Ferranti Atlas 1 & 2 – Software & Sample Programs

Version 1: 11 November 2003

Contents

1. Available Software

Atlas was provided with an assembler language and compilers for a number of high level 'scientific' languages, both international standards and special developments. The assembler was called Atlas Basic Language - ABL – and is detailed in section 2 below. The other languages were:

International standards and pre-existing languages

- FORTRAN
- Algol 60
- Extended Mercury Autocode (EMA)

Specific Atlas languages

- Atlas Autocode
- CPL (Combined Programming Language)
- BCPL (Basic CPL)

Atlas Autocode was developed at Manchester, and CPL at Cambridge and London Universities. BCPL was a reduced version of CPL used to write the CPL compiler. It allowed more flexible data manipulation and was a direct precursor of the C and C++ languages.

Ferranti, and, after being taken over, ICT, provided a number of packages for computer service users. These included:

- A General Survey Program
- Linear Programming

The DTI CAD Centre at Cambridge provided a large number of application packages for the users of its Atlas and the STAR network.

2. Sample ABL

ABL remains close to Atlas machine code – see section **X3** and the instruction summary. The basic instruction layout is thus:

Field:	Function	Index register 1	Index register 2	Address
Atlas Notation:	F	Ba	Bm	S

Instructions are written with commas after each field; thus:

121, 1, 0, 10,

sets B1 to 10. In the address field, * can be used to mean the address of the current instruction. It may be modified by an integer, e.g. –1* means the previous instruction. An instruction may be preceded by a label of the form of a number followed by a right parenthesis, e.g. 3). This may be referred to in the address field of another instruction as (3) and may be modified in the same way as *, e.g. 1(3) means the instruction after that labelled 3. Address fields (or literals) may be written as decimal numbers or as left-justified octal numbers preceded by J. Writing to fixed store addresses, which have a 1 in the most significant position (i.e. J4) has no effect so, for example :

346, 0, 0, J4

will merely clear the accumulator, A..

All basic floating point arithmetic can be carried out by the following instructions:- 320-322, 324-325, 356, 362-363, 367, 374.

Library routines, for example for input and output conversion, may be referred to by a special form of address field containing 'L'. Thus, for example, the extracode

1101, 90, 0, A2/L100

will read a line from the current input stream and convert it to a floating point number which is left in the accumulator.

The end of a program document is signalled by a line with ***Z. Alternative endings are allowed to signal special situations, e.g. ***B to indicate binary data follows.

3. Job Descriptions

See section **X2**-4. All programs and associated data must be accompanied by a job description. This contains the information the Supervisor requires to assemble and schedule the program. The job description is an Atlas document, just like the program and data. It relates logical numbers used in the program for peripheral streams to input documents, output equipment and magnetic tapes. It also provides estimates of the computing time needed, total execution time, including time spent waiting for tape transfers, store used and the amount of output produced. If these estimates are exceeded, the Supervisor may terminate the program.

A job description takes the following form:

JOB
<the name of the job>
COMPILER <the language the program is written in>

These are optionally followed by a list of input documents with their assigned program numbers, preceded by the word INPUT, and output devices used, also with numbers and size estimates, preceded by OUTPUT. Input 0 is the document containing the program to be compiled; for simple programs this may follow directly after the job description.

4. A Simple Program

This section illustrates machine-level programming by a simple program written in "Intermediate Input", an early version of ABL, for the Manchester Atlas in 1963. Figure 1 is a printout of the input to the system, which was on a paper tape document. The program is intended to find integers a, b, c, all less than k, such that $a^2+b^2=c^2$.

```
JOB
j.k.b.-3/SUMS OF SQUARES


COMPILER  INTERMEDIATE INPUT
(0) = 0
4)      +50/+0  |k
        121,    1,      0,      0,
1)      124,    1,      0,      1,
        150,    1,      0,      (4),
        226,    127,    0,      2(0),
        1117,   0,      0,      0,

        121,    2,      1,      -1,

2)      124,    2,      0,      1,
        150,    2,      0,      (4),
        227,    127,    0,      (1),

        121,    3,      2,      0,

3)      124,    3,      0,      1,
        150,    3,      0,      (4),
        227,    127,    0,      (2),

        121,    4,      1,      0,
        121,    5,      2,      0,
        121,    6,      3,      0,

        1302,   4,      1,      0,
        1302,   5,      2,      0,
        1302,   6,      3,      0,

        121,    7,      4,      0,
        124,    7,      5,      0,
        122,    7,      6,      0,

        214,    127,    7,      (5),
        217,    127,    7,      (2),
        121,    127,    0,      (3),

5)      1064,   0,      0,      2,
        1067,   1,      0,      0,
        1067,   2,      0,      0,
        1067,   3,      0,      0,

        121,    127,    0,      (2),

        E1(4)
```

Set a value of 50 for k into the top half of the word
b1'=0

b1'=b1+1
bt'=k-b1
if bt>=0 jump over next, else stop

b2'=b1-1

b2'=b2+1
bt'=k-b2
if bt>=0 go to 1)

b3'=b2

b3'=b3+1
bt'=k-b3
if bt>=0 go to 2)

b4'=b1
b5'=b2
b6'=b3

b4'=b4xb1
b5'=b5xb2
b6'=b6xb3

b7'=b4
b7'=b7+b5
b7'=b7-b6

if b7=0 go to 5)
if b7<0 go to 2)
go to 3

output new line
output b1
output b2
output b3

go to 2)

Figure 1: A Simple Program

The instructions have been annotated. The first few lines form a simple job description, which assumes a number of defaults for output, etc. The code begins on the line labelled 4) where 50 is stored in the top half of the word (and 0 in the bottom half). As the only comment (!) shows, this is the value for k. The bottom line, which indicates the end of the code, also shows that the entry for execution to begin is 1(4), i.e. the next instruction after the word labelled 4). The program is by no means typical since it uses only B registers and the B arithmetic unit, but it does illustrate some aspects of Atlas programming. The following points are worth noting:

- The sequence of instructions 150 followed by 226 or 227 set the test register bt and then jump according to its value, by changing the program counter B127
- (0) refers to the current instruction in the same way as *
- 1302 is an extracode to provide multiplication of B registers (reasonably uncommon in a normal program, using the floating-point accumulator for arithmetic operations)
- 1117 signals to the Supervisor that the program is finished
- 1064 and 1067 are members of a set of extracodes that provide standard-format outputs.

Output was to paper tape, which was then printed to give the results shown in Figure 2.

```
00
TIME 16.19.04.
j.k.b.-3/SUMS OF SQUARES

 3 4 5
 5 12 13
 6 8 10
 7 24 25
 8 15 17
 9 12 15
 9 40 41
 10 24 26
 12 16 20
 12 35 37
 14 48 50
 15 20 25
 15 36 39
 16 30 34
 18 24 30
 20 21 29
 21 28 35
 24 32 40
 27 36 45
 30 40 50
TIME 16.19.09. 0
```

Figure 2: Results

5. References

The following references in section **X5** are most relevant to this section:

6, 7, 9, 13, 25, 26, 27, 31, 33, 44, 46, 48, 49, 50, 52, 53, 54, 55.