INTERNAL USE ONLY

INTERNATIONAL COMPUTERS AND TABULATORS LIMITED

Programming Languages Division

FORTRAN NOTE 21
28.1.66

# The FORTRAN IV Object Time Input/Output System

This note replaces FORTRAN NOTE 5. It gives a brief description of the input/output system used by FORTRAN IV object programs. The emphasis is on the methods of communication between the different parts of the system. Further information appears in Fortran notes 4, 7 and 18.

# 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 diagramatically 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.

For double precision variables a call is generated to %FIODP instead of %FINOUT, but %FIODP uses %FINOUT for all processing except numeric conversion.

Similarly %FIODA is called instead of %FIOTA for double precision arrays.

## 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 three common areas: %FIOLIST, % 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 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.

An initial entry to %FINOUT will search %FIOLIST for 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 buffer is given in section 4. 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. 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. The length of the buffer 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 2 word common area %FIOLEN :

Word 1 (formatted) No. of characters of information in Buffer

(unformatted) " " words " "

Word 2 Start address of information

On reading, BO of word 1 will be set to 1 by the peripheral routines when the last block of a record is read.

On writing, this bit is set to 1 by %FINOUT whenever a record is to be terminated.

3. Structure of %FIOLIST

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

For each programmer's number in the statement there is a 2 word pointer:

1. Programmer's number
2. Address of information block

The first word of %FIOLIST contains the number of entries in the list.

For each occurrence of the word (MONITOR) in a statement there is a similar pointer, except that the 1st word is n, n < 0. The value of n indicates whether the word is available for input or output being -1 for input and -2 for output. These pointers are for use by the object time diagnostic routines e.g. TRACE (%FMN4).

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 block
5. Any further information (Optional.)

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

Item 1 is used by % FINOUT.

Item 2 consists of 2 words as follows:

| | Value of Word | Meaning |
|---|---|---|
| Word 1 | Top bit = 1 | Device Allocated |
| | Bits 1-8 | Type of device (as in PERI control area) |
| | Rest of word = 1 | INPUT |
| | = 2 | OUTPUT |
| | = 3 | USE |
| | = 7 | CREATE |
| Word 2 | Top bit = 1 | Device not yet used. |
| | l.s. 12 bits = n | n is number in 1900 device name, e.g. 3 in MT3. |

The top bits of these words are refered to and changed only within peripheral routines; some peripheral routines ignore them altogether.
Item 3 is used as working space for the peripheral routines, e.g. %FIOMT stores statement type in bits 0-5 and data block count in bits 6-23.
Item 4 is zero if there is no further information
Item 5 (if item 4 non-zero) contains file name plus any other information which may prove desirable. Only up to 12 characters of the file name are significant to executive, names less than 12 characters being space-filled on the right.

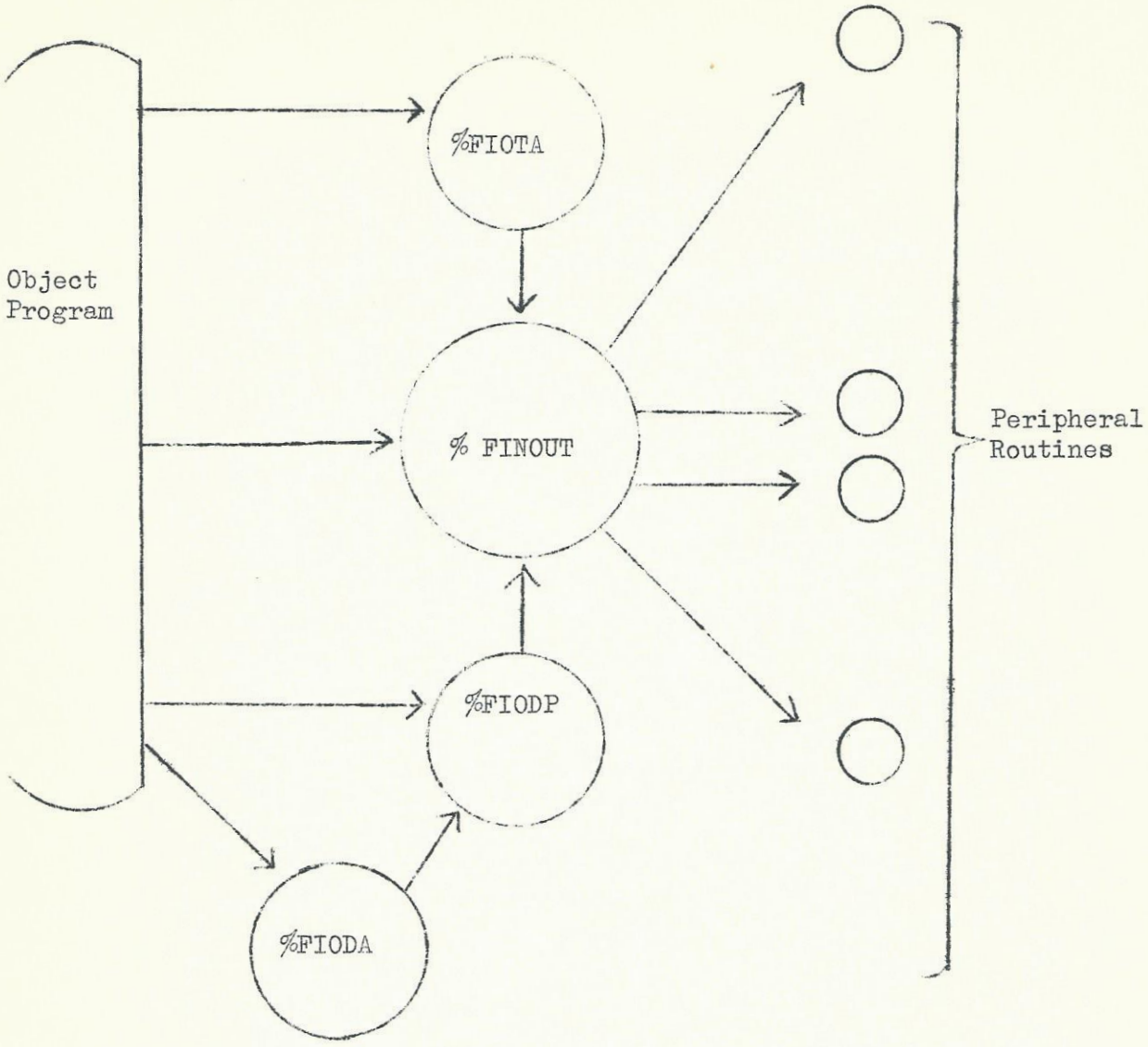
4. Structure of the buffer

For character peripherals, except the line printer, user information starts at the beginning of the first word; for the line printer, at character 3 of the first word.
On magnetic tape the first few words of the buffer are used as control words and user information starts at character 2 of word 2 for formatted tape (for compatability with #XQMP) and at word 5 for unformatted tape.

Although each peripheral routine may in theory use a separate buffer in practice all existing routines use the common area %FIOBUF.

Alison Finch

Diagram of Input/Output System



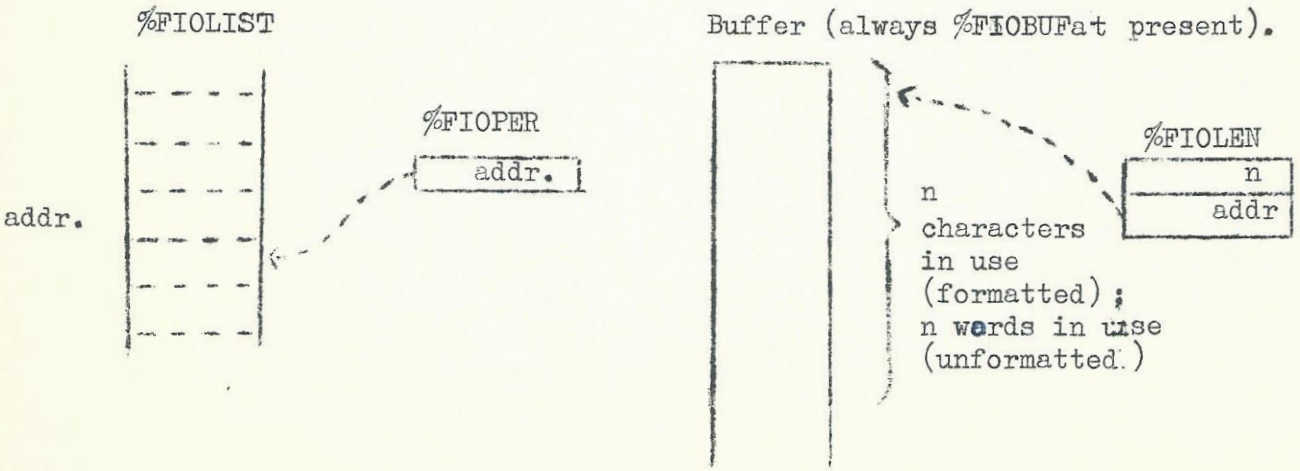




FIGURE 1