Ferranti Pegasus, Perseus and Sirius Computers

Software

All three machines were designed before 1960, when high-level languages had not yet emerged, and programs were prepared in machine code or simple assembler languages and typically debugged by the programmer hands-on to the machine. Although the machines were designed by different teams in different locations, and were very different in applications, they had one common characteristic - the instruction sets were powerful and elegant, and were easy and intuitive to remember. The instruction sets were different for each machine and so machine code programs were not portable from one machine to another. However, Pegasus Autocode was available for both Pegasus and Sirius permitting a program to be written for both machines.

The regularity and "no awkward exceptions" nature of the instruction sets allowed simple assembler programs to be used. These were not true symbolic assemblers, as reference to store addresses was always numeric, albeit the store references could often be relative to some implied or explicit local or global variable. Functions were specified by their number, not by mnemonics. Numbers in programs were expressed in either decimal or in octal, depending on the context. For example, functions in Pegasus and in Perseus were conventionally written in octal whereas in Sirius they were written in decimal.

Pegasus

The built-in Initial Orders (IOs) in Pegasus were stored on a reserved and write-protected part of the magnetic drum. The IOs were entered by operation of the Start switch on the control desk. They then examined the control desk hand-switches to see whether any directive had been set there by the operator, otherwise they read paper tape from the first tape reader, looking for acceptable directives. One such directive was that the IOs should read the remainder of the tape as a program to be assembled and loaded into the machine for subsequent execution. The IOs provided many facilities for monitoring and debugging programs, for accounting for machine usage, for amending programs and for producing binary versions of programs ("Object Code" in modern parlance) for subsequent fast input and production use. The use of the IOs and how to write programs for Pegasus is described in meticulous detail and precision in the famous Pegasus Programming Manual written by George Felton. At the time this became something of a benchmark in the industry for a good programming manual.

Supplementing the IOs internal to Pegasus was a comprehensive collection of Library Programs, which could either be used freestanding, or else assembled into a user's program. The libraries were held in binary form on reels of paper tape, typically mounted on the second tape reader. If a user program called for a library routine, then this was copied by the IOs as the user program was being assembled. Library programs covered both scientific and commercial calculations, ranging from powerful facilities for formatting data for printing, to equation solvers and PAYE calculations. One widely used library was the Matrix Interpretive Scheme, used heavily in the aircraft industry.

A number of standard application programs were available for such applications as engineering frame-stressing calculations, pipe-stressing, operational research, critical path analysis, survey analysis and data analysis, production control, stock control and payroll. Moreover each Pegasus installation would build up its own software suites on a very diverse range of tasks, and such software was often shared.

Pegasus Autocode was a primitive high-level language which was interpreted by the Autocode library. Use of the language allowed rapid programming of applications by persons unfamiliar with the details of the design of Pegasus. An example of an Autocode program to calculate the root mean square (RMS) value of the variables *v*1 to *v*100 follows, with an annotation (which is not punched on the program tape) after each instruction:

```
    n1 = 1                  set integer 1 to value 1
    v101 = 0                set (floating point) variable 101 to value zero
2)  v102 = vn1 x vn1        set v102 to the product of variables indexed by n1
    v101 = v101 + v102      accumulate total
    n1 = n1 + 1             increment index
    →2, 100 ≥ n1            jump to label 2 if n1 is less than 100
    v101 = v101 / 100       divide total by number of entries
    v101 = SQRTv101         call built-in square root function
```

Perseus

The built-in Initial Orders (IOs) in Perseus were stored on a reserved and write-protected part of the magnetic tape on mechanism 0. The IOs were entered by operation of the Start switch on the control desk. They then examined the control desk hand-switches to see whether any directive had been set there by the operator, otherwise they read paper tape from the first tape reader, looking for acceptable directives. One such directive was that the IOs should read the remainder of the tape as a program to be assembled and loaded into the machine for subsequent execution. The IOs provided many facilities for monitoring and debugging programs, for accounting for machine usage, for amending programs and for producing binary versions of programs ("Object Code" in modern parlance) for subsequent fast input and production use. The use of the IOs and how to write programs for Perseus is described in detail in the Perseus Programming Manual written by Peter Hunt.

A set of library routines was available for Perseus, created by Ferranti programmers and the two customers' programmers. Since both machines were used in Live Insurance offices, the routines included standard magnetic tape operations, check-pointing and restarts, sorting, card translation, formatting for printing on the Samastronic printer and so on.

Sirius

The Initial Orders (IOs) in Sirius were fed iton and stored in the first 200 words of the store, where they could be optionally write-protected. The IOs were entered by operation Clear Control button on the control panel. They then read paper tape from the first tape reader, looking for acceptable directives. One such

directive was that the IOs should read the remainder of the tape as a program to be assembled and loaded into the machine for subsequent execution. The IOs also provided facilities for monitoring and debugging programs. The use of the IOs and how to write programs for Sirius is described in detail in the Sirius Programming Manual.

Sirius was also supplied with a comprehensive collection of Library Programs similar to those supplied with Pegasus. Library programs covered both scientific and commercial calculations, ranging from powerful facilities for formatting data for printing, to equation solvers and PAYE calculations. The Matrix Interpretive Scheme was also provided. Pegasus Autocode was also available.

Existing Machines

Two Pegasus machines are known to exist. Machine serial number 25 has been restored to working order by the Compuer Conservation Society and is in the Computing Gallery of the Science Museum in London. Demonstrations are given of the machine working approximately every two weeks.

Pegasus number 6 is in the Museum of Science and Industry in Manchester. It is not complete and is not in working order, but has been conserved to prevent deterioration.

The only two Perseus systems were scrapped, and no relics are known to have survived.

Several Sirius systems still exist. One is believed to be in storage of the London Science Museum, and was probably owned by the Admiralty. No attempt has yet been made to restore thiat machine.

There are two Sirius machines in Melbourne, Australia. It is believed that there is an intention to restore one of them by local enthuiasts.

Simulators

A realsitic simulator of Pegasus is available for downloading from the Computer Conservation Society ftp archive. The simulator runs on a personal computer and presents a view on the screen similar to that when sitting at a real Pegasus. The simulator has all the library programs and many demonstration programs, together with a tutorial and facilities for "punching" and editing virtual paper tape.

No simulator is known for Perseus.

A simulator for Sirius exists and it is hoped will be released soon.