

INTERNAL USE ONLY

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

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

FORTRAN NOTE 5  
30.4.65.

The FORTRAN IV Object Time Input/Output System

This note gives a brief description of the input/output system used by FORTRAN IV object programs. The emphasis is on how communication between the different parts of the system is achieved. Further information on the system will appear in later notes. (Some information has already appeared in Note 4.)

Amendment to FORTRAN Note 4, page 2.

Entries 0, 24, 26 and 28 to %FINOUT are replaced by entry 0 meaning: all other operations. The actual operation is specified by an integer parameter in X3.

## The FORTRAN IV Object Time Input/Output System

### 1. Outline of System

The FORTRAN IV input/output system is modular. At run time, only those parts that are needed will be present. The relationship between the different parts is shown diagrammatically in Figure 1.

For each source language input/output statement the object program will contain one or more calls to %FINOUT. %FINOUT is a segment that carries out the bulk of the work of I/O in a way that is independent of the I/O medium involved. %FINOUT enters one of a series of peripheral routines to carry out actual I/O operations. Each peripheral routine is a segment.

%FIOTA is a segment that is called for each occurrence of a total array in a READ or WRITE statement.

### 2. Communication between %FINOUT and the Peripheral Routines

%FINOUT does not know about the different I/O media but merely enters appropriate peripheral routines to perform the actual I/O operations. Communication takes place via four common areas: %FIOLIST, %FIOBUF, %FIOPER and %FIOLEN.

The Program Description for a FORTRAN IV program generates in the common area %FIOLIST a list of programmer's numbers that may be used by that program. Associated with these numbers are the details of the actual devices and all auxiliary information required for the successful operation of these devices, including the addresses of the appropriate peripheral routines.

The term "Programmer's number" replaces "peripheral value" of Note 4.  
for

An initial entry to %FINOUT will search %FIOLIST/auxiliary information corresponding to a specified programmer's number. Then the one word area %FIOPER is set to point to this information for use by the peripheral routine.

The layout of the common area %FIOBUF is shown in Figure 2. It is used to transmit information between %FINOUT and the peripheral routines during READ and WRITE operations. This information is transmitted in units of blocks, where one record may consist of one or more blocks. The block size depends on the peripheral routine. A formatted record is likely to consist of one block and an unformatted record of several blocks. Formatted information will be in standard 6-bit code (without shifts). Unformatted information will be in binary form. %FIOBUF is defined as one word in %FINOUT but its actual length is the maximum of the lengths defined in the peripheral routines.

During a READ or WRITE operation the initial entry to the peripheral routine will set the one word common area %FIOLEN equal to the maximum number of characters that %FINOUT may assume to exist in %FIOBUF.

### 3. Structure of %FIOLIST

%FIOLIST contains a chained list of information about the peripherals used by a program. It contains an entry for each of the INPUT, OUTPUT, USE or CREATE statements in the Program Description (see Note 2). USE and CREATE replace INPUT/OUTPUT and OUTPUT/INPUT of Note 2. An entry consists of one or more pointers and an information block.

For each programmer's number in the statement there is a three word pointer:

1. List word
2. Programmer's number
3. Address of information block.

The 'list word' contains the address of the previous pointer in %FIOLIST. If there is no previous pointer the word is zero. The first word of %FIOLIST points to the end of the list, i.e. it contains the address of the last pointer put into the list.

For each occurrence of the word MONITOR in a statement there is a similar pointer, except that the second word is n, where  $n < 0$ . The value of n indicates whether the device is available for input, output or both.

An information block contains the following items:

1. Address of the start of the Peripheral Routine (1 word).
  2. Device details (2 words).
  3. Zero word.
  4. Length in words of rest of information block (1 word).
  5. Length in words of file name (1 word).
  6. File name (0 or more words).
  7. Any further information.
- } Optional

N.B. Before entering a peripheral routine, %FINOUT sets %FIOPER equal to the address of the information block.

Item 1 of the information block gives the starting address of the appropriate peripheral routine and is used by %FINOUT.

Item 2 consists of two words as follows:

	<u>Value of word</u>		<u>Meaning</u>
<u>Word 1</u>	Top bit = 1		Device not released
	Rest of word = 1	} Indicates type of statement in Program Description	{ INPUT OUTPUT USE (ex INPUT/OUTPUT) CREATE (ex OUTPUT/INPUT)
	= 2		
	= 3		
	= 7		
<u>Word 2</u>	Top bit = 1		Device not yet used
	Rest of word = n		n is the number in the 1900 device name, e.g. 3 in MT3.

N.B. The top bits of both these words may be referred to and changed only within peripheral routines since they may not have precisely the same meaning for all peripheral routines; some peripheral routines may ignore them altogether.

Item 3 of the information block is a word that is available as working space for the peripheral routine.

Item 4 may be zero.

Item 5 may be zero.

Item 6 will contain the file name left justified. Spaces will be added on the right to make a whole number of words. The compiler imposes no limit on the length of a name.

Item 7 may be a retention cycle and/or any other information that proves desirable.

K. F. James.

V. K. Taylor.

Diagram of Input/Output System

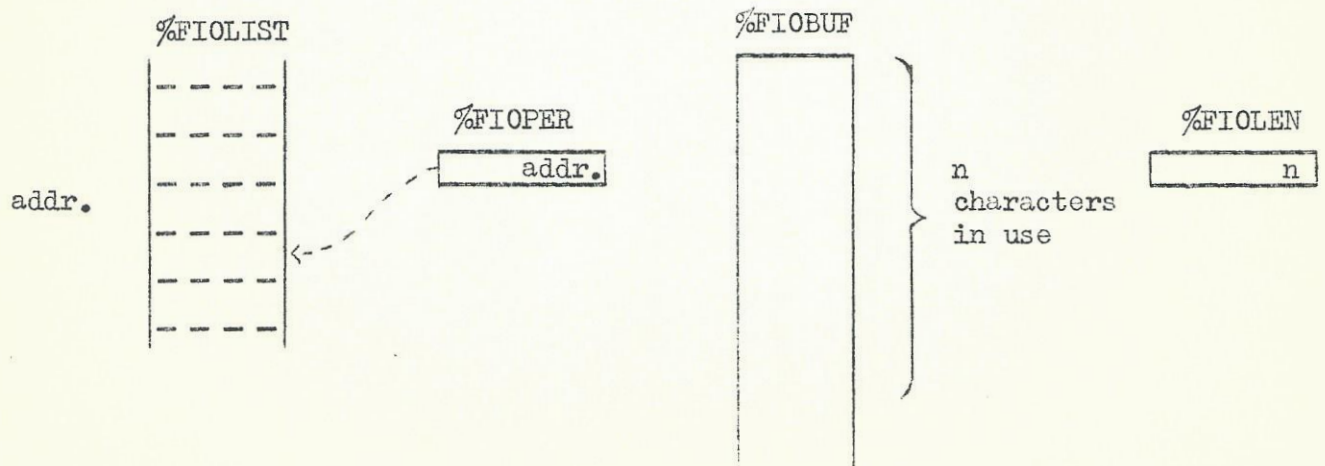
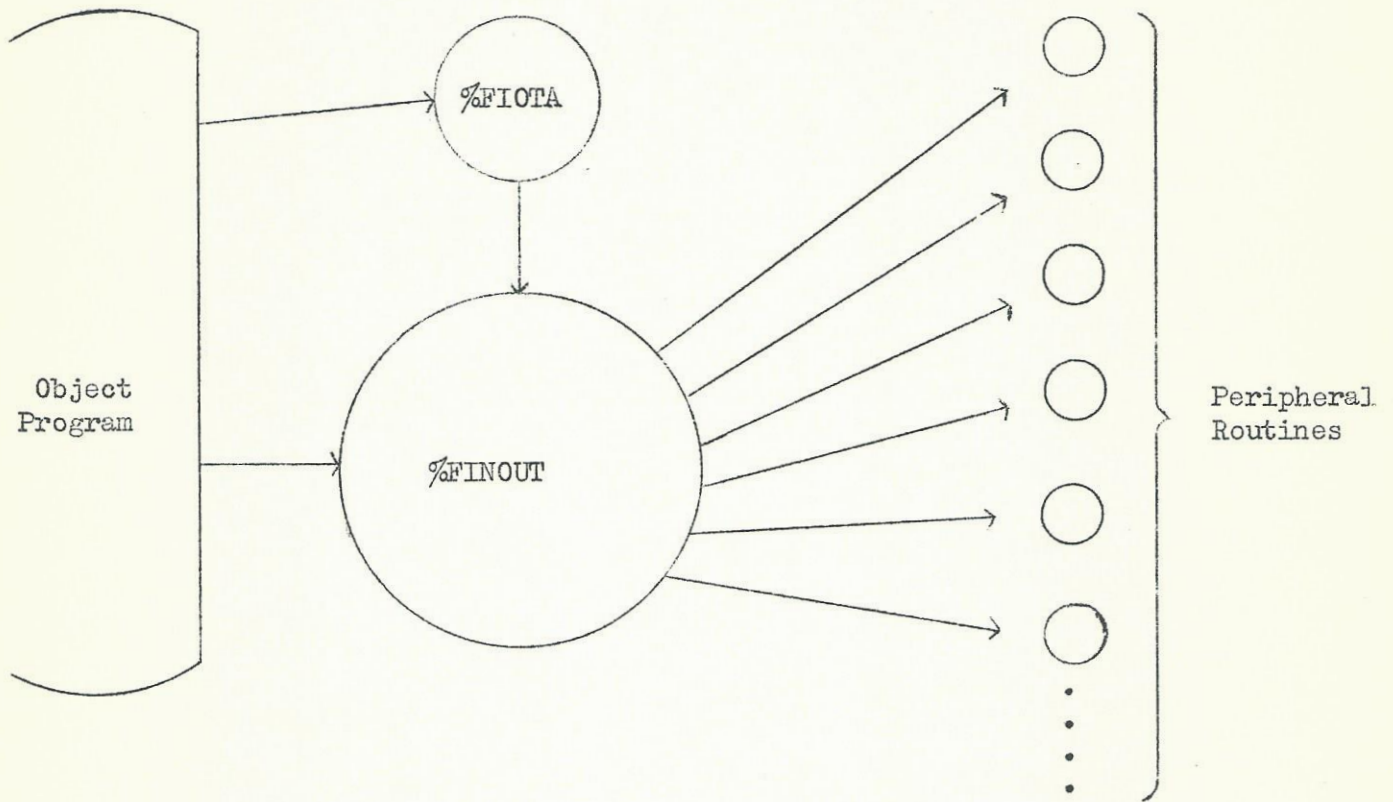


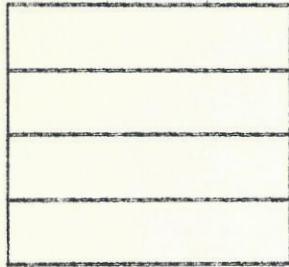
FIGURE 1

%FIOBUF

Layout

Formatted

Unformatted



Words 1-4

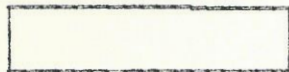
Used by Peripheral Routines



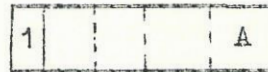
Word 5

Top bit = { 1 - last block of a record  
0 - other than last block of a record. }

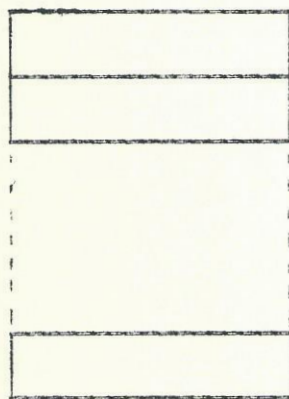
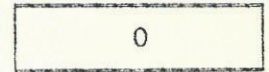
Rest of word - no. of characters of information \*



Word 6



Top bit = 1  
A = 1st char. of information.



Rest of Information

Information

\* No. of characters of information starts at 3rd character of word 6 for formatted information and at 1st character of word 7 otherwise.

FIGURE 2