8 ORION LIBRARY

8.0 About this section.

8.1 GENERAL ORGANIZATION OF LIBRARY

The library for Orion will be available on magnetic tape, and will be a mixture of Basic Input language and binary information. The name of the library document is ORION/SYSTEM/LIBRARY/+. The magnetic tape library is written with regard to the conventions of section 6.1 of this manual. Each individual document of the library can be a simple document on paper tape or cards.

8.1.1 Library subroutines

The items of a subroutine which need to be distinguished by a compiler are:-

(i) The name: this is of the form of any (simple) document name;

(ii) A set of instructions and constants, which will be in the language Basic Input;

(iii) A set of input and output parameters. An output parameter is some function of the subroutine, e.g. its length, or the address of its entry points, which can only be determined by the subroutine. An input parameter is some information required by the subroutine, e.g. the address of an auxiliary routine. The subroutine will normally contain optional values for ell input parameters;

(iv) Calls for other subroutines;

(v) 'Public interludes', i.e. interludes which calculate the value of an input or output parameter;

(vi) 'Secret interludes' e.g. an interlude to select one of a number of tables, according to some input parameter. Any interlude not public is secret.

8.1.2 Relocatable Accumulators; Accumulator length

A subroutine uses accumulators as sources of operands, destinations of results, and working space. If a subroutine uses p accumulators, then p is the accumulator length.

Some of the accumulators may be in special positions relative to others, e.g. l.s. halves of double-length accumulators. These are not independently relocatable. The other accumulators, which are relocatable, will be referred to in the subroutine by identifiers. Optional settings of these identifiers will be contained in the subroutine, so that, optionally the subroutine uses A1 onwards. The symbolic programmer may relocate the accumulators by setting his own values to the identifiers.

A separate identifier gives the address of the first of the block of accumulators (optional setting A1); by resetting this, the symbolic programmer can move the block of accumulators used by the subroutine.

For example, a subroutine which uses 6 accumulators, 1 for a small parameter, 1 as working space, and 4 as 2 double-length accumulators has an accumulator length of 6 and 4 relocatable accumulators. If it identifier these as L3l, L32, L33 and L34 , the subroutine will contain optional equations.

G30 = A1　　　L30 is the ‘separate’ identifier referred to above.

G31 = L3O　　　If the subroutine needs to refer to the l.s

G32 = L30 + 1

G33 = L30 + 2　　　halves of the double-length accumulators,

G34 = L30 + 4　　　it will call them L33+1 and L34+1 respectively.

8.1.3 An identifier, in Basic Input, is of the form Ln.p, $0 \le n \le 127$, $0 \le p \le 63$. The following identifiers will be standard for all subroutines:

(i) Entry points

L0.1, L0.2, ..., L0.29, equivalent to symbolic identifiers E1, E2, ..., E29 (at most 29, densely numbered, starting at L0.1)

(ii) Relocatable accumulators

L0.31, L0.32, ..., L0.60, equivalent to symbolic identifiers X1, X2, ..., X30 (At most 30, densely numbered, starting at L0.3l).

L0.30 is the accumulator block address (as L30 above). It is equivalent to symbolic identifier X0.

(iii) Link store

L0.62, equivalent to symbolic identifier L.

(iv) L0.63 is the length of the absolute stored form of the subroutine

(v) Preset parameters.

L1.1, L1.2, ..., L1.63, equivalent to symbolic identifiers P1, P2, ..., P63. (At most 63, densely numbered, starting at L1.1).

All identifiers can represent at most 24 bits.

8.1.4 Calls for a library subroutine by a library subroutine will be of the form:

(i) The name of the subroutine to be called.

(ii) A set of equivalences of the form ‘set L2.1 of the calling subroutine to have the value of L0.1 of the called subroutine.

(e.g. :ARCSIN/ calls :SQRT/; since :ARCSIN/ contains the instruction
86 Entry point of :SQRT/ Link store of :SQRT in the form

```
86    L2.1    L2.2,
```

the equivalence above is required).

8.1.5 Interludes are written in the language of Basic Input. ‘Public’ interludes must be run at an early stage when the compiled program is being read by Basic Input, since their results may be required to set the transfer address (forward references in the setting of transfer addresses are prohibited in Basic Input). ‘Secret’ interludes can be run at any time by Basic Input; this will normally be while the subroutine is being read from the library tape. Such interludes are stored with the subroutine as part of the same document.

8.2 Layout of the Library Tape

The magnetic tape library may be on a separate tape or it may be on the Systems tape. In the former case, the tape name is ORION/SYSTEM/LIBRARY/+, and the document list is ORION/SYSTEM/LIBRARY/DOCULIST/DATE, the document list containing the full name of every subroutine and other library documents on the tape. When the library is on the systems tape, the tape name is ORION/SYSTEM/+, and the document list is ORION/SYSTEM/DOCULIST/DATE.

8,2.1 Standard specifications

The first document of the library proper is a document ORION/SYSTEM/LIHRARY/STDSPECS. The function of this is to provide compilers with some of the information presented to programmers in the printed specifications of the subroutine. The standard specifications are written in binary.

8.2.1.1 In general there are 2 words for each standard specification; a word of packed information followed by one word containing the name of the subroutine within the library, e.g. ORION/SYSTEM/LIBRARY/SQRT/- is the subroutine :SQRT.

An abbreviated form is necessary; since compilers require random access to this information, it must be stored in the core or drum stores during compilation.

8.2.1.2 The layout of the word of information is:

no. of bits	1	5	6	6	6	6	12	6
information	S	X	P	A	E	H	L	N

m.s.end l.s.end

where:

S = 0 if the subroutine does not call for other library subroutines, and does not have a public interlude;

X, P and E are respectively the number of relocatable accumulators, preset parameters, and entry points;

A is the accumulator length;

L is the length (in words) of the absolute stored form of the subroutine; 0 if variable

H (not applicable if S=0) is the height of the subroutine (c.f. 8.2.2.1)

N is the number of words in the name which follows; normally N=1; N=7 denotes last standard specification, and this specification is a dummy.

8.2.2 Supplementary specifications

A document called ORION/SYSTEM/LIHRARY/SUPSPECS follows the document ORION/SYSTEM/LIBRARY/STDSPECS on the tape. Only those subroutines which call for other subroutines, or which have a public interlude (or both) have a supplementary specification.

8.2.2.1 The supplementary specifications may well be too long to store for random access; they are, therefore, arranged in sets. The sets occur in order, but are not ordered internally.

The last set consists of supplementary specifications of those subroutines which call for no subroutines. Other sets consist of supplementary specifications of subroutines which call for subroutines with supplementary specifications in following sets, or with no supplementary specifications.

H=1 for those subroutines with supplementary specifications in the last set; 2 for those in the penultimate set etc. (c.f. 8.2.1.2).

This arrangement makes serial access possible.

8.2.2.2 A supplementary specification is of the form:

A marker word, followed by

One word containing the name, followed by

0 or more calls for other subroutines, followed by

0 or 1 public interludes.

(i) The marker word is of the form:

no. of bits	1	1	4	18	6	6	12
information	S	I	C	Spare	H	N	K

m.s.end l.s.end

where

S=1 if this is the last supplementary specification (a dummy)

I=1 if there is a public interlude

C is the number of calls in this supplementary specification

H and N are as in the standard specification (N=1 here)

K is the number of words in this supplementary specification.

H, here, is intended primarily to facilitate insertion of a new subroutine on the library tape

(ii) The name is the same as that in the standard specification.

(iii) A call is of the form:

A marker word, followed by.

the name of the subroutine to be called, followed by some equivalences.

(a) The marker word is of the form:

24 bits	24 bits
Length of name (in words)	Number of equivalences

m.s. end
l.s. end

D0 of this marker word is 1 or 0 according as the subroutine being called is or in, not to be loaded into an auxiliary chapter.

(b) The name is the same as that in the standard specification.

(c) The equivalences are of logical form:
"Set Ln of the calling subroutines = Lp of the called subroutine" and of physical form:

12 bits	12 bits	12 bits	12 bits
n1	p1	n2	p2

i.e. 2 equivalences per word, n = p = 0 will be ignored.

Lp must refer to a relocatable accumulator, preset parameter, entry point, the link or the length of the called subroutine.

(iv) An interlude is of the form:

A marker word, followed by

The instructions of the interlude in the language of Basic Input.

The marker word is an integer, the number of words required to store the interlude in its Basic Input form.

8.2.3 Following the supplementary specifications on the tape are the subroutines, in Basic Input language, each one a separate document. Secret interlude are included in these documents. The subroutines may be in any order, but will be placed in the library tape according to priority i.e. those subroutines used most will come at the beginning.