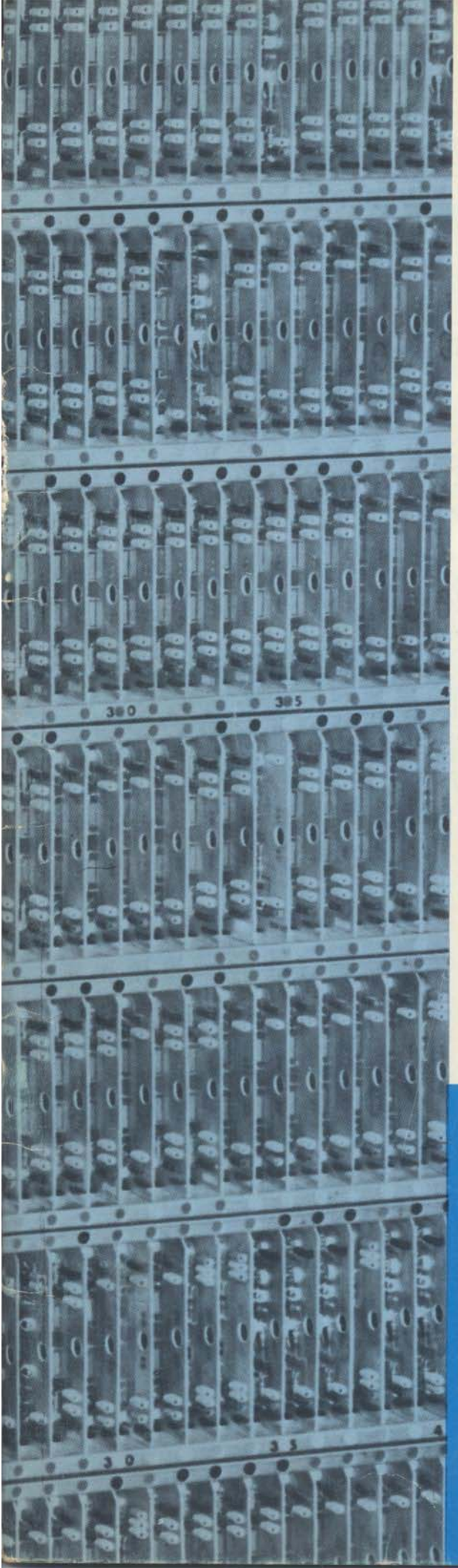


the FP6000 computer system

FERRANTI FP PACKARD

FERRANTI-PACKARD ELECTRIC LIMITED

Introduction

The FP6000 is a fast medium-size computer system, constructed in a packaged, modular form. At the centre of the system is a general purpose digital computer with a minimum core store of 4096 words, which may be increased in 4096 word modules to 32,768 words. The store cycle time can be either 2 or 6 microseconds. A large variety of peripheral devices can be added to the computer, making the system very versatile and adaptable to a wide range of commercial data processing, scientific and real time applications. Due to the modular method of attaching peripherals a particular system can be expanded at any future time to meet increasing demands and wider applications. As well as orthodox peripheral devices such as magnetic tape, punched cards and line printers, the system has been designed to facilitate the inclusion of specialized equipment — for example: field equipment connected to the central computer by data transmission communication lines.

An important feature of the system is the comprehensive time-sharing scheme which enables several programs to run concurrently with no risk of interference. Time-sharing is organized by a program called EXECUTIVE and is quite automatic; the programmer does not have to consider it when writing his programs. Time-sharing virtually eliminates the need for special purpose off-line equipment such as card-to-tape converters since the time spent by one program in waiting for a slow peripheral is utilized by another program. EXECUTIVE controls all peripheral transfers and the allocation of time between programs and also provides macro-instruction facilities such as the floating point routines.

The order-code of the FP6000 was designed by collaboration between engineers and programmers. The Ferranti Companies have many years of experience in the digital computer field, and the order-code is a development of previous well-tried and successful order-codes, combining ease of programming with a comprehensive code. The basic input routine accepts machine code programs punched on paper tape or cards.

It is expected that most programmers will take advantage of the ASSEMBLER scheme which enables programs to be written in a simplified form, with mnemonic function codes, symbolic addresses and tags.

FP6000

System highlights

* High Speed
* Autonomous Peripheral Transfers
* Simultaneous Program Capability
* Time Sharing with Full Program Protection
* Comprehensive Instruction Code
* Ease of Programming
* Powerful Symbolic Programming Scheme
* Algebraic Compiler
* Monitoring and Diagnostic Facilities
* Modular Construction
* Adaptability for Special Applications.

Central computer

The central computer consists of a basic arithmetic and control unit, to which must be added at least one block of 4096 words of core store and some input/output equipment to make up a minimum system.

Information comes into the system and leaves it in the form of alphanumeric characters and alphabetic information is also stored in this way. However all the internal arithmetic is carried out in binary form. The provision of suitable input and output programs and of automatic programming systems almost completely conceals the internal number representation from the programmer who does not wish to concern himself with it.

The computer operates with words of 24 binary digits. Such a word can be used to represent an instruction, a signed integer or fraction, four 6-bit alphanumeric characters; two words together may represent a double length number or a floating point number. Macro-instructions provide floating point operations and the order-code has been chosen so as to facilitate both floating point and multiple precision working.

The computer is provided with a working store which is a ferrite core matrix with a cycle of 2 or 6 microseconds and which has a capacity ranging from 4096 words to 32,768 words. This store is made up of registers each holding one word of 24 bits. A 25th bit is attached to each register to provide a parity check; the parity of each word is calculated and stored with the word when it is written and is automatically checked whenever the word is used. The store is not divided into blocks or sections in any way. The first eight core store registers assigned to each program are used as the program's accumulators which can be used for arithmetic and counting. Accumulators 1, 2 and 3 may further be used for indexing. Overflow and carry registers are also provided.

A comprehensive repertoire of instructions is provided which in general specify one register and one accumulator. Some typical operating times are:

Addition or Subtraction	7 microseconds
Multiplication	40 microseconds
Division	43 microseconds
Programmed Floating Point Addition	111 microseconds

The console provides the means of communication with the central computer and consists of an operator's typewriter and other control keys.

The operator can communicate with EXECUTIVE by typing messages on the keyboard. A copy of the message will be printed as it is typed and EXECUTIVE will not take action on a message until the operator has checked it and depressed the ACCEPT key. Similarly EXECUTIVE may communicate with the operator by printing out messages. The typewriter makes operation of the system very simple since all communications can be conducted in a standard set of English phrases which can be understood by EXECUTIVE and by the operator.

FP6000

Peripheral Equipment

Number	Description	
FP6010	Control Console with Typewriter	
FP6011	Enquiry Station	
FP6110	Paper Tape Reader	385 characters/second
FP6130	Paper Tape Punch	110 characters/second
FP6200	4096 Words of Core Store	6 microseconds cycle
FP6210	4096 Words of Core Store	2 microseconds cycle
FP6310	Punched Card Reader	
FP6330	Punched Card Punch	
FP6410	Printer	300 lines/minute
FP6411	Printer	1000 lines/minute
FP6420	Audit Printer	
FP6421	Pocket Printer	
FP6500	Magnetic Tape/Drum Control	For FP6510
FP6501	Magnetic Tape/Drum Control	For FP6512
FP6510	Magnetic Tape Unit	Atlas/Orion compatible
FP6512	Magnetic Tape Unit	62.5 Kc.
FP6520	Magnetic Tape Unit Exchange	
FP6530	Magnetic Drum Unit	163,840 words
FP6531	Magnetic Drum Unit	32,768 words
FP6615	MICR Document Sorter	

Peripheral Transfers

All peripheral transfers are carried out by means of program entries to EXECUTIVE. Since the transfers are autonomous and each type of peripheral device has its own control unit, the computer is able to continue with other operations while the transfer is effected. An instruction is provided which tests whether or not the peripheral transfer is complete and a program can suspend itself if it has reached the end of the work which it can usefully or safely do before the transfer is completed. A part of EXECUTIVE is then entered to determine how this time may be used by another program.

From time to time during a peripheral transfer the control unit concerned will require access to the working store to extract or insert a single word or character. When this occurs the computer is made to hesitate, i.e. it is held up while the peripheral control has access to the store. These hesitations may slow the computer slightly. The effect is significant only when the fastest devices are in operation, but even with magnetic tape the speed reduction is less than 15%. The importance of the hesitation principle is that all peripheral transfers are buffered using the working core store without the need for expensive special purpose buffering equipments.

Peripheral transfers may start at any location in the working store (provided always that program limits are not violated) and, where appropriate, may be of variable length in words or characters. Two consecutive instructions are usually necessary to implement a peripheral transfer — one to specify the function, unit number and core store address, the second to specify the number of words or characters to be transferred and also the mode of operation when applicable.

The standard available peripheral equipment is listed opposite. Special equipment may be attached to FP6000; thus it is easy to provide for applications requiring a combination of special purpose and general purpose equipment.

FP6000

Time Sharing

Since peripheral transfers are autonomous the central computer is free to proceed with other computations while the transfer is being carried out. Moreover, several peripheral transfers may be performed simultaneously under the control of separate control units.

It frequently happens that a program is unable to use all of the time between peripheral transfers to do useful work. A card-to-tape transcription program, for example, utilizes probably less than 10% of the computer's time and spends most of the time waiting for the card-reader — a slow peripheral. This situation was considered uneconomic even with early, slower computers and often led to costly special-purpose off-line equipment being built to convert data onto a faster medium for input or from a faster medium for output.

The concept of time-sharing has virtually eliminated the need for such equipment, and enables much more efficient use to be made of the complete system.

It is possible to have several programs in the system at once, many of them running at full speed. To an observer these programs all appear to be running simultaneously although in fact the computer obeys first one program and then another, switching from one to another perhaps as many as fifty times per second. This switching process is handled by a part of EXECUTIVE which ensures that the computer always obeys the highest priority program which is not held up. It is of the utmost importance that the whole process be entirely automatic; each programmer writes his own program without knowing where or when it is going to be interrupted or what programs will be running at the time — indeed without any reference to time-sharing at all. Each program uses its own peripheral devices which normally cannot be shared.

When a program requires to carry out a peripheral transfer, it obeys an instruction which in effect transfers control to EXECUTIVE which initiates the transfer and then returns control to the program just left. This program may proceed until it requires to use the information read in by the transfer (or requires the space from which data is being transferred). At this point the program will obey the instruction:

"Suspend me if my peripheral unit X of type N is busy".

Control is now returned to EXECUTIVE which scans the list of programs in priority order to find one which is free to proceed and is not held up for a peripheral device. This program then makes use of the time which otherwise would have been wasted. When the transfer relating to the first program is complete an automatic interrupt occurs, and, assuming that the first program had higher priority, control is transferred back to it now that it is able to proceed.

While the number of programs held in the system simultaneously could theoretically be quite large, a practical figure is probably three or four programs. When a request is made for a new program to be entered into the computer, EXECUTIVE will decide whether the required peripherals and sufficient core storage are available. The programs are each allotted their own core store areas and they remain in the core store when not being obeyed; the time-sharing facility does not necessitate dumping of the programs not being obeyed at any instant onto a backing store.

Since programs often contain mistakes even if they are carefully written and thoroughly tested, it can never safely be assumed that they are entirely free from blunders. In a computer system using time-sharing such errors would be particularly dangerous if the offending program spoiled the operation of the others, perhaps in such a way that the error was not immediately detected.

This situation would be quite intolerable.

To prevent such undesirable interference between one program and another, FP6000 is provided with a reservations system. This ensures that while any particular program is being obeyed the computer is allowed to use only those parts of the store and those peripheral devices which are reserved for that program. Should there be an attempted violation of a reservation the program is at once suspended and EXECUTIVE is entered. When the computer switches from one program to another under the control of EXECUTIVE, the reservation settings are changed to those appropriate to the new program; the special instruction needed to do this is not available to ordinary programs.

By means of the reservations it is made impossible for one program to interfere with the operation of another and it is quite safe for a new program to be tested by use of time-sharing while other programs are doing useful work.

FP6000

Executive

Probably the most important aspect of a computer system is the method by which it operates. A balance between hardware and software control of operation has been chosen for FP6000 so as to provide a fast, powerful but flexible set of operating procedures. EXECUTIVE may be thought of as an integral part of the hardware package, but in the light of experience gained in running any particular set of applications it remains possible to effect minor alterations to the way in which the machine works.

The main functions performed by EXECUTIVE are:

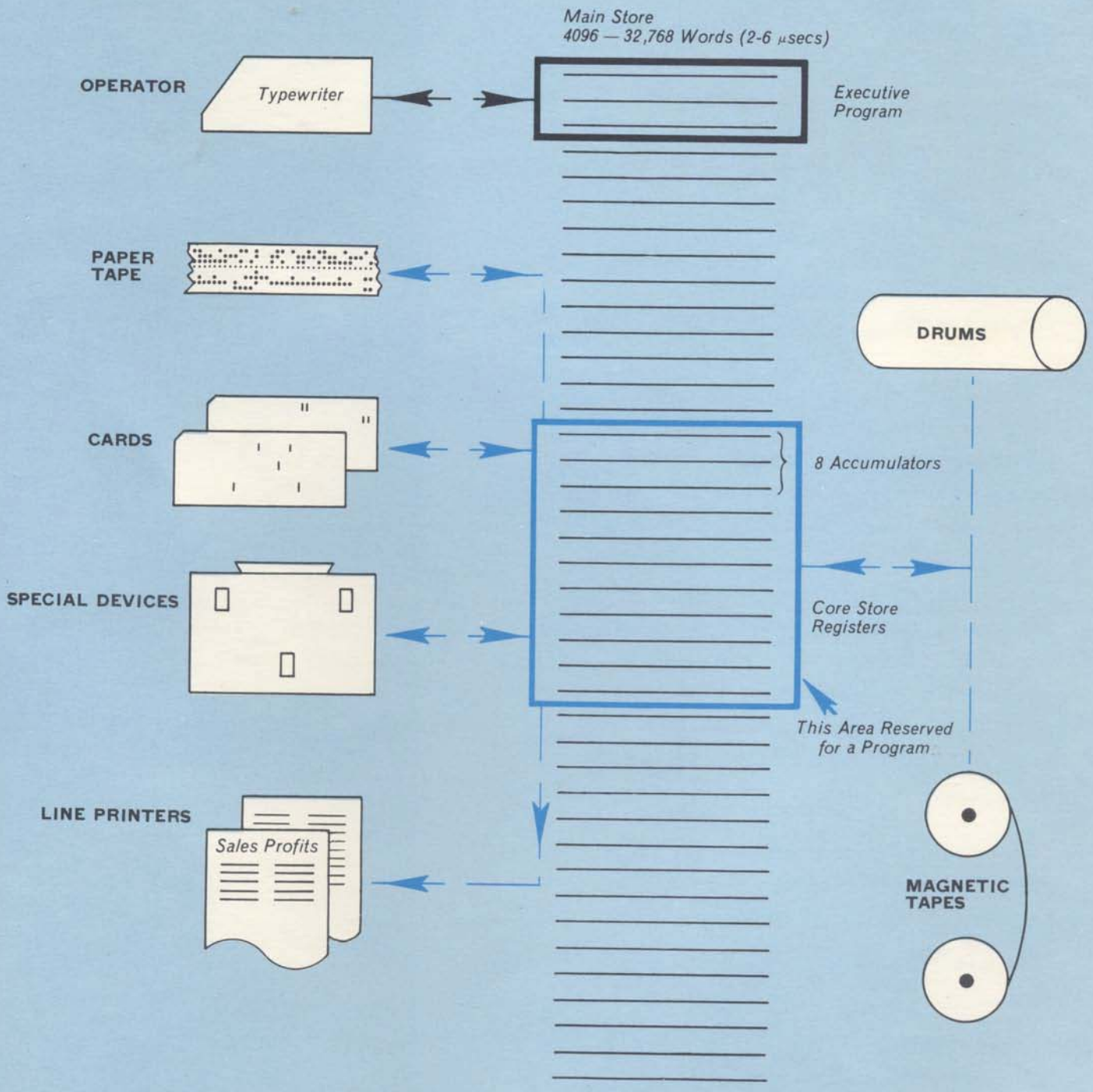
1) Control of peripheral transfers and provision of information about peripherals,
2) Communication between the operator and any individual program,
3) Allocation of the time of the central processor among programs, so that even the slowest peripherals do not cause wastage of time or require expensive buffering,
4) Monitoring,
5) Macro-instructions.

Any program may ask EXECUTIVE to perform a peripheral operation on its behalf and the complexities of checking the transfer are concealed from the programmer. He need not cater for unlikely eventualities in peripheral transfers, since in the event of a failure, diagnostic routines are entered automatically, information concerning the fault is displayed and the program in question suspended temporarily. Meanwhile of course, another program will have been entered and so machine down-time is reduced to a minimum.

EXECUTIVE will be of considerable help during program development since it provides facilities for monitoring programs, printing out the contents of selected locations, printing out all branching instructions which actually caused a transfer, accepting changes to individual instructions and generally providing the facilities necessary for rapid development. It is thus possible to develop a program at the console over a period of hours although the actual time spent obeying the instructions of the program may be counted only in seconds.

The ability to try out untested programs in parallel with operational routines without the risk of interference is one of the most important advantages of FP6000.

Structure of FP6000



FP6000

Systems adaptability

FP6000 has been designed to meet the needs of special systems while retaining all the advantages of general purpose computers. It frequently happens that a situation requires the construction of a unique piece of peripheral equipment and the interfaces in the FP6000 system are such that equipment may be attached with a minimum of disturbance. Equipment such as communication systems buffers are unlikely to be common to several situations and an important aspect of the FP6000 design is the flexibility with which such devices may be attached. It is no longer necessary to bend the requirements to suit the needs of the computer.

A few of the types of equipment which may be attached to FP6000 in addition to conventional peripherals are:

Analogue/Digital Converters and Multiplexers,
Communications Buffers,
Graph Plotters,
High and Low Speed Data Links,
Equipment to permit several FP6000s to be connected as a system.

Instruction Code

The following notes and diagrams explain the various meanings and the corresponding abbreviations that may be assigned to an FP6000 word.

N is a core store address or a number.

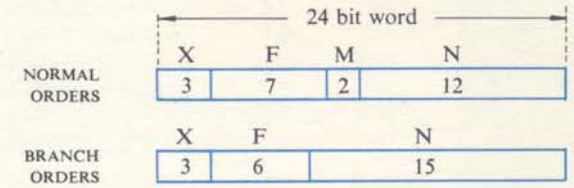
X is an accumulator (registers 0-7).

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

F is a function.

n,x,m refer to the contents of N,X,M respectively. x* refers to the contents of X + 1.

n+x ⟶ n means that the result of adding n and x replaces n, only the quantity following the arrow has been altered.

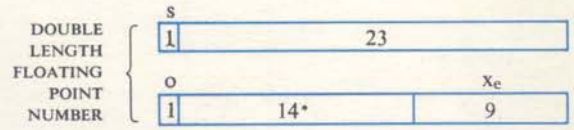


x: indicates a double length number, either fixed point, or floating point.

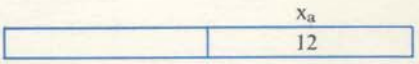
s is the sign bit or the most significant bit.

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

x_e, the signed floating point exponent, is the least significant 9 bits of the contents of X + 1.

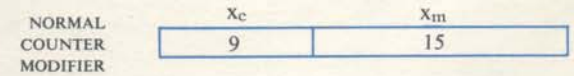


x_a is the least significant 12 bits of x (the N address).



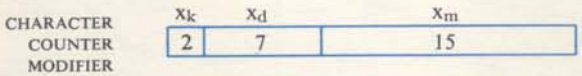
x_c is a 9 bit counter at the most significant end of x.

x_m is the least significant 15 bits of x.

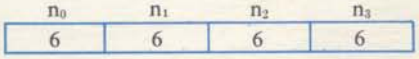


x_k is the most significant 2 bits of x, used in character modifying with end-around-carry to x_m.

x_d is the least significant 7 bits of x_c.

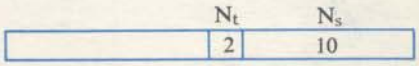


n_j is any one of n_0, n_1, n_2, n_3, the four 6-bit characters of n. Similarly, the four 6-bit characters of x are x_0, x_1, x_2, x_3.



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

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



In general all subscripts are applicable to x or n.

C is the carry digit register; c is used in orders 000-017 and 100-107. These orders perform single length arithmetic but the inclusion of c permits their operation on multiple length words; when carry or borrow occurs this is incorporated in the next part of the multiple length word. C is always left clear by any order unless that order sets C.

ONR is the order number register.

V is the arithmetic overflow indicator. V remains set until cleared.

In the following table, the times correspond to unmodified orders. For all modified orders an additional 2 microseconds should be added.

Instruction code (cont'd)

	Function Number	Description	Time (μsecs)	Notes
GROUP 00	000	$n + c \longrightarrow x$	7	
	001	$x + n + c \longrightarrow x$	7	Overflow may be set on exit but Carry is cleared
	002	$-n - c \longrightarrow x$	7	
	003	$x - n - c \longrightarrow x$	7	
	004	$n + c \longrightarrow x$	7	Overflow cannot be set Sign of result always positive Carry set if appropriate
	005	$x + n + c \longrightarrow x$	7	
	006	$-n - c \longrightarrow x$	7	
	007	$x - n - c \longrightarrow x$	7	
GROUP 01	010	$x + c \longrightarrow n$	7	
	011	$n + x + c \longrightarrow n$	7	As 000-003 but with n and x interchanged
	012	$-x - c \longrightarrow n$	7	
	013	$n - x - c \longrightarrow n$	7	
	014	$x + c \longrightarrow n$	7	
	015	$n + x + c \longrightarrow n$	7	As 004-007 but with n and x interchanged
	016	$-x - c \longrightarrow n$	7	
	017	$n - x - c \longrightarrow n$	7	
GROUP 02	020	$x \,\&\, n \longrightarrow x$	7	Logical AND
	021	$x \vee n \longrightarrow x$	7	Logical INCLUSIVE OR
	022	$x \not\equiv n \longrightarrow x$	7	Logical EXCLUSIVE OR
	023	Obey n as an instruction	3	
	024	$n_j \longrightarrow x$	7	Extract character
	025	$n_e \longrightarrow x$	7	
	026	Set C if $n \neq x$ or $c = 1$	7	
	027	Set C if $n + c > x$	7	Similar to 007 but result not stored
GROUP 03	030	$n \,\&\, x \longrightarrow n$	7	
	031	$n \vee x \longrightarrow n$	7	As 020-022, result in N
	032	$n \not\equiv x \longrightarrow n$	7	
	033	$0 \longrightarrow n$	5	Clear n
	034	$x_3 \longrightarrow n_j$	7	
	035	$x_e \longrightarrow n_e$	7	Part word replacements
	036	$x_a \longrightarrow n_a$	7	
	037	$x_m \longrightarrow n_m$	7	
GROUP 04	040	$n.x \longrightarrow x:$	40	Unrounded multiply
	041	$n.x + 2^{22} \longrightarrow x:$	40	Rounded multiply
	042	$n.x + x^* \longrightarrow x:$	41	Semi-cumulative multiply
	043	$10.x: + n_j \longrightarrow x:$	27	Decimal-binary conversion
	044	$x:/n \longrightarrow x^*$	42-45	Unrounded double length division (remainder to X)
	045	$x:/n \longrightarrow x^*$	43-48	Rounded double length division (remainder to X)
	046	$x^*/n \longrightarrow x^*$	41-44	Unrounded single length division (remainder to X)
	047	$10.x: \longrightarrow x:$ Char $\longrightarrow n_j$	27	Binary-decimal conversion
GROUP 05	050	Branch if $x = 0$	5	
	052	Branch if $x \neq 0$	5	
	054	Branch if $x \geqslant 0$	5	
	056	Branch if $x < 0$	5	
GROUP 06	060	Single word modify	5	Add 1 to modifier; Add 2 to modifier; Add ¼ to modifier — Subtract 1 from counter, branch if counter non-zero
	062	Alternate word modify	5	
	064	Character modify	5	
GROUP 07	070	Subroutine entry	8	Store ONR and V in X clear V, branch to N
	072	Subroutine exit	4	Branch to N + x, V reset as at entry unless set by subroutine
	074	Branch on Condition X; X = 0 unconditionally X = 1 if V set X = 2 if V set, clear V X = 3 if V clear X = 4 if V clear, clear V X = 5 if C set X = 6 if C clear	3	

Instruction code (cont'd)

	Function Number	Description	Time (μsecs)	Notes
GROUP 10	100	$N + c \rightarrow x$	5	As 000-003 but N used in place of n
	101	$x + N + c \rightarrow x$	5	
	102	$-N-c \rightarrow x$	5	
	103	$x - N - c \rightarrow x$	5	
	104	$N + c \rightarrow x$	5	As 004-007 but N used in place of n
	105	$x + N + c \rightarrow x$	5	
	106	$-N-c \rightarrow x$	5	
	107	$x - N - c \rightarrow x$	5	
GROUP 11	110	Left Shift x, N_s places	$6 + N$	$N_t = 0$ Cyclic Shift $N_t = 1$ Logic Shift $N_t = 2$ Arithmetic Shift $N_t = 3$ Special Shift
	111	Left Shift x:, N_s places	$15 + N$	
	112	Right Shift x, N_s places	$6 + N$	
	113	Right Shift x:, N_s places	$15 + N$	
	114	Normalize x	$15 + N$	
	115	Normalize x:	$16 + N$	
GROUP 12	120	$x \,\&\, N \rightarrow x$	5	As 020-022 but N used in place of n
	121	$x \vee N \rightarrow x$	5	
	122	$x \not\equiv N \rightarrow x$	5	
	123	No Operation	3	Dummy instruction
	124	$N \rightarrow x_c,\ 0 \rightarrow x_m$	5	Set counter
	125	Set Mode N	4	
	126	Block Transfer	$13 + 4N$	N words from address x to address x*
	127	Check Sum $\rightarrow$ x	$14 + 3N$	Sum N words from address x* ignoring overflow
GROUP 13	130	Convert Fixed to Floating		Floating Point Operations
	131	Convert Floating to Fixed		
	132	$x: + n: \rightarrow x:$		
	133	$x: - n: \rightarrow x:$		
	134	$x: \cdot n: \rightarrow x:$		
	135	$x: / n: \rightarrow x:$		
	136	$\sqrt{n}: \rightarrow x:$		
GROUP 14	140 to 147	Spare		
GROUP 15	150	Suspend if peripheral unit X of type N is active		
	151	Suspend if subprogram N is active		
	152	Activate subprogram X entering at instruction N		
	153	Release peripheral unit X of type N		
	154	Abolish this program in mode X		
	155	Suspend this program in mode X		
	156	Suspend this program pending operator intervention on unit X of type N		
	157	Suspend this program pending operator action and type out message		
GROUP 16	160	Magnetic tape transfer with program suspension		
	161	Drum transfer with program suspension		
	162	Paper tape read with program suspension		
	163	Paper tape punch with program suspension		
	164	Read from typewriter		
	165	Print on typewriter		
	166	Print one line with program suspension		
GROUP 17	170	Magnetic tape transfer without program suspension		
	171	Drum transfer without program suspension		
	172	Paper tape read without program suspension		
	173	Paper tape punch without program suspension		
	174	Card read		
	175	Card punch		
	176	Print one line without program suspension		

FERRANTI-PACKARD ELECTRIC LIMITED

INDUSTRY STREET,
TORONTO 15, CANADA.

UNITED KINGDOM

Ferranti Limited,
London Computer Centre,
68 Newman Street, London W 1,

U. S. A.

Ferranti Electric Inc.,
Industrial Park No. 1,
Plainview, L.I., New York.

AUSTRALIA

Ferranti Ltd.,
34 Queens Road,
Melbourne SC2, Victoria.

(650) IM-9/62

PRINTED IN CANADA