

INTERNAL USE ONLY

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

Scientific Programming Department
I.C.T. 1900 Series.

FORTRAN NOTE 7
21.5.65.

The FORTRAN IV Object Time Peripheral Routines

This note describes the structure of the peripheral routines that may be used in FORTRAN IV object programs. It assumes a knowledge of FORTRAN NOTES 2, 4 and 5.

FORTRAN IV Object Time Peripheral Routines

Input/Output in FORTRAN IV object programs is handled by one basic segment called %FINOUT and one or more peripheral routines. There is a separate peripheral routine for each type of input/output medium. %FINOUT does most of the work; the peripheral routines carry out only the physical transfers and any required conversion to or from 1900 internal code. Communication between %FINOUT and the peripheral routines is described in FORTRAN NOTE 5.

The compiler will decide which peripheral routines are needed by looking at the peripheral statements in the program description. These may be typified by:

```
INPUT 1 = p/q
OUTPUT 5 = p/q (.....)
```

where: p is a 1900 peripheral name, e.g. MT3
q is a qualifying phrase.

In a particular case one of p or /q may be absent.

Usually, only p will be present. The compiler will include details of the device in an information block and will cause a standard peripheral routine to be included in the object program. The following routines will be available as soon as possible:

<u>p</u>	<u>Peripheral Routine</u>	<u>For</u>
TRn) TPn)	%FIOPT	8-track paper tape in 1900 code.
CRn) CPn)	%FIOCARD	punched cards in 1900 code.
LPn	%FIOLP	line printer.
MTn	%FIOMT	magnetic tape.

n is an integer.

If q is present it acts as a qualifier and indicates that a special purpose peripheral routine is required. The actual routines that may be provided will depend on the demand. Possibilities are:

<u>p/q</u>	<u>Peripheral Routine</u>	<u>For</u>
TRn/FIVE	%FIOPT5	5-track tape in ICT code
MTn/EVEN	%FIOEVEN	magnetic tape recorded in even parity.
CRn/IBM	%FIOIBMC	punched cards in IBM code.

If p is omitted the specified peripheral routine is included but no information about an actual peripheral is recorded. It follows that the routine must sort out for itself what peripherals it may use. One very special qualifier is 'NONE', e.g.

```
OUTPUT 3 = /NONE
```

This causes the peripheral routine %FIONONE to be included in the program. This routine is a dummy and any output (or input) associated with the programmer's number 3 is suppressed. This feature may be useful during program testing.

If it proves feasible the programmer will be allowed to define his own values of q and associate them with his own peripheral routines. The relevant statement could be:

```
DEFINE (q, r)
```

where q is a qualifier

r is the name of a peripheral routine; this may be a string of up to 11 digits and letters starting with a letter.

Then, wherever q appears in a peripheral statement in the Program Description the compiler will arrange to include the routine r. For example, the statements

```
DEFINE (MINE, MYROUTINE)
INPUT 3 = MT1/MINE
```

would cause the routine MYROUTINE to be used in association with MT1.

[Question: Can anyone think of a better word than 'DEFINE'?]

Entry Points

All entries to peripheral routines made by %FINOUT are relative to the first word of the routine. It follows that entry points are completely standard and some routines must have entry points that will not sensibly be required; e.g. a paper tape routine must have a Backspace entry. Entry points are as follows ('name' is the name of the peripheral routine):

name	+0	Initiate Read
	+1	Read one formatted block
	+2	Read one unformatted block
	+3	Terminate Read
	+4	Initiate Write
	+5	Write one formatted block
	+6	Write one unformatted block
	+7	Backspace
	+8	Rewind
	+9	Endfile
	+10	All other operations.

Initiate Read sets up any initial conditions required for input. %FINOUT obtains an input record one block at a time by using one of the entries Read one formatted or Read one unformatted block. The peripheral routine signals to %FINOUT when the last block of a record has been read by setting a particular bit in %FIOBUF. The Terminate Read entry reads up to the end of the current record if this has not already been reached.

Initiate Write sets up any initial conditions required for output. %FINOUT passes a record over to a peripheral routine one block at a time by using one of the entries Write one formatted or Write one unformatted block. The peripheral routine terminates a record when a particular bit in %FIOBUF is found to be set.

How a record is divided into blocks is dictated by the buffer size. In practice, a formatted record will usually consist of only one block; in particular, 1 punched card = 1 block = 1 record.

Backspace and Rewind are faults in most peripheral routines. End-file will output an end-of-file record whose form will depend on the output medium.

The effect of entry 10 is decided by an integer parameter in X3 as follows:

- 0 Release device
- 1 Regain device
- 2 Disengage device
- 3 Runout device

Other operations may be implemented later.

Special actions

For some peripherals some special action is required on the first transfer (e.g. a magnetic tape file must be opened, an extra new line is output on a paper tape punch, etc.) and a bit is available in a peripheral's information block to indicate whether or not a device has previously been referred to.

Availability of Peripherals

Slow peripherals are reserved by the compiler in the program's request slip. Others must be reserved by the peripheral routine when it is entered for the first time.

Checks on whether or not a peripheral is available occur in %FINOUT, so all entries to peripheral routines (except of course 'regain') may assume that a device is available, though not necessarily opened.