836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL.(617) 851-4111, TWX 710 343-6769, TELEX 94-7421

The Option 2 General I/O ROM (Read Only Memory) for the Wang System 2200B adds five BASIC language statements to the system (see Table 1).

Table 1. Option 2 Statements

Statement	Description
$GIO	A generalized I/O statement designed to perform data input, data output, and I/O control operations with a programmable signal sequence.
$IF ON	A statement designed to test the Ready/Busy signal of an I/O device (or test the Empty/Full signal of the input buffer on a device controller board) and initiate execution of a conditional branch to a specified line number.
$TRAN	A statement designed to facilitate high-speed character code translations.
$PACK $UNPACK	Statements designed to facilitate data packing and unpacking by fields or delimiters, between a specified alphanumeric array buffer and specified variables in an argument list.

The $GIO statement is unlike any other System 2200 statement and merits extra attention. By a technique similar to machine language programming, I/O operations can be custom-tailored in a "general input/output" format executable within the framework of the high-level BASIC language of the System 2200. Usage of the $GIO statement falls into two categories:

1. $GIO statements can be used to support the operation of peripheral devices and instruments specially interfaced to the System 2200 via the Model 2207A, 2227, 2250, and 2252 interface controllers. Being able to customize I/O operations is a valuable asset for anyone interfacing non-standard devices to the System 2200B. By using the $GIO statement and one of the interface controllers, the cost and time required to design, build, and test special interfaces for particular devices can be reduced or eliminated.
2. $GIO statements can be used to improve the operation of some current Wang peripherals. However, only those $GIO statements and programmed sequences specifically documented by Wang Laboratories should be used for such purposes. Examples will be furnished in the Option 2 Reference Manual.

Since the $GIO statement lets System 2200 users specify signal sequences for special input or output operations, the $GIO format accepts an optional comment inserted for identification of a particular operation (for example, WRITE, REWIND, READ). The comment is ignored by the system during execution of the statement.

OP-2 GENERAL I/O ROM

To customize an I/O operation, a microcommand sequence is specified in the $GIO statement -- either directly as illustrated in Example 1 (Line 90) or indirectly as illustrated in Example 2 (Line 90). In Example 2, the microcommand sequence is stored in the alphanumeric variable C$ in Line 20. The $GIO statements in Examples 1 and 2 are functionally identical. The indirect microcommand technique illustrated in Example 2 can be used to conserve space if a particular statement is used repetitively in one program. Also, the indirect technique can be used to change an I/O statement without repeating the entire statement.

```
Example 1
      10 DIM RS10, DS(20)60
      .
      .
      .
      90 $GIO WRITE/23B (6CFA 4400 A206 8607, RS) DS ()

Example 2
      10 DIM C$12, RS10, DS(20)60
      20 C$ = HEX(6CFA4400A2068607)
      .
      .
      .
      90 $GIO WRITE/23B (C$, RS) DS()
```

Each microcommand is represented by a four-hexadecimal-digit-code (two bytes). The first pair of digits specifies the type of operation and the signal sequence to be executed. The second pair of hex-digits specifies information such as the character to be output or the register containing the character to be output. Seventeen categories of microcommands are provided in Option 2 (several categories contain many subcategories).

The significance of the first and second pair of hex-digits for every category and subcategory of Option 2 microcommands is described in detail in the Option 2 Reference Manual. For example, the single microcommand represented by the code 8607 in Examples 1 and 2 instructs the system to execute the following composite operation: set the CPU Ready/Busy state to Ready, await the input of a single character from the I/O device, and save the character in Register 7 (the seventh byte of the variable R$).

Although the Option 2 ROM is designed primarily to support peripheral devices and instruments specially interfaced to the System 2200, Option 2 can be of general value when used with current System 2200 peripherals for data conversion and I/O device scanning operations.

The $IF ON statement provides the capability to test the ready/busy condition of a specified peripheral device and initiate a conditional branch to a specified line in a program being executed. Since an output-ready or an input-ready condition can be tested, the $IF ON statement provides a flexible method of scanning I/O devices in a System 2200 configuration. For example, if a hardcopy of an incoming data stream from a device interfaced to the System 2200 via a Model 2227 Telecommunications Controller is to be obtained by data output to a Wang Model 2221 Line Printer, a program can be written to load the printer line buffer with a line of data to be printed and then scan the printer (with an $IF ON statement) to see if the line has been printed while additional input characters are being received from the interfaced device.

The $TRAN statement provides a high-speed character conversion capability. Character conversion for a specified block of data is implemented via a table look-up procedure. By storing conversion tables in alphanumeric arrays and specifying a particular array in a $TRAN statement, any desired code-conversion algorithm can be programmed easily. Characters not in ASCII code can be received and then converted to the ASCII characters set used in the System 2200. Similarly, output data can be converted to any desired character code. An optional parameter R, if specified in a $TRAN statement, indicates that only selected characters (rather than all the characters) stored in a data block are to be replaced (translated). An optional mask, if specified, deletes selected bits in each byte of data before the translation is accomplished. The masking capability is useful for data editing and parity bit removal.

The $PACK and $UNPACK statements provide the capability to scatter (unpack) data from a record or to gather (pack) data into a record and simultaneously convert the data. That is, data can be taken sequentially from specified variables in an argument list, converted, and then packed into one alphanumeric array (record). Conversely, data can be taken sequentially from an alphanumeric array, converted, and then stored in specified variables in an argument list (unpacked). In a packing or an unpacking operation, the alphanumeric array (the record) is treated as a contiguous group of characters (that is, element boundaries within the array are ignored). Each field in a record can be identified uniquely by its position (expressed in terms of the starting character and field-length) or by defining special delimiters. Several different formats can be specified for the fields in one record. The $PACK and $UNPACK statements do not implement direct data transfer (sending or receiving) with respect to I/O devices; however, a $PACK or $UNPACK statement can be combined with a $GIO statement in a two-step operation designed to transfer data between the CPU (Central Processing Unit) and an I/O device. The $PACK and $UNPACK statements are especially useful when processing or preparing input/output records in formats required by non-Wang peripherals. The statements are of value in many System 2200 applications when data of different lengths and precisions are being packed or unpacked for optimal utilization of memory and peripheral storage areas.

ORDERING SPECIFICATIONS

A read-only-memory option for the Wang System 2200B providing five additional BASIC language statements designed to facilitate high-speed character code translations, to facilitate data packing and unpacking (by fields or delimiters) between a specified alphanumeric array buffer and specified variables in an argument list, to test the Ready/Busy signal of an I/O device and initiate a conditional branch to a specified line number, and to perform data input/output/control operations with a programmable signal sequence.

Standard Warranty Applies

WANG LABORATORIES, INC.
836 NORTH STREET, TEWKSBURY, MASSACHUSETTS 01876, TEL (617) 851-4111, TWX 710 343-6769, TELEX 94 7421

Printed in U.S.A.
700-3494
12-74-25M