INPUT/OUTPUT OPERATIONS

The MICRO 1600/21 and MICRO 821 provide three types of input/output:

Program-controlled transfer of data bytes via the Byte Input/Output Bus

Buffered concurrent transfer of data bytes via the Byte Input/Output Bus

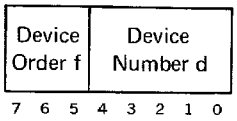
Direct transfer to memory via the direct memory access (DMA) channel

The Byte I/O Bus provides a path for transfer of data, control, and status between the processor and external peripheral devices. The direct memory access (DMA) channel communicates directly with memory.

BYTE INPUT/OUTPUT INSTRUCTIONS

Byte programmed input/output operations provides transfers of data, control, and status over the Byte I/O channel. This multiplex channel permits intermixed program and concurrent I/O transfers. More than one device on the bus may be operating in a concurrent block transfer mode at the same time. A maximum of 32 devices may normally be addressed on the Byte I/O bus.

The second byte of the instruction is a control byte which provides a three-bit device order and a five-bit device number as follows:



Byte input/output is basically a two phase operation. First, the control byte is placed on the output bus prior to the actual transfer of data. All devices examine the transmitted device number. The device, whose assigned number is the same as contained in the control word, accepts the control byte and performs the input or output of a single byte. When a device order does not require a data transfer, the second byte is disregarded by the device controller.

Device Address

Each device on the Byte I/O bus is assigned a unique five-bit device number. The numbers are assigned by means of selectively placed jumper wires on the printed circuit board of the device controller. The assigned device number is used by the device controller to compare against the device number of the control byte to determine if it is being addressed, and for identifying the device to the processor when requesting an interrupt or concurrent I/O transfer. Device number zero is always assigned to the parallel teletype interface.

Device Orders

The 3-bit device order specifies the type of I/O operation which will be performed. The device order accompanies the device number and is sent prior to each programmed transfer or to start a concurrent transfer.

Standard device orders designate the operations given in Table 2. Order codes 2, 3, 5, 6, and 7 are shown with their standard assignment, but they may be changed, depending on individual interface requirements.

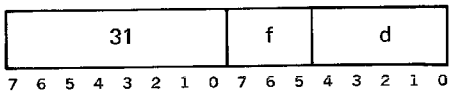
Status Bytes

The eight-bit status byte input as the result of a status order has four bits which are common to all devices and four which are device dependent. This byte is input to the A or B register or to memory by an input instruction with device order 1. The meaning of the status bits is given in Table 3.

INSTRUCTIONS

Three input and three output instructions provide for byte transfers between external devices and the A register, B register, or memory. When the transfer is to or from the A or B registers, only the eight low order bits are used. Interrupts or concurrent I/O requests are not recognized immediately following the execution of all Byte I/O instructions except, input byte to memory.

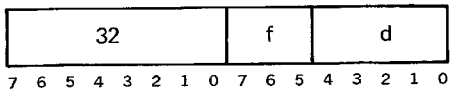
IBA Input Byte to A



The order code, f, is sent to the device designated by d. An eight-bit data byte is input from the device and placed in the low order bits of A. The eight high order bits of A remain unchanged.

Affected: A (low order 8-bits)

IBB Input Byte to B



The order code, f, is sent to the device designated by d. An eight-bit data byte is input from the device and placed in the eight low order bits of B. The eight high order bits of B remain unchanged.

Affected: B (low order 8-bits)

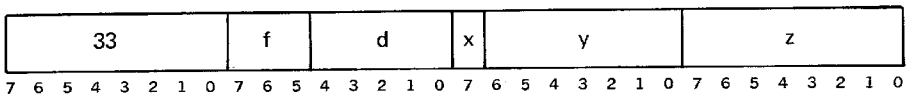
Table 2. Device Orders

Order Number	Operation	Description
0	Data Transfer	A data byte will be transferred between the addressed device and the processor. Direction of the transfer will depend on whether the instruction is an input or an output.
1	Status/Function	A status byte will be input from the addressed device or a function byte will be output to the addressed device, depending on whether the instruction is an input or an output.
2	Block Input/INT	The addressed device will start a concurrent block input to memory and will generate an external interrupt at the conclusion of the transfer unless the interrupt has been subsequently disarmed. This order should be sent by an output instruction.
3	Arm Interrupt	Permits the addressed device to make an external interrupt request upon the satisfaction of an interrupt condition. This order should be sent by an output instruction.
4	Disconnect	The block transfer in progress by the addressed device is stopped and end of block interrupt will occur unless the interrupt has been disarmed. This order should be sent by an output instruction.
5	Disarm Interrupt	Inhibits the addressed device from making an external interrupt request under any condition. This order should be sent by an output instruction.
6	Block Output/INT	The addressed device will start a concurrent block output from memory and will generate an external interrupt at the conclusion of the transfer unless the interrupt has been subsequently disarmed. This order should be sent by an output instruction.
7	Unassigned	This order, if assigned, may perform any required function as interpreted by the individual interface. If a byte transfer is desired the order may be sent by an input or an output instruction.

Table 3. Status Bytes Definition

Bit Number	Status	Description
0	Ready	This bit is a 1-bit when the external device is in a ready state.
1	Input Flag	This bit is a 1-bit when the external device has a byte ready for input to the computer.
2	Output Flag	This bit is a 1-bit when the external device is ready to receive a byte from the computer.
3	Error	This bit is a 1-bit when an error has occurred during a transfer. Errors may be timing, or device malfunction. This bit is cleared when the status byte is input.
4-7		Device dependent

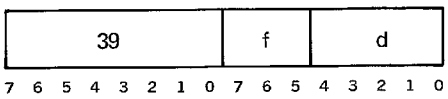
IBM **Input Byte to Memory**



The order code, f, is sent to the device designated by d. An eight-bit byte is input from the device and stored in memory at the effective memory address.

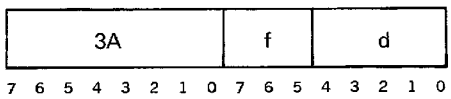
Affected: Memory

OBA **Output Byte from A**



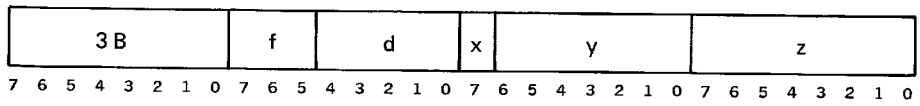
The order code, f, is sent to the device designated by d. The contents of the eight low order bits of A are output to the device. The contents of A remain unchanged.

OBB **Output Byte from B**



The order code, f, is sent to the device designated by d. The contents of the eight low order bits of B are output to the device. The contents of B remain unchanged.

OBM Output Byte from Memory



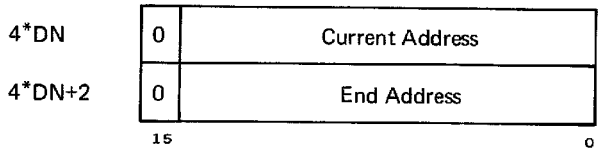
The order code, f, is sent to the device designated by d. The contents of the eight bit byte at the effective memory address are sent to the device. The contents of memory remain unchanged.

CONCURRENT INPUT/OUTPUT

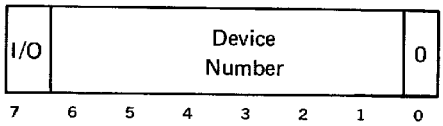
The concurrent I/O allows for block transfers between the external device on the Byte I/O bus and memory, at a maximum rate of 20,000 bytes per second. The transfers are fully automatic, and once started, proceed without program intervention. Concurrent I/O takes priority over instruction execution and forces momentary sequence breaks during executions of long instructions to insure that concurrent I/O delays are not excessive.

Address Control

Concurrent I/O addresses for each external device are controlled by a pair of two-byte address words. These two words are located in memory starting at an address equal to four times the device number. The first word is the current address (CA) and contains the address of the next memory byte to be used for the transfer. The second word is the End Address (EA) and contains the address of the last byte of the block. The first 128 locations in memory are reserved for storing concurrent I/O addresses for control of up to 32 external devices. The four bytes for each device have the following format:



When the processor detects a request for concurrent I/O, it inputs an externally supplied address (ESA) from the requesting device. This byte must contain a device address in bits 5-1, zeros in bits 0 and 6, and an output flag in bit 7. When bit 7 is a one, it signifies that the device is requesting an output transfer; otherwise an input is performed. The ESA is used by the processor to define the type of concurrent I/O operation requested and to locate the appropriate address control words. The ESA has the following format:



Concurrent Operations

Concurrent I/O operations are started by executing byte I/O instructions with the proper device order codes. These codes are given in Table 2. A block transfer can be performed with or without an interrupt at the end of the transfer. After a concurrent I/O operation is initiated by a processor instruction, byte transfers proceed automatically until the last byte of the block is transferred. Following each transfer, the processor increments the current

address. When the current address (CA) is greater than the end address, the processor automatically sends a disconnect order code to the device. This order code causes the concurrent I/O operation to cease and a device interrupt to be generated, unless it was previously disabled.

EXTERNAL INTERRUPTS

External interrupts originate with device controllers or interrupt modules on the Byte I/O bus. An interrupt module provides control of eight external interrupt signals. Device controllers may also generate interrupts to signify individual data transfers, end of operation, or error conditions.

The external interrupt system contains a single interrupt line, a priority line, and a select line. A device may initiate an interrupt request only if priority has been received from higher level interrupts on the priority chain. Devices not requiring interrupt service will propagate priority to the next device in line.

When the processor recognizes the interrupt signal, it enables the select line for the interrupt system. Each device in order will interrogate the select line and, if not requesting, will propagate this signal to the next device in line. Once the select signal has been propagated by a device, it will be locked out from acknowledging this signal until the select is removed. When the select signal is received by the requesting device, it will input its address on the data in bus. This ESA address may be six bits, (bits 6-1) since interrupt modules may take on interrupt addresses in the range of 32 to 63. The ESA address is used to locate the interrupt subroutine address located in memory starting at location 100_{16}. The processor reads this subroutine address and performs a call to the specified address. This entails storing all the operational registers into the push down stack and performing a jump to the subroutine address. Interrupts or concurrent I/O requests cannot be recognized before the execution of the instruction located at the subroutine address.