HONEYWELL INTEROFFICE CORRESPONDENCE

ENTER MAIL STATION NUMBER AFTER EACH NAME

194

DATE June 3, 1969

FROM R. O'Donnell, Dept. 263, Waltham - MS 456

DIVISION Computer System Operations

SUBJECT 3200 INPUT/OUTPUT SYSTEM

This preliminary copy of the 3200 Input/Output System is being circulated as a working document. It's content is not final and is, therefore, subject to change. Technical questions should be directed to the author as indicated on the cover page. Comments will be solicited by the author at a later date.

The Acknowledgment Receipt accompanying this document must be completed and returned to Engineering Publications, immediately. This copy has been issued in your name.

3200 INPUT/OUTPUT SYSTEM

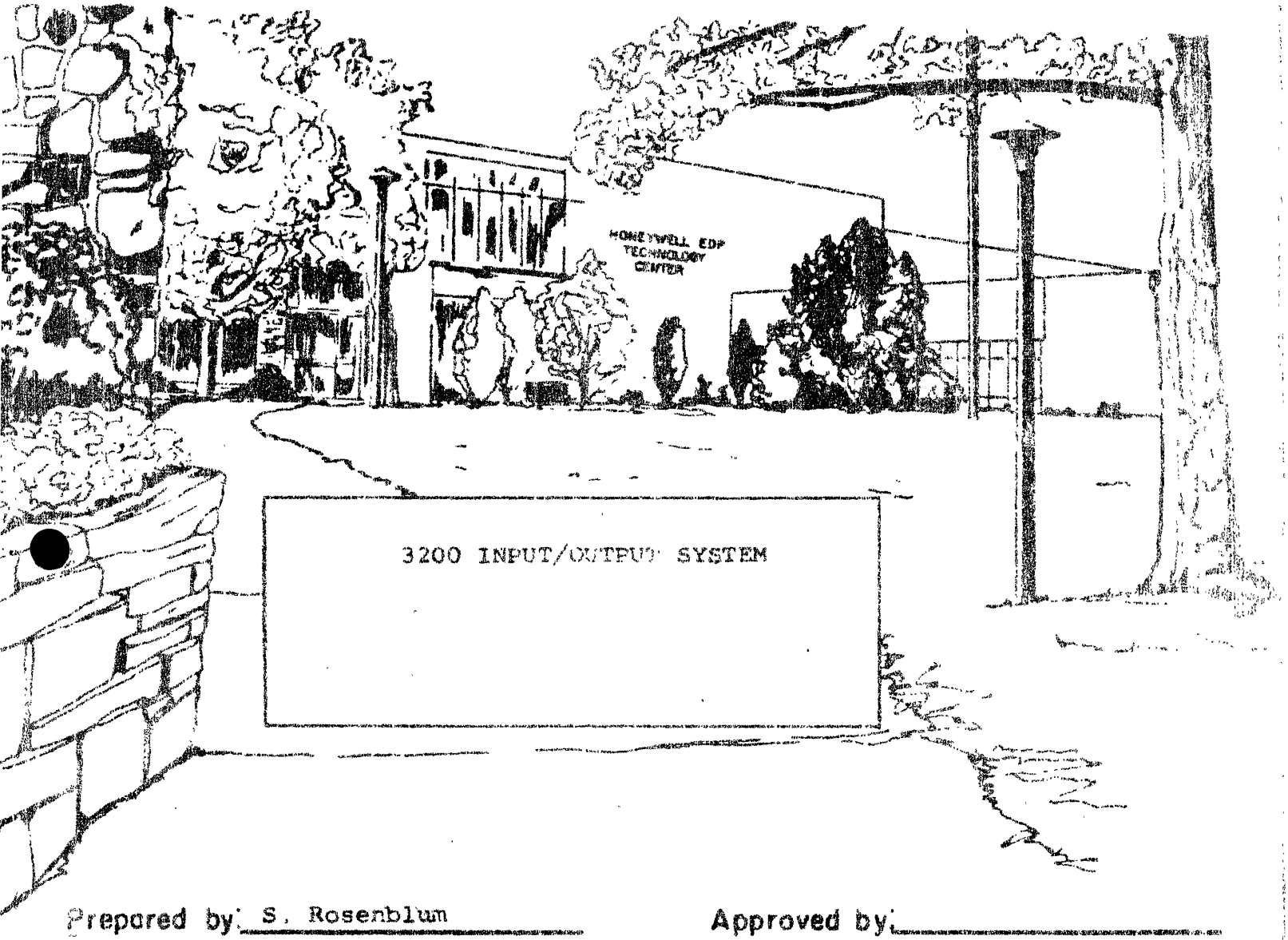
NAME	M.S	QUANTITY	DOCUMENT NO.
S. Alini	458		001
R Angel	472	20	002 - 021
C. Barbagallo	181		022
W. Bean	476		023
R Bender	181	5	024 - 028
K Bhavnani	476		029
C. Blanchette	461		030
E Boebert	476		031
T Bonn	408		032
G. Boothe	461		033
J Bradley	475	25	034 - 059
D Breden	476		060
R Broadbridge	479		061
R Brown	475		062
R. Brown	455		063
J. Busch	221		064
W. Buss	476		065
F Cardosi	472		066
R. Clippenger	480B		067
B. Colten	410		068
J Croy	476		069
T Cull	480A		070
F. Decker	251		071
D. Devoy	475		072
J Eachus	414A		073
E. Fabiszewski	410		074
R. Fairbrother	453		075
M. Ferdman	475		076
T. Fitzgerald	410		077
R. Gilstad	472B		078
L. Goldstein	477		079
H. Goldman	476		080
W. Gray	461		081

NAME	M.S	QUANTITY	DOCUMENT NO.
K Hakozaki (NEC)	461	5	082 - 086
C Harrington	461		087
A Hirtle	476		088
G Hoff	459	5	089 - 093
G Holt	476		094
T Holtey	476		095
J Homberg	475		096
D. Hudson	458		097
E. Hughes	458		098
E. Jackson	508		099
F. Jacob	408		100
T. Joyce	475		101
J Kanter	505	5	102 - 106
K Karger	186		107
J. Kent	458	4	108 - 111
R. Keyes	476		112
K. Krishnaswamy	475		113
R Leonard	418		114
W Lethin	461		115
J. Levin	480B		116
W. Lippold	H'Smitt		117
P. Liston	456		118
R. Little	476		119
N Lourie	461		120
C. Lynch	153		121
R. MacDonald	458		122
E. Masterson	410		123
E. McFaden	456		124
J Mekota	476		125
J. Melus	458		126
J Nackley	478	3	127 - 129
R. O Donnell	456		130
L Oliari	475		131
R. Oschman	446		132

NAME	M.S.	QUANTITY	DOCUMENT NO
N Papantonis	477		133
T. Pappas	181		134
R Pasciuto	458		135
D Peters	458		136
M Plager	458		137
C. Plumley	476	3	138 - 140
A Podesta	461		141
J Relyea	458		142
R Richwagen	221	5	143 - 147
G Rittenberg	476		148
J C Rogers	410		149
R Rosenblum	455		150
B. Rourke	475		151
L Ruthazer	418	3	152 - 154
M Sack	480B		155
S. Schwartz	479		156
J. Silverberg	440		157
J. Stohlberg	456		158
J Sullivan	456		159
V Virball	458	3	160 - 162
M. Weller	476	5	163 - 167
A Wolfson	461		168
R. Wuanrud	508		169
I Wyman	461		170
CED Master File	461		171
J Skoog	461		172
File Copies			173 -

Preliminary Copy
Circulating For Comments

Date: June 3, 1963
Doc. No: SY2P-02
Rev. DRAFT



3200 INPUT/OUTPUT SYSTEM

Prepared by: S. Rosenblum
SYSTEMS GROUP

R O Donnell
PUBLICATIONS

Approved by:

ELECTRONIC DATA PROCESSING
TECHNOLOGY CENTER

3200 Input/Output System

FORWARD

This document is an introduction to the concepts, philosophy and operation of the I/O portions of the 3200 System. It is intended as a companion document to the 3200 Systems description.

The document is not a specification. Any specific information on configurability, formats, encoding, etc., should be obtained from the appropriate functional specifications.

3200 INPUT/OUTPUT PHILOSOPHY

INTRODUCTION

Two fundamental elements of the 3200 System Philosophy are the creation of a division between data and procedure and the mechanization of multitasking. The 3200 I/O System, an integral part of the 3200 System concept, has made consistent use of the fundamental elements

The basis function of the 3200 Input/Output Subsystem is to allow several independent peripheral processes to operate concurrently with one or more central processor operations. Further, the shared access to main memory by central processor(s) and I/O controller(s) is proposed as the limiting factor in system throughput. Practically, the mechanisms which initiate and terminate multiple access to main memory also prove to be factors in system throughput. However by providing many independent mechanisms to conduct the initiation and termination functions, the amount of direct central processor involvement in peripheral operations is minimized

The 3200 System consists of a main memory with ports to each of one or more central processors or input/output controllers (IOC). The purpose of the IOC is to expand its single port to main memory so that it may be employed by several peripheral devices simultaneously. Additionally, because individual I/O data transfer rates are slow (mostly considerably so), but collectively may exceed the capacity of the memory port, the I/O subsystem must manage priorities of each peripheral access to main memory

I/O Operation

The central processor initiates and must be aware of all input/output activity. System design places responsibility upon the central processor tasks (programs) to issue those instructions which start I/O activity. If the initiation satisfies systems requirements in terms of availability of system resources and security restrictions,

responsibility for completion of the I/O process is relinquished to an I/O program under control of the IOC. The I/O program consists of an array of I/O commands, the structure of which bear no resemblance to their counterpart central processor instructions.

After completion of data transfer or control operations, the I/O command array notifies and returns control to the central processor. Needless to say, two I/O command arrays operating concurrently have no means of mutual communication and must rely upon the central processor for control of any mutual activity (data transcription).

Central Processor I/O Facilities

Two levels of I/O control exist within the central processor itself. While individual tasks may issue I/O instructions, a systems monitor or supervisor (collection of tasks) must oversee the assignment of I/O facilities as well as assign priorities, assure security and monitor completion and interruption signals from the (relatively) free running I/O command array programs. System elements such as the device specification table, traffic registers, simultaneity table, trapping mechanism and I/O start array provide the means for a systems monitor to carry out its responsibility.

The device specification table maintains information relative to every device attached to the system. In addition to transfer rate and individual logical status it contains directions for trapping or rejection of I/O instructions based upon device status or other factors. The traffic registers and simultaneity table automatically check to see that neither transfer capacity nor maximum level of simultaneity is exceeded each time an I/O instruction is attempted. The trapping mechanism and instruction rejection are used to ferret out instructions which represent exceptions to normal operation.

The I/O start array is the prime vehicle for communication from the peripheral devices to the central processor. It provides the central processor with an integrated hardware-software-controlled ability to distinguish among a large variety of device-generated sig-

nals. The mechanism steers the central processor to a task specialized for each signal, a task which can handle all signals or any level between the two extremes. Each device can conceivabley deliver up to 256 discernable signals.

Configurability

The multiprogram (via task swapping, I/O start array), multiprocess (via concurrent processing of I/O command array programs) approach to I/O system design provides much flexibility. The degree of flexibility is governed in large part by system configurability both in hardware and software.

The basic hardware configuration of the system relative to I/O operation is depicted as follows:

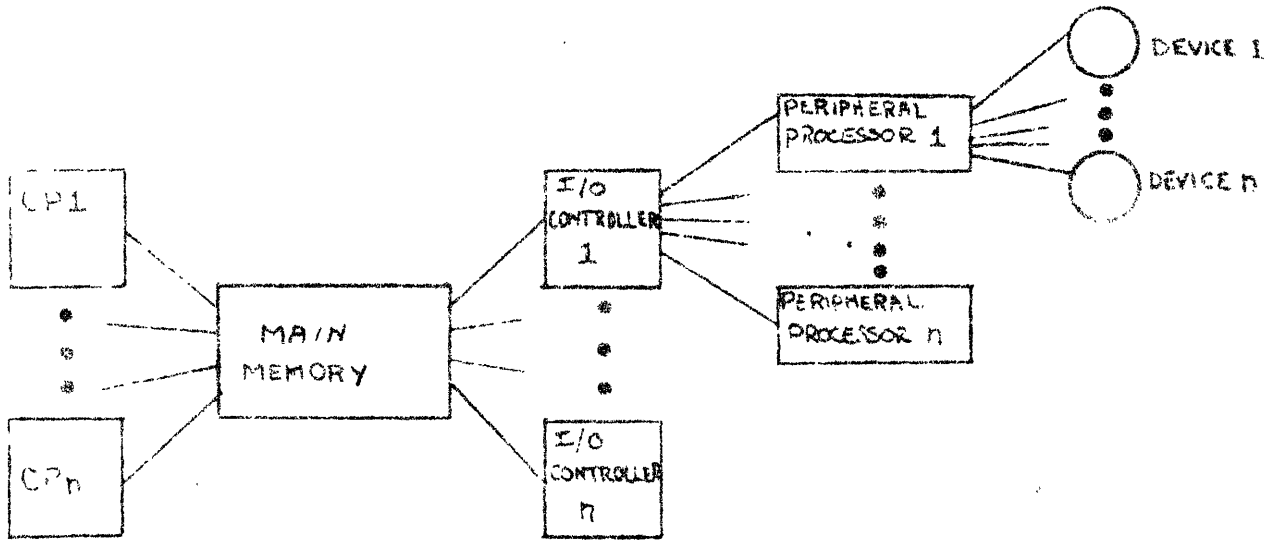


FIGURE 1 BASIC HARDWARE CONFIGURATION

The input/output controller, peripheral processor and peripheral device are those physical system facilities whose sole purpose is to conduct input/output operations. The numbers of IOC's, Peripheral Processors and devices which can be attached to the system or operated

simultaneously is the subject of detailed system specifications. Although IOC's are functionally identical, peripheral processors, and devices will differ according to the media with which they deal. Information on primary and secondary busses also may be found in other specifications Central processor facilities such as device and simultaneity tables and traffic registers will be available consistent with actual I/O facilities.

Software configurability will be contingent upon the creation of tasks to perform all I/O functions with efficient use of main memory storage By means of central processor task swapping and task priority mechanisms, it will be possible to use single tasks for many similar devices in serial fashion, thereby creating a many-to-one ratio of devices to tasks. The flexibility, however, enables the software to configure itself as required

Protection

The main element of 3200 protective philosophy is the allocation of data storage by means of structors. The structor defines the location, extent and characteristics of a data field When instructions are used to access or operate on data, they reference the structors rather than the location of fields themselves Provision of structors can be tightly controlled by a management facility such as an operating system.

This concept is extended to the I/O system by having I/O instructions reference structors just as the non-I/O counterparts. The I/O operands are thereby afforded the same degree of protection as data which resides in memory. Hence, it is possible to draw an analogy between the classic move instruction which moves protected data from one area of memory to another and the I/O instriction which moves a set of protected peripheral commmands to another protected area, the I/O subsystem.

A more explicit form of protection comes from the use of logical mask and logical status. These form a part in every I/O instruction.

The intention of logical status is to allow specification, within the instruction format, of a logical peripheral element such as a file, a volume, a remote terminal, etc. Although a proper physical path may be established to a device, comparison of logical status and logical mask restrict use of the device for the specific purpose of referencing a known logical element upon which the device is to operate.

The lock and key mechanism is used to protect a device from unauthorized access. A control mechanism such as a monitor or operating system can assign a device (via setting the lock) to a specific taks (to which it supplies a key) and thereby prevent its use by other non-proviledged tasks

Peripheral Processors

The peripheral processor complements the central processor and input/output controller in conducting input/output operations. As the ability of one I/O instruction to initiate processing of an I/O command array, so has the peripheral processor inability to assign one or several functions to a single I/O command. The availability of up to 256 individual device commands makes it possible to trade peripheral control between the I/O command array and peripheral device. In all cases, however, the central processor exercises ultimate control over all I/O operations, regardless of complexity.

SYSTEM ELEMENTS

Understanding of the 3200 input/output subsystem rests in large part on a familiarity with terminology describing the system elements. Although few completely new terms have been introduced, very explicit new meanings have been attached to familiar terms. Table 1 summarizes certain concepts and terms which appeared in the preceding section while developing a consistent language for use in understanding the system operation as described in the following section.

Other terms have been used within the document for descriptive purposes which may connote specific operations in the context of non-3200 systems. Such terms have been used liberally and no attempt at

definition will be made, lest it cause need for additional terms or create tight definitions not really intended Such terms as facility, resource, element, operation, pointer, interrupt, etc., fall into this category.

TABLE 1
DEFINITION OF TERMS

TERM	DESCRIPTION
BUS	A physical collection of wires destined to carry signals between two points, normally refers to the connection between the IOC and a peripheral processor, sometimes called the primary or I/O bus. A secondary bus may connect peripheral processors to peripheral devices.
CONDITION CODE	An eight bit field in the Procedure Index of the Task Status Block containing status relative to the I/O instruction just completed. The condition code should be checked following each I/O instruction.
DATA PATH	A list of facilities (IOC, I/O bus, Peripheral Processor, etc.) to be used in accessing a particular device. Depending upon the configuration, either more than two devices have the same eight bit address, or multiple paths are available to the same device. In both cases, further qualification is given by the data path.
DEVICE SPECIFICATION TABLE	A table in main memory containing information about each peripheral device connected to the system. See Systems Operation, Device Specification The entry is used during each I/O instruction.
FLAG	A field within each entry in the device specification table giving the central processor explicit directions to follow (trap, rejection, etc.) if certain conditions exist (device busy, no comparison of logical mask and status, etc).

(Continued)

TABLE 1 (Cont.)

TERM	DESCRIPTION
I/O COMMAND	A structor which contains information directing a device to carry out a single operation. The structor is meaningful only to the IOC. When several I/O commands are grouped together, they form an I/O command array which itself is a program executable by the IOC and peripheral processor.
I/O CONTROLLER	The physical unit permitting several peripheral processors simultaneous access to main memory. The IOC contains facilities (sequencers) necessary for simultaneity executing several I/O command array programs.
I/O INITIATED START	A signal generated by the I/O device or IOC for the central processor. The signal can cause steering to a particular task associated with the cause and nature of the start signal. An I/O start array contained in main memory points to all the tasks associated with I/O operations. The I/O start signal resembles a peripheral interrupt under certain circumstances
I/O INSTRUCTION	An instruction in the central processor repertoire used to initiate a peripheral operation. The I/O instruction may ultimately cause the execution of an I/O command array program in the I/O subsystems. However, the I/O instruction has explicit meaning only to the central processor.
I/O SUBSYSTEM	The collection of all physical and logical facilities and resources necessary to carry out peripheral I/O activity.
LOCK-KEY	A logical protection mechanism which allows the peripheral processor (or device) to require a positive comparison before allowing a peripheral operation to proceed. The lock is preloaded into the peripheral processor while the key is referenced in all I/O instructions

(Continued)

TABLE 1 (Cont)

TERM	DEFINITION
LOGICAL MASK-STATUS	A logical selection mechanism which allows the central processor to selectively access a peripheral entity such as a file or volume, normally exceeding the address obtained via device number or data path. The logical status is preloaded into the device specification table and the mask is referenced during the execution of the I/O instruction.
PERIPHERAL DEVICE	Single addressable data source or data sink. It may be a unit producing physical media such as a disk or tape drive or an electronic media such as a communication channel.
PERIPHERAL PROCESSOR OR CONTROL UNIT	This unit interfaces the IOC and one or more peripheral devices. Its responsibility includes interpretation of device commands, maintenance, update and reporting of device status, etc.
REJECTION	The halting of an I/O instruction for various reasons (busy, overflow of traffic registers, etc.) before responsibility is turned over to the IOC. Rejection causes the central processor to sequence to the next central processor instruction and stores the reason for rejection in the condition code field of the Procedure Index in the Task Status Block.
SIMULTANEITY (level, table)	The ability to conduct a data transfer between main memory and a device. Implies that several such transfers are being concurrently executed through a single facility (bus, IOC). The simultaneity table is maintained in the central processor to protect total system simultaneity capacity from overaccesses.
TASK	A list of instructions which form a logical sequence of operations. A task may be considered a program designed for a very limited purpose.
TRANSFER RATE	An indication relative to the amount of I/O bus capacity required by a particular device. A dynamic check of the availability of bus capacity is kept by a traffic register. Overflow causes rejection of the instruction.
TRAP	An automatic control transfer within a task which results from a hardware detected exception condition

SYSTEMS OPERATION

This section provides a description of the I/O System operation while avoiding the level of detail normally found in a specification. Two methods are used to achieve this end: 1) bit level description of registers or fields is avoided wherever possible, and 2) exception cases are either left out or covered in a cursory manner. When covered exception cases are relegated to the end of the section to avoid destroying the continuity of general descriptive material.

Assignment of Facilities

The ability to invoke I/O resources is managed by three facilities, one software and two hardware controlled. A software monitor system may manage the accessibility of peripheral devices via entries in the device specification table (see Figure 2). Although hardware actually allows or prevents I/O activity, the device specification entries are, in almost all cases, made by software. The traffic registers control assignment of peripheral bus transfer capacity and the simultaneity table oversees access to levels of simultaneity by hardware control. The three facilities are activated each time a task attempts to issue an I/O instruction. Successful initiation of the instruction depends upon satisfaction of the requirements of each management facility.

The management facilities, however, only represent physical resources within the system. The ability of the peripheral device, peripheral processor or input/output controller to perform the I/O instruction actually governs successful operation. The I/O instruction explicitly and implicitly specifies those resources which it desires to use through entries in the A and B operands of the instruction. The B operand specifies device number and data path while the type of instruction determines usage of a level of simultaneity. Protection of devices by key and resources by a logical mask prevent unauthorized use of either the device or bus transfer capacity.

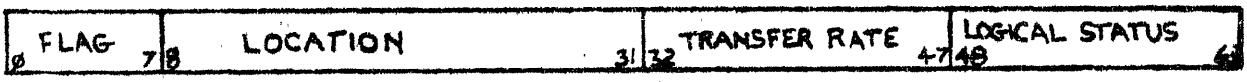


FIGURE 2. ENTRY IN DEVICE SPECIFICATION TABLE

Device Specification

The device specification table entry contains those fields which govern the initiation of the I/O instruction. The flag field is loaded by software to cause unconditional trapping or conditional trapping, rejection or acceptance based upon such factors as device busy, malfunction or comparison of logical mask and status. The flag field busy bit is automatically set when an instruction is successfully issued to a device.

Transfer Rate

The transfer rate is used by the traffic registers to detect system or resource overrun. Overrun and peak transfer rate causes instruction rejection.

Simultaneity

The simultaneity table is incremented following all other checks before execution is initiated in the device or peripheral processor. If all levels of simultaneity available to the resource or the entire system are exhausted, the instruction is rejected.

Condition Code

During the extraction of the I/O instruction, the device specification table is checked for busy or device malfunction. The presence of either condition will cause rejection and corresponding reporting via the condition code. A failure to compare logical mask and status will have the same result. Rejection is noted by a bit indicating the instruction was satisfactorally initiated. An instruction which checks the condition code should always be placed after an I/O instruction to detect failure to execute.

Device Identifier Structor

The B operand of an I/O instruction is the Device Identifier Structor (see Figure 3) It contains the key and logical mask and specifies resources such as data path and device number.

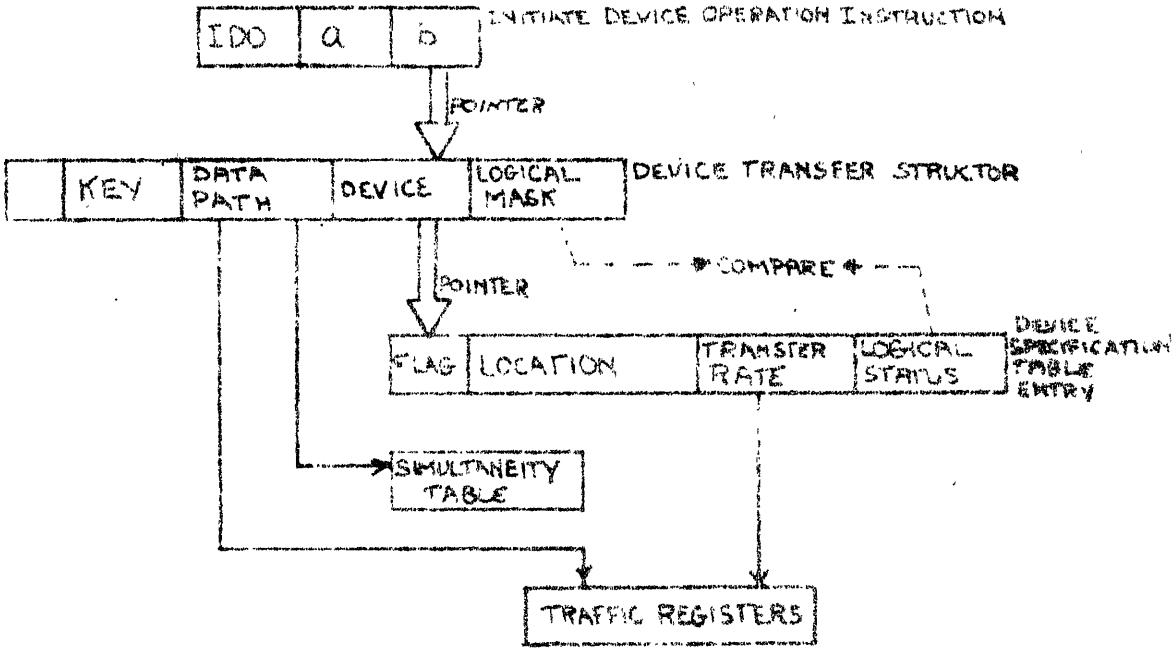
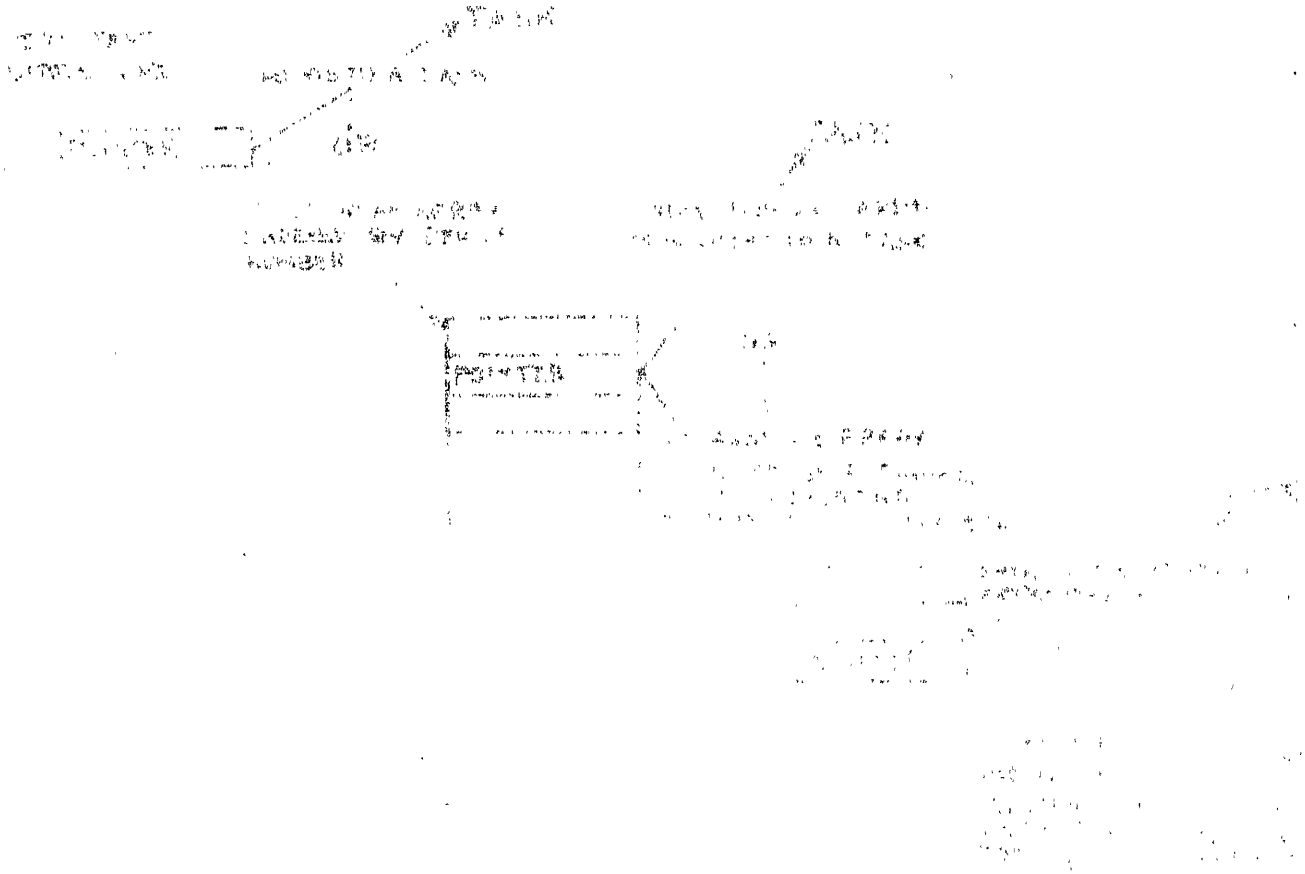


FIGURE 3. USE OF THE DEVICE SPECIFICATION TABLE

Task Algorithms

Before issuing an I/O instruction, the I/O start array (Figure 4) would have been loaded with pointers to preset software algorithms to accommodate any eventuality which could result from the execution of the I/O instruction. Normal and abnormal terminations (along with indexed variations thereof if the number and frequency of termination causes justify separate hardware steering), attention signals, residue indications and programmable interrupt entries in the I/O start array reference tasks programmed to react uniquely to each.



The tasks indexed by the I/O start array could instead be merged into one or more software tasks with varying accompanying levels of software steering. In any case, the steering capabilities of the I/O-initiated starts require preassignment of tasks to take advantage of the information which the I/O directs to the central processor.

The diagram of the mechanism which steers through the I/O start array has been oversimplified for clarity. Entries in any of the array elements point either to a Task Status Block (shown as a task), an array indexed by the status of the I/O device (device number, interrupt code, qualifier), a FIFO array (not shown), or a NULL Entry which ignores the signal. A more detailed description of the mechanism is given in the 3200 System Description.

If independent starts are expected to initiate the same task, a FIFO array must be employed to avoid potential loss of some I/O starts. This could occur because of independent starts from the same device or starts from several independent devices to the same task. The FIFO array automatically stacks the starts in the order in which they arrive.

Input/Output Instructions

The I/O instruction is the language element which a Central Processor uses to initiate one or a sequence of peripheral operations. It supplies some explicit information about the nature of the operation (within the op-code, but mainly reference additional information (operands) necessary for execution. The simplest instruction, a Halt Device Operation, references a Device Identifier Structor only. The most complex I/O instructions, in addition to the Device Identifier Structor, may reference an array containing up to 256 individual device commands. The individual device commands, in turn, normally reference fields in main memory to be used for subsequent transfer of data.

The Initiate Device Operation (IDO) instruction is used for most peripheral activity. There are three varieties of IDO each causing different demands on the I/O facilities. The normal data transfer specifies a device, a level of simultaneity and a portion of bus trans-

fer capacity. A second type, used for control, may be capable of execution without requiring a level of simultaneity. An example of such an instruction would be a rewind order to a magnetic tape drive. The third type of IDO, an Alter Operation, would be utilized to temporarily interrupt execution of an existing IDO in order to insert special information, either control or data, and then resume the interrupted instruction. This instruction is of value in a data communications environment. Although the instruction may require an additional level of simultaneity, it can use the same device number as the interrupted IDO.

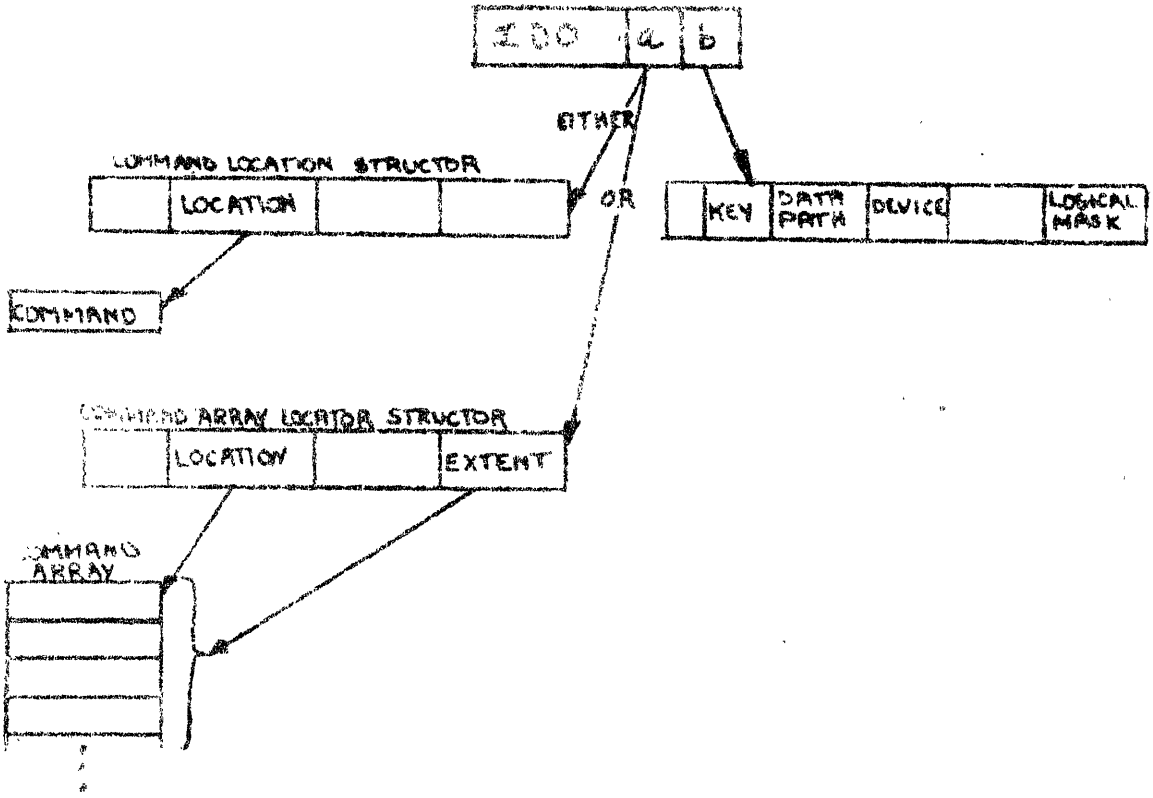


FIGURE 5. IDO INSTRUCTION

The IDO instruction's A operand delivers a locator of either a single I/O command or an array of I/O commands. When an array is addressed, the Locator Structor contains a field denoting the extent of the array. The B operand is the Device Identifier Structor, used in conjunction with the device specification table as explained under Device Identifier Structor.

Extraction

Successful extraction of the Device Identifier and Locator Structors rests upon the availability of resources, successful comparison of mask and logical status, and no rejection or trap. Successful, extraction results in the delivery of an I/O operand to the I/O controller, initiating execution of the operation herein. The I/O operand contains all information required by the peripheral processor and I/O controller in order to independently assume control of the peripheral operation(s). The I/O operand is structured as shown in Figure 6.

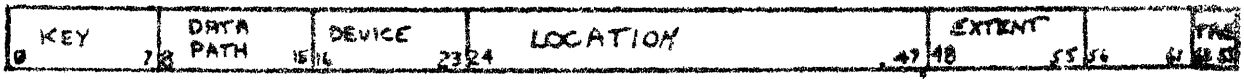


FIGURE 6. I/O OPERAND

Successful extraction also causes storage of the location in the device specification table as well as the setting of the busy bit in its flag field. At this point, the central processor is free to initiate a new instruction while peripheral processing independently proceeds.

Unsuccessful extraction, i.e., rejection or trap, will serve to terminate the instruction and allow the central processor to initiate a new instruction. Reports on the reason for termination are made through the condition code or the trapping mechanism (see 3200 Systems Description for information on trapping). The condition code should be checked in the instruction following the I/O instruction.

Input/Output Commands

The I/O command is used to cause a functional operation in a peripheral device. A single command or an array of up to 256 commands may be the result of one I/O instruction issued by the central processor. The command contains information on the type of operation, starting location of information to be transferred, amount of information to be transferred and a transfer variant which instructs the I/O controller how to control the operation and sequencing through the command array. Although I/O commands generally fall into the categories of data transfer, control or status inquiry operations, the differentiation between them is a function of peripheral processor assignment of device command codes.

The I/O command array is actually a stored program and the execution processor and I/O controller share responsibility in its processing. Processing within any one I/O command array is performed serially although signals such as programmable signals, device activated attention signals and residue signals can initiate parallel central processor activity.

The individual I/O command is structured as shown in Figure 7.

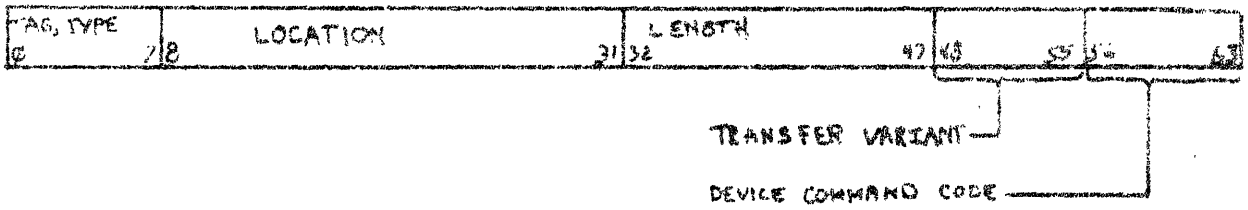


FIGURE 7. INDIVIDUAL I/O COMMAND

Device Command Code

Up to 256 device command codes may be specified for each type of peripheral device. The I/O command may be used to transfer data,

initiate control operations with or without accompanying control parameters, or extract status.

The termination of an I/O command is controlled by the peripheral device. However, the peripheral device may use the information that the data buffer in main memory has been exhausted or may use particular data codes to cause termination. In the course of termination, the peripheral processor may initiate a branch to other than the next command in sequence, if the transfer variant had previously allowed such action. The location to which the sequencer branches may be pre-programmed or optionally controlled by the peripheral processor itself.

The branching process may be used as in the following example:

> Having previously been issued a write data command, the peripheral processor, upon termination, may indicate either normal sequencing or branching to one or more error or alternative activities. The first type of error may be recoverable and could cause a retry of the original command. Another type of error, unrecoverable in nature, could cause the interrogation of device status in order to analyze the error before reporting to the central processor. One or more alternative next command solutions might be possible depending upon various types of (non-error-indicating) outcomes of the peripheral processor.

The branching capability, then gives the I/O Subsystem a large degree of independent decision making capability.

It must be noted that misuse of the branching capability could result in the execution of an endless loop of I/O commands. This so called "silent data" can be detected only in the central processor by means of a timeout. A common encounter with this problem could result from the repeated retry of a peripheral command which failed to successfuly terminate because of an error condition (bad

communications line, dirt on magnetic tape, etc.). The peripheral processor must maintain a retry counter if such a procedure is to be considered. Consult individual peripheral processor functional characteristics to verify the presence of the retry counting capability.

TRANSFER VARIANT

The transfer variant contains control bits which can monitor residue, unconditionally cause interrupt, allow peripheral or conditional branching, signal end of array, and give information on location of branches.

LENGTH, LOCATION

The length field will specify the number of bytes to be transferred and the location field, their starting location.

Examples of Command Array

a. The following command array might be used in a data communications environment where a station is dialed, a message transmitted, an acknowledgement checked, an end of transmission message sent and the line disconnected.

Device Command	Buffer	Termination Indicator	Examples of Typical Branching Conditions
dial	tel number		Retry if no answer up to 3 times; branch to +5 if no answer 3 times; go to next command if answer.
transmit	message		retry if negative acknowledgement received up to 3 times; branch to +4 if receive 3 negative acknowledgement; go to next command if positive acknowledgement.
transmit	end of Transmission message		
disconnect		X	
read status	error buffer		
disconnect		X	

b. In the course of accessing a record on a disk file, it is necessary to position the head and then search through the headers and keys on a track until the desired key and its associated data are located. At such time, the data may be read. In a sequential file structure, records are stored in ascending order related to their keys. Although the key codes need not be contiguous (all codes not present), they are arranged in order.

A peripheral processor for mass memory having the capability to compare header addresses and keys to stored counterparts need not cause interaction with central processor until a read and, perhaps, a new track or head seek are initiated.

Device Command	Transfer Variant	Buffer	Remarks
Search Header	branch * until comparison	)first header(	Search for header = ØØ
Search Key	branch * until ≥ comparison	)desired key(	Search for key ≥ K
Read		Data Buffer	
Seek	array termination	New Track	Position to new track in anticipation of next record to be accessed. Terminate Array.

Input/Output Initiated Starts

The I/O controller is capable of providing several types of I/O starts. These starts can cause hardware directed steering through the I/O start array to an entry representing a specific device, a specific type of start (termination, residue, attention, etc.) and a level of indexing qualifying the type of start. Each I/O start will cause transfer of an I/O status word. It will contain the device address and signal code (indicating type of start) as well as the qualifier.

NORMAL TERMINATION

Normal termination of the I/O instruction will occur when an I/O command is reached which contains indication that it is the last command in the array. An I/O initiated start signifying normal device termination will occur accordingly. The I/O status word will contain status (i.e., cause of termination, etc.) supplied by the peripheral processor as well as data path.

ABNORMAL TERMINATION

Abnormal terminations indicate that the array of operations could not be completed or was completed with an error or exception condition. The resultant I/O status word will contain pertinent status (cause or error, etc.).

PROGRAMMABLE SIGNAL

A Programmable signal will result from the setting of the appropriate bit in the transfer variant and can occur at any time during execution of an array of I/O commands. The I/O status word will deliver only the sequence location for which the signal occurred.

RESIDUE SIGNAL

Residue storage signals are an indication that a difference exists between the length field of an I/O command and the corresponding physical record which was transferred so far. The I/O status word will deliver the residue (positive for underflow, zero for overflow) and the array sequence location. The transfer variant or the peripheral processor status must have indicated an intention to register residue counts in order for the signal to occur.

ATTENTION SIGNAL

Attention signals may be used when the device is not presently executing an I/O operation or for special conditions not requiring termination. The I/O status delivers cause of interrupt.

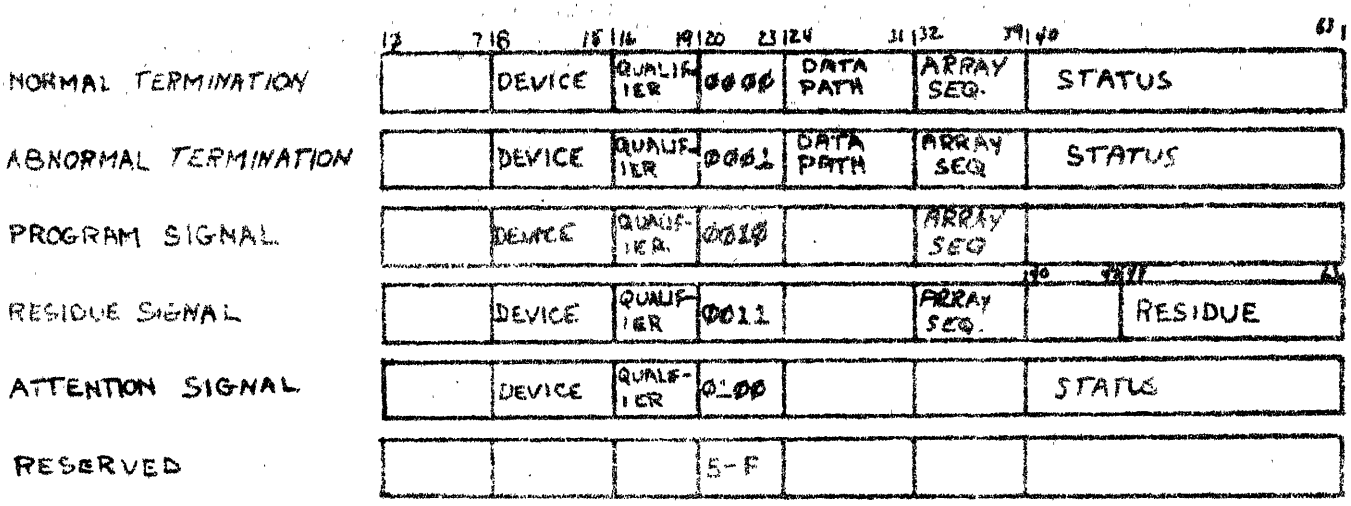


FIGURE 8. SUMMARY OF I/O STATUS WORDS

Execution of I/O Initiated Starts

The central processor receiving the I/O initiated start will sense the I/O status word for the cause of the start. Terminations will cause the release of transfer rate from the traffic registers and the busy condition from the device specification table flag. Terminations and other signals will then cause the central processor to attempt indexing of the I/O start array until a pointer to a Task Control Block is reached. That task will be checked for priority and executed if it meets the priority criteria.

Additional IDO Formats for Single Commands

An IDO instruction may specify only a single command. Such instructions may be used for the purpose of sending control information or inquiring of peripheral processor status. The B operand, device identifier structor, need not specify an extent (which would normally indicate the size of the I/O command array). Further, early release of facilities such as transfer rate and level of simultaneity can be achieved. The resultant I/O operand carries such information to the I/O controller.

Additional Instructions

The Halt Device Operation Instruction is an unconditional halt order to a device. Because no data is associated with this instruction, only the device identifier structor operand is required.