system consists of:

Processor

Microprogrammed bit-sliced 16-bit CPU

High-level language directed architecture

Integrated input/output controllers

Speed: up to 1 million Q-codes per second (Q-code is variant of P-code)

4K Writable Control Store at 170ns cycle time

Memory

256 Kbytes - 1 Mbyte of 680ns RAM with parity checking

Display

Free-standing

Screen diagonal: 350mm

Raster size – height: 275mm width: 210mm

Manually adjustable screen brightness

60Hz refresh rate

P-104 white tube phosphor

Manually adjustable elevation: $\pm 5^\circ$

Screen capacity: 1024 \times 768 pixels

Dot spacing: 0.27mm

Maximum distance from base unit: 2.5m

Character cursor selectable as any character under software control

Hardware cursor user definable, up to 64 \times 64 pixels

Keyboard

Free-standing

ASCII character code

N-key rollover

Auto-repeat

Electronic shiftlock

Selectric layout + extra function keys

Maximum distance from base unit: 2.5m

Graphics tablet

Free-standing

Overall dimensions: 395 × 395 × 45mm

Active area: 280 × 280mm

Magnetostrictive operation

Power requirements: 220/240V AC

Resolution: 0.127mm

Fixed disc unit

14" Winchester rigid disc

24 Mbytes formatted capacity

2 discs, 8 heads (2 per surface)

Rotational speed: 2964rpm

Transfer rate: 887.5 Kbytes/sec

Average latency: 10.1ms

Track-to-track seek: 20ms

Floppy disc drive

8" double-sided, double density

77 tracks/side, 26 sectors/track

1 Mbyte formatted capacity

Rotational speed: 360rpm

Transfer rate: 62.5 Kbytes/sec

Head load time: 35ms

Settling time: 8ms

Latency: 83.3ms

Track-to-track seek: 8ms

GPiB interface

Full IEEE 488-1975 standard implementation of the

General Purpose Instrumentation Bus

RS-232-C interface

Full duplex, serial data port

Supports asynchronous and synchronous communications

Speeds: up to 56 Kbits/sec

Programmable speed and data format

Software Specification

Operating system

Supports multiple process, virtual memory system

Display window manager

Symbolic debugger

Editor

Linker

Dis-assembler

File manager

Network support

Pascal compiler

The PERQ Pascal is an upward compatible extension of the programming language defined in the 'Pascal User Manual and Report' by Jensen and Wirth.



International Computers Limited

World Headquarters
ICL House Putney
London SW15 1SW England


Every effort has been made to ensure the accuracy of this document and of the information and statements herein. The equipment or software performance figures quoted are nominal and (within acceptable tolerances) are those which ICL expects to achieve under test conditions.

ICL points out, however, that the attainment of any stated performance or other figures is affected in operational use by such factors as the conditions under which the system is operated, the efficiency of any software (or modifications thereto) with which the Customer provides himself and the adoption of any advice, recommendations and operating instructions given by ICL.

The policy of ICL is one of continuous development and improvement of its products and services, and the right is reserved to amend or alter the specification of the system described in this document; but wherever practicable, and at the Customer's request, ICL will before contract, confirm to the Customer the accuracy of any statement upon which the Customer wishes to place particular reliance.

© International Computers Limited 1981

Published by Corporate Communication Division, Putney
St. 1037 Printed in England 3M 85USP

<p>000 LDX $x' = n + c$ V 001 ADX $x' = x + n + c$ V 002 NGX $x' = -n - c$ V 003 SBX $x' = x - n - c$ V 004 LDXC $x' = n + c$ C 005 ADXC $x' = x + n + c$ C 006 NGXC $x' = -n - c$ C 007 SBXC $x' = x - n - c$ C</p>	<p>070 CALL Subroutine Entry Link in X 072 EXIT Subroutine Exit V Link in X 074 Conditional Branch to N:- BRN Branch unconditionally BVS Branch if V is set BVSR Branch if V is set and clear V BVC Branch if V is clear BVCR Branch if V is clear or clear V BCS Branch if C is set BCC Branch if C is clear BVCI Branch if V is clear and/or invert V V ‡076 Test floating point accumulator</p> <p>X = 0 X = 1 X = 2 X = 3 X = 4 X = 5 X = 6 X = 7</p>	<p>* 130 FLOAT Convert n: from fixed to floating * 131 FIX Convert a from floating to fixed V ** 132 FAD $a' = a + n$: ** 133 FSB $a' = a - n$: ** 134 FMPY $a' = a \cdot n$: ** 135 FDVD $a' = a/n$: ** 136 LFP $a' = n$: ** 137 SFP $n' = a$: If $X = 1$, Unrounded $X = 2$, Not normalized $X = 4$, Interchange a and n If $X = 1$, $a' = 0$ If $X = 1$, $n' = a$, $a' = 0$</p>
<p>010 STO $n' = x + c$ V 011 ADS $n' = n + x + c$ V 012 NGS $n' = -x - c$ V 013 SBS $n' = n - x - c$ V 014 STOC $n' = x + c$ C 015 ADSC $n' = n + x + c$ C 016 NGSC $n' = -x - c$ C 017 SBSC $n' = n - x - c$ C</p>	<p>100 LDN $x' = N + c$ 101 ADN $x' = x + N + c$ V 102 NGN $x' = -N - c$ 103 SBN $x' = x - N - c$ V 104 LDNC $x' = N + c$ 105 ADNC $x' = x + N + c$ C 106 NGNC $x' = -N - c$ C 107 SBNC $x' = x - N - c$ C</p>	<p>* 150 X N(M) SUSBY Suspend if peripheral N(M), unit X, is active- * 151 X N(M) REL Release peripheral N(M), unit X <i>disengage</i> * 152 X N(M) DIS Disengage peripheral N(M), unit X * 153 X N(M) Unassigned * 154 X N(M) CONT Read more program from peripheral N(M), unit X * 155 X N(M) SUSDP Suspend and dump program on peripheral N(M), unit X * 156 X N(M) ALLOT Allocate peripheral N(M), unit X, to the program * 157 X N(M) PERI Initiate peripheral transfer according to control area N(M), unit X</p>
<p>020 ANDX $x' = x \& n$ 021 ORX $x' = x \vee n$ 022 ERX $x' = x \neq n$ 023 OBEY Obey the instruction in N 024 LDCH $x' = n_j$ 025 LDEX $x' = n_e$ 026 TXU Set C if $n \neq x$ or $c = 1$ 027 TXL Set C if $n + c > x$</p>	<p>110 SLC Shift x left N_s places. Circular SLL Shift x left N_s places. Logical SLA Shift x left N_s places. Arithmetic V 112 SRC Shift x right N_s places. Circular SRL Shift x right N_s places. Logical SRA Shift x right N_s places. Arithmetic SRAV Shift x right N_s places. Special V ⊙ 114 NORM Normalize x ‡ 116 MVCH Transfer N characters</p> <p>$N_t = 0$ $N_t = 1$ $N_t = 2, 3$ $N_t = 0$ $N_t = 1$ $N_t = 2$ $N_t = 3$</p>	<p>* 160 0 N(M) SUSTY Suspend and type message on console typewriter * 160 1 N(M) DISTY Type message on console typewriter without suspension * 160 2 N(M) DELTY Delete program and treat message as console directive * 161 0 NN(M) SUSWT Suspend and type HALTED NN on the console typewriter * 161 1 NN(M) DISP Type DISPLAY NN on the console typewriter without suspension * 161 2 NN(M) DEL Delete program and type DELETED NN on the console typewriter * 162 X 0 SUSMA Suspend if subprogram X is active * 163 X N(M) AUTO Activate and enter subprogram X at N(M) * 164 0 0 SUSAR De-activate the current subprogram * 165 X N(M) GIVE If N(M) = 0, X will contain date in binary If N(M) = 1, XX* will contain date in character form If N(M) = 2, XX* will contain time in character form If N(M) = 3, X will contain core store allocated to this program</p>
<p>⊙ 040 MPY $x_i' = n \cdot x$ V ⊙ 041 MPR $x_i' = n \cdot x$ rounded, x^* spoiled V ⊙ 042 MPA $x_i' = n \cdot x + x^*$ V ⊙ 043 CDB $x_i' = 10 \cdot x_i + n_j$ V ⊙ 044 DVD $x^* = x/n$, x' = Remainder V ⊙ 045 DVR $x^* = x/n$ rounded, x' = Remainder V ⊙ 046 DVS $x^* = x^*/n$, x' = Remainder V ⊙ 047 CBD $x_i' = 10 \cdot x_i$, $n_j' =$ Character V</p>	<p>⊙ 111 SLC Shift x: left N_s places. Circular SLL Shift x: left N_s places. Logical SLA Shift x: left N_s places. Arithmetic V ⊙ 113 SRC Shift x: right N_s places. Circular SRL Shift x: right N_s places. Logical SRA Shift x: right N_s places. Arithmetic SRAV Shift x: right N_s places. Special V ⊙ 115 NORM Normalize x ‡ 117 SMO Supplementary modifier to next instruction</p> <p>$N_t = 0$ $N_t = 1$ $N_t = 2, 3$ $N_t = 0$ $N_t = 1$ $N_t = 2$ $N_t = 3$</p>	<p>Notes The function codes 140 to 147 are undefined. C These instructions may set the carry register but cannot cause overflow. The carry register C is left clear by any order except 023 and 123, unless that order sets C. V These instructions may cause overflow. ⊙ These instructions are performed on 1902, 1903 by extracode and by hardware on the other machines. ** These instructions are performed on 1902, 1903, 1904 by extracode and by hardware on the other machines. * These instructions are performed by extracode on all machines.</p>
<p>050 BZE Branch to N if $x = 0$ 052 BNZ Branch to N if $x \neq 0$ 054 BPZ Branch to N if $x > 0$ 056 BNG Branch to N if $x < 0$ 060 BUX Single word modify: $x_m' = x_m + 1$ 062 BDY Double word modify: $x_m' = x_m + 2$ 064 BCHX Character modify: $x_m' = x_m + .1$ ‡ 066 BCT Count least significant 15 bits of X. $x_m' = x_m - 1$ Branch to N if $x_m' \neq 0$</p>	<p>120 ANDN $x' = x \& N$ 121 ORN $x' = x \vee N$ 122 ERN $x' = x \neq N$ 123 NULL No operation 124 LDCT $x_c' = N$, $x_m' = 0$ 125 MODE Set mode N ⊙ 126 MOVE Transfer N words from address x to address x^* ⊙ 127 SUM $x' =$ Sum of N words from address x^*</p>	<p>† These facilities are not available on 1902, 1903 processors with less than 16 K store. ‡ These instructions are available on 1906, 1907 processors only.</p>
<p>INTERNATIONAL COMPUTERS AND TABULATORS LTD.  I. C. T. 1900 SERIES PLAN - SUMMARISED PROGRAMMING INFORMATION FORM 11/129 (1.67) DATE OF PUBLICATION - MARCH 1965</p>		<p>TECHNICAL SERVICES, 113 UPPER RICHMOND RD., PUTNEY, LONDON, S.W.15. © International Computers and Tabulators Ltd.</p>

NOTATION

N is a core store address or a 12 bit number.

X is an accumulator (registers 0-7).

M is a modifier register (registers 1-3).

F is a function.

C is the carry register.

c is the content of C (0 or 1).

V is the overflow register.

A is the floating point accumulator.

a is the content of A .

x, m are the contents of X, M respectively.

n is the content of N after modification by m if necessary.

X^* is the accumulator $X + 1$ ($X7^* = X0$)

x^* is the content of X^* .

x', n', a' are the contents of X, N, A after an instruction has been obeyed.

x, n : are double length numbers in $X, X + 1$, and $N, N + 1$ respectively.

S is the sign bit (bit 0).

The most significant bit of the second word of a double length number is always zero.

Subscripts

In general these are applicable to x or n .

x_e is the least significant 9 bits of x . The exponent of a floating point number occupies this portion of the second word.

x_a is the least significant 12 bits of x (the N address of an instruction).

x_c is a 9 bit counter at the most significant end of x .

x_m is the least significant 15 bits of x (the modifier part of an index register).

x_h is the most significant 2 bits of x , used in character modifying with end-around - carry to x_m .

x_d is the least significant 7 bits of x_c .

x_j is any one of x_0, x_1, x_2, x_3 , the four 6-bit characters of x .

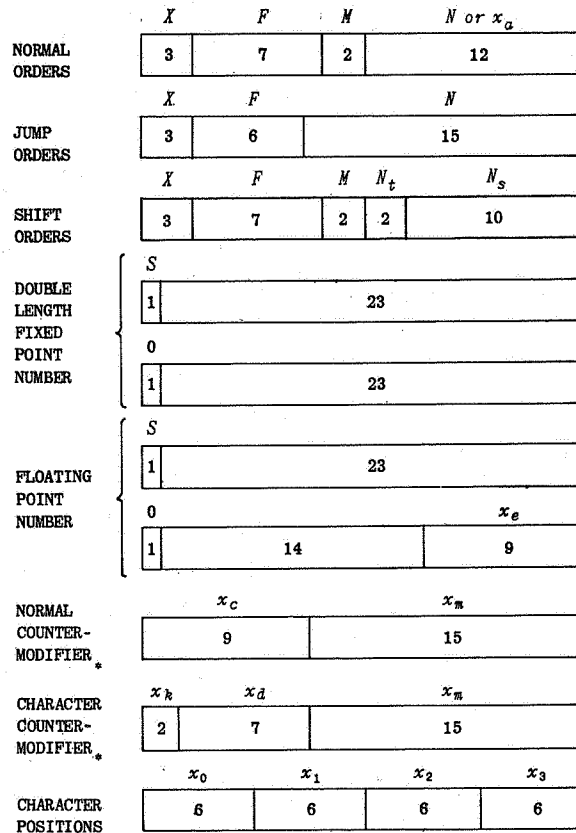
N_t is the most significant 2 bits of the 12 bit N address.

N_s is the least significant 10 bits of the 12 bit N address.

Note: -

* When in extended mode (1906 and 1907 only) the modifier extends to 22 bits, the count being held separately.

24-bit I.C.T. 1900 word



MAJOR DIRECTIVES

The appearance of any directive in this group cancels the effect of any previous directive in the group.

- # PROGRAM - introduces a section of program instructions
 - # LOWER - introduces lower data (below location 4096)
 - # UPPER - introduces upper data (not Plan 1)
 - # PERIPHERAL - is followed by specification of peripherals (other than magnetic tapes)
 - # MACRO - indicates that a description of a private macro follows (Plan 3 only)
 - # END - the last statement of a segment; ends compilation
 - # FINISH - indicates that this is the last segment to be compiled
- PLAN 1 only
- # COMPLETE - indicates that the program is to be output in consolidated form.

MACRO INSTRUCTIONS (PLAN 3 ONLY)

INSTRUCTION	EFFECT	NO. OF BASIC INSTRUCTIONS	
LDX XX* N(M)	$x' = n$	2	
ADX XX* N(M)	$x' = x + n$	2	
NGX XX* N(M)	$x' = -n$	2	
SBX XX* N(M)	$x' = x - n$	2	
STO XX* N(M)	$n' = x$	2	
ADS XX* N(M)	$n' = n + x$	2	
NGS XX* N(M)	$n' = -x$	2	
SBS XX* N(M)	$n' = n - x$	2	
BXU X $N_1(M), N_2$	If $x \neq n_1$ jump to N_2	2	
BXU XX* $N_1(M), N_2$	If $x \neq n_1$: jump to N_2	3	
BXE X $N_1(M), N_2$	If $x = n_1$ jump to N_2	2	
BXE XX* $N_1(M), N_2$	If $x = n_1$: jump to N_2	3	
BXL X $N_1(M), N_2$	If $x < n_1$ jump to N_2	2	
BXL XX* $N_1(M), N_2$	If $x < n_1$: jump to N_2	3	
BXGE X $N_1(M), N_2$	If $x \geq n_1$ jump to N_2	2	
BXGE XX* $N_1(M), N_2$	If $x \geq n_1$: jump to N_2	3	
LDSA X N(M)	$x' = n_a$	2	
LDLA X N(M)	$x' = n_m$	2	
LDPL X N	$x' = N(15 \text{ bits})$	1	
TAPE MACROS	WTM X	Write tape mark on MTX	1
	REW X	Rewind MTX	1
	BSP X	Backspace MTX	1
	BTM X	Move back past tape mark on MTX	1
	FTM X	Move forward past tape mark on MTX	1
	CLOSE X	Close MTX	1
	SCR X	OPEN MTX and leave scratch	1
UNL X	Close file and unload	1	

PROGRAM AREA DIRECTIVES

These directives appear in PROGRAM area only.

- # CUE - gives a label to the following instruction for use by all segments
- # ENTRY - makes the following instruction entry point N, where N is written in the operand field
- # MONITOR - introduces specification of monitor printing

GENERAL PURPOSE DIRECTIVES

The directives may appear anywhere in the program

- # SET - used to define a name (may be reset)
- # DEFINE - used to define a name (may not be redefined)
- # - used for writing comments
- # PAGE - causes paper throw on printer.