Ferranti Pegasus, Perseus and Sirius

Instruction sets and instruction times

Ferranti Pegasus

Instruction Set

Arrangement of a typical 19-bit order

7 bits	3 bits	6 bits	3 bits
N	***X***	***F***	***M***
Address or constant	*accumulator*	*op-code*	*modifier (index register)*

Two 19-bit orders are fitted in to one 39-bit Pegasus word. The most significant bit is called the Stop/Go digit. When a program is assembled, the system normally sets the digit to 1. If during execution of the program the digit is found to be zero, then the computer stops.

Address map for the Pegasus fast store.

decimal addr.	***program notation***	***description***
0 -> 7	0 -> 7	accumulators, X0 – X7. (X0 is always zero).
8 -> 14	-	(unassigned: always contain zero)
15	15	handswitches (20 bits)
16	16	input/output (5 bits, checked)
17	17	input/output (5 bits, unchecked)
18 -> 31	-	(unassigned: always contain zero)
32	32	constant (-1.0)
33	33	constant (1/2)
34	34	constant (2 ↑-10)
35	35	constant (2 ↑-13)
36 -> 63	-	(unassigned: always contain zero)
64 -> 111	0.0 -> 5.7	48 words for program/data, as six block of 8 words each.
112 -> 127	-	(unassigned: always contain zero).

Computing store addresses 64 to 111 in theTable are identified in the form *Block.Position*, running from 0.0 through to 5.7. The eight accumulators, denoted as X0 – X7, can be used either for computation or for address-modification and loop-counting. X0 is always zero. Accumulators X6 and X7 act as a double-length pair *p,q* during certain multiply, divide, and shift instructions.

Notation used in Instruction Set

N	First address in order
X	Accumulator
x, x'	Word in X, before and after execution
n, n'	Word in location of address N, before and after execution
c	Counter
(pq)	Contents of registers 6 and 7
OV'R	Overflow indicator

Pegasus instruction set.

The following tables refer to Pegasus 1. See Pegasus Programing Manual for Pegasus 2 additions.

00	x' = n *(n is contents of an addr)*	40	x' = c *(c is an 8-bit signed number (literal)).*
01	x' = x + n	41	x' = x + c
02	x' = – n	42	x' = – c
03	x' = x – n	43	x' = x – c
04	x' = n – x	44	x' = c – x
05	x' = x & n	45	x' = x & c
06	x' = x *XOR* n	46	x' = x *XOR* c
07		47	
10	n' = x	50	x' = 2^{N}x arithmetic shift
11	n' = n + x	51	x' = 2^{-N}x rounded shift
12	n' = – x	52	Shift x up N places (logical shift)
13	n' = n – x	53	Shift x down N places (logical shift)
14	n' = x – n	54	(pq)' = 2^{N}(pq)
15	n' = n & x	55	(pq)' = 2^{-N} (pq) unrounded
16	n' = n *XOR* x	56	(pq)' = 2^{μ} (pq); x' = x – $2^{-38}\mu$; normalise
17		57	
20	Multiply: (pq)' = n.x	60	Jump if x = 0
21	Multiply and round-off in X6: p' = $(n.x)_r$	61	Jump if x ≠ 0
22	Multiply and add: (pq)' = n.x + (pq)	62	Jump if x ≥ 0
23	Justify (nq)	63	Jump if x < 0
24	Divide, unrounded: q' = (xq)/n. p' = remainder	64	Jump if overflow clear
25	Divide, rounded: q' = $[(xq)/n]_r$ p' = remainder	65	Jump if overflow set, and clear
26	Divide, rounded, q' = $[x/n]_r$	66	Unit-modify: increment modifier & jump if $x_p \neq 0$.
27		67	Unit-count: decrement counter & jump if ≠ 0.
30	Unallocated	70	Single-word read from main (drum) store to X1.
31		71	Single-word write to main (drum) store from X1.
32		72	Block read from main (drum) store
33		73	Block write to main (drum) store
34		74	Select input/output device (external switching).
35		75	
36		76	
37		77	Stop

Instruction Timing

Timings are measured in word-times or *beats*, the time for a word to pass a given point in the (serial) computer, A beat is 42 x 3 = 126 microseconds.

Normal execution of an order pair takes 5 beats as follows:

Order-pair to Order Register	1 beat
a-order executed	2 beats
b-order executed	2 beats

Extra time is needed by some orders as in the table:

Order	Extra beats
Groups 0,1,4	None
20, 21	13
22	14
23	None
24 - 26	41
50, 51, 53 - 55	N number of places shifted
52	If N<25 then N; if N>24 then N-25
56	m+2 where m is number of places shifted
70 - 73	See programming manual
74	None
76	8

Peripheral timing

The paper tape readers operate at 200 characters per second. The computer is held-up and waits if an attempt is made to read from a tape reader within 5mS of the previous read operation.

The paper tape punch operates at 30 characters per second. The computer is held-up and waits if an attempt is made to write to the tape punch within 33mS of a previous write operation.

The teleprinter operates autonomously at 7 characters per second.

The Ferranti Perseus Data Processing System

Instruction set

In 1959, this was known as the “Order Code” of Perseus. A Perseus word is 72 bits long and is capable of holding three orders

Arrangement of a typical order

1	6 bits	5 bits	3 bits	6 bits	3 bits
Stop/go digit	*B*	*P*	*S*	*F*	*M*
	N		R	L	

Notation

N	First address in order	
X	Mixed radix accumulator	
Y	Binary accumulator	
B	Block in the the computing store	*B.P-R* defines a particular order.
P	position of word in block	
R	Position of order in word	
L	Link digit	
S	Selector	
H	Shunting quarter block	
V	Block of buffer store	
B.T	Quarter block of main store	
x, x'	Word in X, before and after execution	
y, y'	Word in Y, before and after execution	
n, n'	Word in location of address N, before and after execution	
m, m'	Modifier, before and after execution	
c, c'	Counter, before and after execution	
d	Contents of punched card	
b.t	Contents of quarter block of computing store	
v, v'	Contents of buffer store, before and after execution	
a	Contents of multiplier register (18)	
(pq)	Contents of registers 16 and 17 (24 decimal digit capacity)	
OV’R	Overflow indicator	

The Order Code

In the tables below, the digits in the left column are the function digits, read as two octal digits, and occupying the F bits of an order. Each table represents one "group" of instructions

00	$n' = x$
01	$x' = n$ (complementing)
02	$x' = x + n$
03	$x' = x - n$
04	$x' = x$ (sign and modulus)
05	$x' = B.P$
06	$x' = x + B.P$
07	-

40	$m' = B.P\text{-}R$	*for setting modifier*
41	$m' = m + B.P\text{-}R$	*for changing modifier*
42	$c' = N$	*for setting counter*
43	$c' = c + N$	*for counting*
44	$s' = B.P$	*for setting selectors*
45	$(pq)' = n \times a$	*multiplication (24 decimal digits accuracy)*
46	$(pq)' = (pq) + n \times a$	*cumulative multiplication*
47	Form $(pq)/n$	*division (from 24 decimal digits dividend, if required)*

10	$n' = y$
11	$y' = n$
12	$y' = y + n$
13	$y' = y - n$
14	$y' = y \,\&\, n$
15	$y' = N$
16	$y' = y + N$
17	$n' = 0$

If N is negative the shift will be in the reverse directio n.

50	Arithmetical shift of x, N characters up
51	Arithmetical shift of x, N characters down
52	Logical shift of y, N characters up
53	Logical shift of y, N characters down
54	Logical shift of y, N bits up
55	Logical shift of y, N bits down
56	Double length arithmetical shift, N characters up
57	Double length arithmetical shift, N characters down

20	Jump to *B.P-R* (storing the link if $L = 1$)	*if* $x = 0$
21		*if* $x \neq 0$
22		*if* $x \geq 0$
23		*if* $x < 0$
24		*if* $y = 0$
25		*if* $y \neq 0$
26		*if* $y \geq 0$
27		*if* $y = 0$

60	$b' = v$	*transfers between computing store and buffer store for magnetic tapes*
61	$v' = b$	
62	$b' = v;\ v' = b$	
63	$(b.t)' = d$	*transfer contents of punched card from buffer store to computing store*
64	$(b.t)' = h$	*shunting of quarter blocks between computing store and buffer store*
65	$h' = (b.t)$	
66	$(b.t)' = h;\ h' = (b.t)$	
67	Jump to *B.P-R*, obey that order and return	

30	Jump to *B.P-R* (storing the link if $L = 1$)	*if* *OV'R*1 clear	and clear *OV'R*1
31		*if* *OV'R*1 set	
32		*if* *OV'R*2 clear	and clear *OV'R*2
33		*if* *OV'R*2 set	
34		*if* $m = 0$	
35		*if* $m \neq 0$	
36		*if* $c = 0$	
37		*if* $c' \neq 0$; $c' = c - 1$	

70	Read next block of tape
71	Write on next block of tape
72	Step back one block and read from tape
73	Step back one block and write on tape
74	Search for block c on tape
75	Rewind tape
76	Step back one block on tape
77	Stop; on restarting jump tp *B.P-R*

The operation of all the orders is described in detail with examples of their use in the "Ferranti Perseus Computer Programming Manual". A few highlights are mentioned here.

The Stop/Go bit is normally set to value 1 when the order is created. If the machine attempts to execute an order with a zero Stop/Go bit, then the machine halts immediately and a warning lamp is lit on the control panel. The purpose is to guard against a program erroneously jumping into data. The bit is also used to force entry to the Initial Orders for monitoring puposes.

The B.P bits normally represent an address of a word in the computing store, i.e. an operand address. If a particular order in a word is to be addressed, as in Jump instructions, then the two R bits specify which order of the three held in a word. In Jump instructions the L bit if set forces a link to be placed in the Link Register, which is jump back to the instruction following the Jump instruction which placed the link. Thus returning from a sub-routine is simply effected by jumping to the Link Register.

In many arithmetic orders, the S bits specify one of the eight qualifiers (including zero). The qualifier specified contains the Selector, the definition of the field to be selected in the current operand.

Most orders can be modified (index register) by using the M bits to specify one of the qualifiers. The qualifier specified contains a modifier which is added to the current operand address before the operand is fetched from the computing store. Similarly, counters can be kept in qualifiers.

Both multiplication and division is provided in hardware in the Mixed Radix Accumulator, the X-register. A "Halving and Doubling" algorithm is used which simplifies the process in the mixed radix environment. The processes are autonomous, that is, once the operation has started, the computer can get on with other orders provided those do not refer to the Mixed Radix Accumulator. If a reference is made to that Accumulator during the processing time, then the referring instruction is held-up and waits until the long operation is complete

Instruction Timing.

The timing of programs is quite difficult because of the way words are arranged in the computing store, and the fact that three orders are held in one word. Blocks 0 to 4 of the computing store are made up from single-word nickel delay lines, which take one word time (78 x 3 = 234 microseconds) to circulate. Blocks 5 to 31 are made up from 16-word nickel delay lines with a circulation time of 1872 microseconds. The addresses of words in the long lines are scrambled, so that for program instructions arranged sequentially the next word of three orders is ready to read out when the last of the previous triplet has been executed.

One word time is required to read a triplet of orders out of the computing store and into the internal Order Register (OR) of the machine. Once in the OR, the three orders are executed sequentially (assuming there is no jump away from the sequence) with no additional reference to the computing store for orders. If the triplet has to be read from Blocks 5 to 31 of the computing store, and the orderly sequence has been disrupted by a jump or a long order, then the machine may have to wait several word times before the triplet can be read.

The time to access an operand from Blocks 0 to 4 is one word time, and from Blocks 5 to 31 there may be a 15-word waiting time. It is therefore conventional to store data in blocks 0 to 4 and program in blocks 5 to 31.

Assuming that the operand is in blocks 0 to 4, the time for execution of the orders is as follows:

Orders	Word times
00 - 44	1
45 - 46	1 and then autonomous working. The time to complete a multiplication is approximately 3.3 word-times per significant digit in register 18
47	1 and then autonomous working. The time to complete a division is approximately 4(a"+1) + 3n" word-times where a" is the number of significant digits in the double-length dividend and n" is the number of significant digits in the divisor
50 - 51	(N+1) for a shift of N places
52 - 55	N for N places, but 1 for zero places
56 - 57	(N+2) for N places, but 1 for zero places
60 - 62	32
63 - 66	8 for blocks 0 to 4, 16 for blocks 5 to 31
67	1
70 - 71	1 then autonomous. Time to transfer a block to or from tape is about 75 mS
72 - 73	1 then autonomous. Time to step back and repeat is about 150 mS
74	1 then autonomous. Time to seek is about 75mS per block passed
75	1 then autonomous. Time to rewind a 3000 ft reel about 4 minutes
76	1 then autonomous. Time to step back one block is about 75 mS

Both tape readers operate at a maximum speed of 200 characters per second. It therefore takes 5mS to advance the tape after a read operation. If another read operation is attempted within that time then it is held-up and waits until the tape has advanced to the new character.

The control desk teleprinter operates at a maximum of 7 characters per second. It takes one word time to transfer a character to the teleprinter. If an attempt is made to send another character within 143 mS, then the computer is held-up waiting for the previous character to be printed.

The card readers operate at 300 cards per minute. Attempting to transfer from the card buffer within 200 mS since the last transfer will cause the computer to hold-up and wait for the buffer to be refilled.

The Samastronic Printer

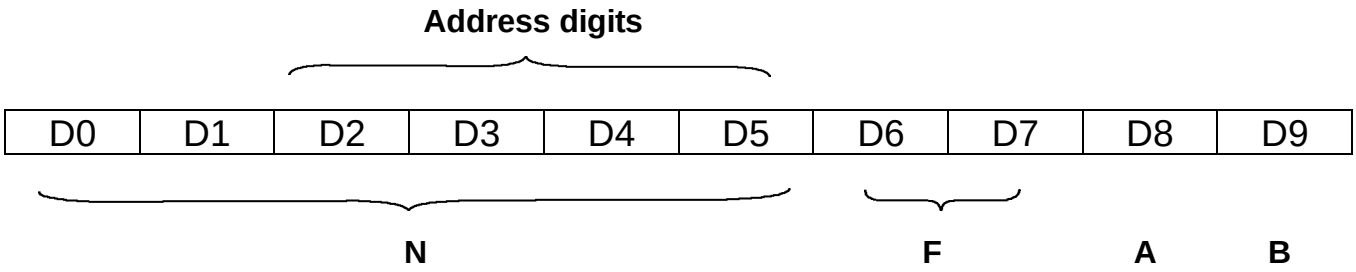
The printer is off-line from the computer so does not affect computing speed. The printer has a repertoire of 50 symbols, printing in 140 columns at 10 columns per inch. Vertical pitch is 6 or 8 lines per inch. Normal printing speed is 300 lines per minute. Maximum paper width is 18 inches, but there are two paper carriages so that two independent webs of paper can be printed on. Apart from the flexibility of laying-out the print positions on the magnetci tape blocks, a comprehensive set of plugboards was provided on the printer to route character positions to any print positions, and to provide stock phrases triggered by single magnetic tape characters. A number of other layout facilities are provided by the plugboards.

The Ferranti Sirius Computer

Instruction Set

A Sirius word is 40 bits long comprising ten BCD digits. When the decimal digits are interpreted as an instruction they are arranged as shown:

Arrangement of a typical order



Notation

- **N** An address in Store (3 or 4 digits) or a literal number (6 digits)
- **F** Function digits arranged as ten groups of ten
- **A** Specifies one of the nine accumulators for arithmetic operations
- **B** Specifies one of the nine accumulators for address modification
- ***n*** Contents of **N**
- ***a, b*** Contents of **A**, **B**
- ***n', a'*** Contents of **N, A** after the operation
- **OVR** Overflow indicator
- $\mathbf{X_m}$ Accumulator m

00	$a' = a + N$	20	$a' = 10a + N$
01	$a' = a - N$	21	$a' = 10a - N$
02	$a' = -a - N$	22	$a' = -10a - N$
03	$a' = -a + N$	23	$a' = -10a + N$
04	$a' = N$	24	$a' = 10a +$ M.S.D of N
05	$a' = a + 10^4N$	25	$a' = 10a + 10^4N$
06	$a' = a - 10^4N$	26	$a' = 10a - 10^4N$
07	$a' = -a - 10^4N$	27	$a' = -10a - 10^4N$
08	$a' = -a + 10^4N$	28	$a' = -10a + 10^4N$
09	$a' = 10^4N$	29	$a' = 10a +$ M.S.D of 10^4N
10	$a' = a + n$	30	$a' = 10a + n$
11	$a' = a - n$	31	$a' = 10a - n$
12	$a' = -a - n$	32	$a' = -10a - n$
13	$a' = -a + n$	33	$a' = -10a + n$
14	$a' = n$	34	$a' = 10a +$ M.S.D of n
15	$a' = a + b$	35	$a' = 10a + b$
16	$a' = a - b$	36	$a' = 10a - b$

17	$a' = -a - b$	37	$a' = -10a - b$
18	$a' = -a + b$	38	$a' = -10a + b$
19	$a' = b$	39	a' = 10a + MSD of b in LSD position
	In all the above, if B = 0 then the number on the keyboard is obtained.		
40	$a' = (a + 5) / 10$	Arithmetical shift down (rounded)	
44	$a' = a / 10$	Arithmetical shift down (unrounded)	
45	$a' = (a + 5) /10$ + L.S.D. of N	(rounded)	
49	$a' = a / 10$ + L.S.D. of N	(unrounded)	
50	Dummy	55	Jump to N unconditionally
51	Jump to N if M.S.D of $a \neq 0$	56	Jump to N if M.S.D of $a = 0$
52	Jump to N if $a \neq 0$	57	Jump to N if $a = 0$
53	Jump to N if OVR set	58	Jump to N if OVR clear
54	Jump to N if $a < 0$	59	Jump to N if $a \geq 0$
	Orders 53 and 58 clear the OVR		
60	$n' = a$	70	x_9' = quotient, a' = remainder on dividing (a, x_9) by b. Unsigned
64	$n' = 0$		
66	$a' = a$ & N	71	a' = TAPE
68	$a' = a$ & 10^4N	72	(TAPE)' = a
69	$a' = x_1$ and jump to N	73	(TAPE)' = a and a' = TAPE
		74	Half-signed multiply
99	Wait	79	(a, x_9)' = b x x_9

Accumulator 1 is the control register and contains the address of the next instruction.
Accumulator 0 = 0 when used as B, = keyboard or display when used as A (with some exceptions).
Order 69 is a jump and store link instruction to enter a subroutine.
The operation of all orders is described in detail in "The Ferranti Sirius Computer Programming Manual" together with examples of their use.

Instruction Timing

Instructions 00 - 09, 15 - 29, 35 - 50 and 65 -68 always take three word times once the instruction is about to emerge from the store, and so take 240 microseconds.

Instructions which access the store for an operand (10 - 14, 30 - 34, 60 and 64) may have to wait up to an additional 4mS for the operand to emerge from the store. The programmer may choose to use "optimum programming" principles to reduce that delay.

With Jump orders (51 -59 and 69) 3 word-times are used if the order does not jump. If the jump is successful, then up to 4mS may have to be added to allow the jumped-to instruction to emerge from the store.

Orders which operate the input/output devices will in general be timed by the speed of the devices.
The time for multiplication or division is typically 4 or 8 mS, but can take up to 16mS in worst cases.