The organization of the input/output section and the control of input/output operations in SYSTEM/360 *are described.*

Emphasis is on the philosophy of control and on the reasons for choosing the particular logical and physical organization.

For each machine feature, the types of tasks requiring the facility are outlined and the significance of the solution is shown.

The structure of SYSTEM/360

Part IV – Channel design considerations

by A. Padegs

Input/output operations in SYSTEM/360 are performed by channels, control units, and input/output devices operating under the control of a supervisory program. The logical design considers the channel an independent entity which executes a program consisting of commands. The central processing unit (CPU) program starts channel operation by specifying the beginning of the channel program and the device to be used. The commands specify the direction of data transfer, the data source or destination in main storage, and all auxiliary functions associated with data transfer. When the channel program ends, the CPU program is interrupted, and pertinent status information is made available. This logical structure is independent of the physical implementation of the channel. Depending upon performance requirements, a system may contain several independent channels, and CPU and multiple channels may time-share common equipment.

Organization of the input/output section

External storage devices, as well as the equipment used to communicate with the external world, are referred to as input/output (I/O) devices, and the use of these devices by the CPU is referred to as an I/O operation. In addition to magnetic tape units and direct-access storage devices, such as disks and drums, I/O devices include card equipment, printers, inquiry stations, visual display devices, process control devices, and devices for receiving and transmitting information over communication lines.

The typical I/O device requires some control equipment unique to its particular function. Referred to as the control unit, this equipment is usually considered part of the I/O device. For some device types such as magnetic tape units, however, the control unit is shared by a number of devices and is considered a separate logical entity. The control unit also contains certain adaption equipment that makes all I/O devices appear identical to the rest of the data processing system.

Finally, the part of the system that ties I/O devices to the CPU and main storage is referred to as a channel. The channel contains the equipment necessary for attaching I/O devices, with their control units, to the system and for synchronizing I/O data cycles with those of main storage. The channel exerts programmed control over I/O operations.

requirements of I/O section

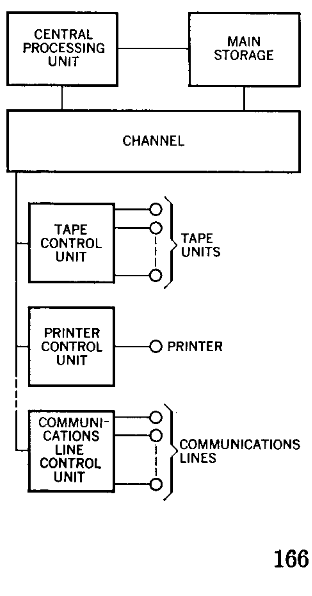
The organization of the I/O section reflects two main objectives of SYSTEM/360. The high-performance model of the system has main storage with an effective data rate of over 1,000,000 64-bit words per second, whereas the highest I/O data rate is 1,200,000 8-bit bytes per second (byps), with typical rates of 340,000 byps, 170,000 byps, and lower. Such disparity in data rates does not justify tying up CPU facilities for the duration of an I/O operation. Data processing in the CPU, therefore, must proceed concurrently with I/O data transfer, with as little interference due to the I/O operation as is consistent with economical use of equipment.

The other objective is that the system should be able to concurrently sustain multiple I/O operations. This requirement is imposed because the effective data rate of most I/O devices does not match the processing capability of the CPU, and, hence, concurrent operations are needed for efficient use of the system.

The requirement of sustaining multiple concurrent I/O operations does not apply to all configurations of CPU and I/O devices. In a system employing main storage with a physical word size of eight bits, certain I/O data rates may cause all CPU activity to be locked out, and the channel tasks of providing buffering and concurrency of operation disappear. In this case, it is not practicable to provide completely separate channel equipment. A similar situation exists at the other extreme of I/O communications. Data rates provided by manual keyboard devices are negligible compared to the access time of main storage and the speed of any electronic equipment, and again do not justify independent equipment for I/O control.

To accommodate the SYSTEM/360 configurations where I/O operations must be executed concurrently and independently of CPU activity, performance of I/O operations in the system is logically separated from processing of data in the CPU. Logically, sequences of I/O operations constitute independent programs, which the channel executes concurrently with the CPU program.

Figure 1 Organization of the system



However, when justified by system configuration, channel equipment is not necessarily independent. The equipment may be shared between CPU and I/O activity, and, depending on the type of I/O operation, the switching of the equipment may occur once for the transfer of a single byte of data, for the transfer of a burst of bytes, or for the execution of a whole I/O program. To preserve compatibility in the system, all channels are logically identical, and the execution of their programs is independent of CPU activity, regardless of any sharing of common equipment.

The components of the I/O system and their relation to the CPU and main storage are shown in Figure 1. Logically, the channel can be considered as an autonomous entity provided for the purpose of transferring information between I/O devices and main storage under control of the CPU.

communication with the channel

Input/output devices are attached to the channel by a standard connection, called the I/O interface. This interface is common to all channels and devices, and provides for the attachment to each channel of up to 8 control units and 256 I/O devices. The basic unit of information-transfer across the I/O interface consists of 8 data bits and an additional parity bit.

The connection between the channel and the main storage provides a path for the channel to fetch and store data as well as some control information that supplements the direct communications between the channel and the CPU. Except for channels intended primarily for low-speed operation, information is transferred across this interface in units of the physical word size of main storage, which varies from one SYSTEM/360 model to another.

The connection between the channel and the CPU provides a path for the CPU program and channel activity to exchange control information. This connection is used by the CPU to start and monitor channel programs, and by the channel to alert the CPU program of the progress and termination of I/O operations.

Function assignment

supervisory functions

SYSTEM/360 is designed to be used in conjunction with a supervisory program which allocates equipment to multiple programs and also monitors the execution of each problem program. To permit unrelated problem programs to execute I/O operations concurrently, the channel hardware, jointly with the supervisory program, must provide a means for assigning to each program the required I/O facilities. This assignment consists of establishing a path for exchanging control and status information between the program and the I/O facility and for transferring data between the I/O device and the designated area of main storage. The design must anticipate erroneous problem programs and provide means whereby one program can be protected from interference by another. Also, since the assignment of physical facilities, such as I/O devices and main-storage areas, to a program may depend upon the facilities currently available, the problem program should be able to specify the facilities symbolically.

Specifically, channel hardware and the supervisory program must jointly solve the following problems:

- I/O devices must be protected. A problem program should have access only to its assigned I/O devices and must be prohibited from writing (and sometimes from reading) on other devices. On such devices as magnetic tape units and disk storage, protection may be necessary for units of storage smaller than the entire recording medium, so that a program may be permitted access to some areas but not others.
- Main storage must be protected for I/O operations. An I/O operation may be permitted to transfer information only to storage areas assigned to its associated program.
- The program is often concerned with a particular recording medium, such as a tape reel, disk pack, or deck of cards, and not with the physical device on which the I/O operation is performed. To permit the allocation of I/O devices to be scheduled at the time the program is executed, facilities must be provided for the program to specify the devices symbolically. The system must translate symbolic names into actual device addresses.
- Since areas of main storage may be assigned to the problem program at execution time, the program must be able to specify data addresses symbolically. The system must translate symbolic storage names into actual addresses.
- When an I/O operation terminates, status information must be channeled back to the program that initiated the operation, even if the initiating program is not currently operating. Furthermore, the I/O device must be identified, and the extent of the main-storage area used must be communicated, in the original language of the problem program.

other supervisory tasks

A number of other interlocks and supervisory functions must be provided by the channel or the programming system. For direct-access storage devices, such as disks and drums, data transfer may involve the execution of a sequence of auxiliary operations, such as positioning the access mechanism and searching for the designated data block. These operations must be executed in the proper sequence, and no other operations may be interspersed. Two programs could interfere with each other during such a sequence by initiating I/O operations at the same channel, or, if the I/O device is accessible from more than one channel, by initiating I/O operations at the same device via two different channels. Therefore, facilities must be provided to protect a chain of I/O operations specified by one program against interference due to another program. The facilities should be such that no program can monopolize any I/O device.

One of the basic ground rules for multiprogrammed operation is that no problem program need be aware of the existence of other problem programs in the system. When an I/O device or its associated control equipment is shared among programs, the device may be busy to one program because of an operation initiated by another program. To remedy this interference, the system should queue requests for shared facilities.

Given sufficient storage, all of the above requirements can be met by the supervisory program. On the other hand, in a system that provides a uniform method of controlling all I/O devices, provision for most of them is feasible in hardware. As a result, one of the basic choices the designers must resolve is the division of functions between equipment and programming system. Specialized hardware, even in the presence of a supervisory program, reduces storage requirements for the supervisor and yields faster executions, whereas implementation of the functions in a programming system clearly reduces machine cost.

routine I/O service tasks

In the delineation of functions, other factors associated with the execution of I/O operations must be considered. Data transfer to or from I/O devices involves a large number of complicated routines, such as those required for blocking of data, analyzing error conditions, and scheduling overlap of I/O and CPU activity. Since these functions are common to most programs and are of routine nature, the problem programmer should be relieved of them. Ideally he should be concerned only with his data files, and the routine service functions should be considered part of the channel operation.

Most service tasks, however, are device-dependent. Examples are the checking of magnetic tape labels, back-spacing and re-reading tape upon data errors, positioning disk access mechanisms, and signaling a printer operator to replenish forms. Furthermore, as applications change, new service requirements are imposed. To provide a system that can adapt to changing requirements and to future attachment of new I/O devices, the service functions are best implemented as programs rather than as channel equipment.

resolving function assignment

Since service functions are usually required at both the beginning and the end of an I/O operation, inclusion of these functions in the programming system means that all communication between the problem program and an I/O device pass through the supervisory program. Furthermore, some of the services provided by the programming system, such as device scheduling, are closely related to device monitoring and can be readily integrated with the monitoring functions. Therefore, in SYSTEM/360, the main responsibility for monitoring I/O multiprocessing is placed on the supervisory program. Only those widely applicable functions that can be implemented without major additions to the basic channel equipment (e.g., storage protection and conditional branching in the channel) are provided in hardware. Any functions that are application oriented or cannot be integrated in the basic channel equipment are assigned to the programming system and are provided by a set of programs, called the Input/Output Control System (IOCS). Facilities are provided whereby programs can be readily switched, and thus the logical power of the CPU is made available for analyzing and reacting to conditions in the

I/O section. The channels are designed to operate in conjunction with the IOCS.

Logical channel organization

The architecture of the channel has two aspects: (1) the logical organization, which is the definition of functions and conceptual communication paths, and (2) the physical organization, which is the allocation of physical facilities. Since these aspects are independent, they are described separately here. The logical channel organization is presented in two parts: execution of channel programs, and communications between CPU and channels.

uniformity of control

The logical organization of the channel provides uniformity in attaching and controlling all I/O devices. Because of the standardization of connection between the channel and the I/O devices, the channel is not aware of the type of device attached. The device may be card equipment, a magnetic tape unit, or an adapter exchanging information with a remote terminal over a communications line. All these I/O devices respond to the same sequences of signals, and, except for some control information at the beginning and end of operation, the channel does not analyze the information transferred. For the channel, the different types of devices are distinguished only by the frequency at which they request or offer bytes of data.

Execution of channel programs

I/O operations are initiated by *commands*, which constitute the channel program. A command is specified in a channel command word (CCW) and is decoded by the channel.

reading and writing

The basic I/O operations are reading, which causes data transfer from an I/O device to main storage, and writing, during which data from storage are recorded at the device. In either case, the CCW designates the storage area by specifying the initial data address and the number of contiguous byte locations the area contains. Data in storage are placed in ascending order of addresses.

A variation of the basic reading operation is provided by the "read backward" command. This command initiates the transfer of data from the device to the channel, with the recording medium (such as magnetic tape) moving in the reverse direction, and causes data bytes to be sent to the channel in a sequence opposite to that on writing. To assemble the data bytes in storage in the original sequence, the command causes the channel to store the bytes in descending order of addresses, starting with the address contained in the CCW. In this case, therefore, the CCW designates the storage area by the highest instead of the lowest address of the area.

The amount of data transferred during an I/O operation is under the control of both device and channel. The device cannot transfer more data than specified by the channel program. When all allocated storage areas are filled or exhausted, the channel terminates the operation and requests the end signal from the device. On the other hand, when the device receives or transfers the block of data associated with the operation, such as a line of print or the information between two gaps on magnetic tape, it signals the end condition regardless of whether or not all storage areas are used up. Special indication is given to the CPU whenever the device ends the operation prematurely or the block at the device exceeds the storage area allocated for the operation.

The program specifies the type of I/O operation by means of an 8-bit command code in the CCW. The channel transfers the entire code to the device, and, for most operations, the code conveys all information needed by the device to initiate data transfer. A portion of the command code is common to all devices and indicates the direction of data flow to the channel. The remainder of the code depends on the type of device and specifies such conditions as density and parity.

On disk files, however, the program must position the access mechanism before data can be transferred. Similarly, use of magnetic tape units involves such operations as rewinding or backspacing tape, or loading a tape cartridge. These functions, unrelated to data transfer, cannot be readily combined with reading and writing without tying up channel facilities unnecessarily. Furthermore, these functions depend on the type of device and cannot be directly initiated by the channel without sacrificing a standardized channel organization.

controlling by orders

To maintain generality in the control and connection of I/O devices, all auxiliary control functions are specified by the programmer as *orders*. Orders are decoded by I/O devices, and the codes are transmitted to the devices in a control operation. The orders may be encoded in the command code of the control operation or, if additional information (such as the disk track address) is needed, they may be obtained from the storage area designated by the CCW. To the channel, a control operation is indistinguishable from writing.

When the device terminates an I/O operation, it provides an 8-bit status byte to the channel, indicating the general conditions detected during the operation. To preserve standardization in the I/O section, the status byte must be common to all I/O devices and cannot convey the detail conditions of termination.

sensing

The signaling of the detail conditions is standardized in SYSTEM/360 by making them available to the program in the sense operation. The sense command is a request to the I/O device for device-dependent status information, such as the position of magnetic tape, the condition of the card stacker and hopper, or the detailed error conditions detected in the last operation. This status information is transferred to the channel as data during reading and is placed in the main-storage area designated by the CCW. To the channel, sensing is indistinguishable from reading.

chaining

Commands are coupled into a channel program by chaining CCW's. Chaining is specified by two flags in the CCW, the presence of either of which causes the channel to fetch a new CCW upon completion of the activity associated with the current CCW. The CCW's are fetched from sequential locations in main storage, unless the transfer-in-channel command is encountered, which causes the channel to branch to the location specified in the CCW.

When data chaining is specified, the channel uses the new CCW to designate another data area for the original I/O operation. Data chaining affects only the allocation of storage areas to the operation; the device proceeds, executing the original operation, and is not aware of the fetching of the new CCW. Data chaining permits the reorganization of information as it is transferred between main storage and the I/O device.

When command chaining is specified, the channel uses the new CCW to initiate a new operation at the device. Command chaining reduces the frequency of communications between the channel and the CPU, and permits the CPU program to start with a single I/O instruction such sequences as printing multiple lines or reading multiple tape blocks. This type of chaining also provides a means to couple auxiliary functions, such as backspacing tape, to the data-transfer operation. Command chaining makes it possible for the channel to execute I/O programs of any number of I/O operations.

By using the IOCS, the problem program can avoid the tasks of assigning storage areas to I/O operations and setting up CCW's. Buffering by the IOCS is especially convenient for batch processing applications. For other types of applications, particularly those involving real-time processing and direct-access storage devices, the problem program itself should be permitted to set up CCW's before requesting the IOCS to initiate an I/O operation. However, to ensure that the I/O operation of one problem program does not destroy information associated with other programs, the supervisor program must reserve control over the storage areas used by a problem program.

storage protection

Since a storage-protection mechanism is needed for the execution of multiple programs in the CPU, protection is implemented in hardware and is extended to storage references made by channels. Implementation of storage protection for I/O operations permits the IOCS to initiate I/O operations without previously inspecting each CCW specified by the problem program and without relocating CCW's to the supervisor's area to prevent subsequent changes by the problem program.

Execution of certain I/O operations is contingent upon the results of the preceding I/O operation. For example, on direct-access storage devices, data transfer can be initiated only when the designated data block is underneath the recording or reading head. Usually the data block is identified by a key field immediately preceding the data area. To establish the relative position of the recording medium and the head, the system must match the identifier specified by the program with that appearing on the recording medium. The matching can be performed either by the channel or the device. It is important that the identifier not be buffered, which would restrict its size, and that the information be available from main storage during comparison.

conditional branching

In SYSTEM/360, the comparison is performed in the device, and the channel can be programmed to execute a closed loop of commands, thereby repeatedly sending the same information to the device. The channel remains in the loop until the device signals a successful match. Facilities are provided whereby the channel, in response to signals provided by the device, can perform a conditional branch, and thereby leave the loop to initiate data transfer.

Communications between CPU and channels

The CPU controls channel activity by means of *instructions*. Communications from CPU to channel are by four I/O instructions: START I/O, TEST I/O, HALT I/O, and TEST CHANNEL. Each instruction contains an address that identifies the channel and, when applicable, the device on the channel, such as a particular tape unit or communications line. In the case of direct-access storage units, such as magnetic disks and drums, each access mechanism is considered a separate device. To ensure IOCS control over I/O operations, I/O instructions are *privileged*, i.e., executable only by the supervisor.

starting the channel program

The CPU initiates the execution of a channel program for an I/O device by issuing START I/O. This instruction provides the channel with storage-protection information and the address of the first CCW, and causes the channel to establish a logical connection with the designated device. The CPU is involved with the operation until the device responds and until both channel and device verify that the operation can be executed. In some control operations that require no information transfer from the designated storage area, the operation is completed during the initiation sequence. In such cases, the channel signals completion during execution of START I/O.

Once the device is started and the channel is set up to execute its program, the CPU is released. In input operations, the channel accepts data from the device, assembles, when necessary, the 8-bit bytes into units equal to the physical word of main storage, and transfers words to the designated storage area. In output operations, the inverse process takes place, with the channel normally fetching full words from main storage and sending 8-bit bytes to the I/O device. The transfer of data to or from main storage and the initiation of new operations by the channel program do not affect the sequencing of instructions by the CPU. The CPU program is not aware of main storage references by a channel, except for the delay in execution caused by channel interference.

The CPU program, however, retains control over the channel program. When I/O activity must be rescheduled in response to conditions occurring after the channel program has been started, the CPU can issue HALT I/O. This instruction immediately terminates data transfer by the current I/O operation and suppresses further fetching of CCW's by the channel.

input/output interruptions

When the channel program is terminated, the channel interrupts the CPU program and makes available to the CPU a channel status word. This word identifies the last CCW used, indicates the amount of data transferred to or from the storage area defined by this CCW, and provides storage-protection information associated with the chain of operations. The status word contains also a status byte received from the device and a set of status bits provided by the channel, both of which describe the conditions of termination. At interruption, CPU control normally passes to the IOCS, which analyzes the information provided by the channel.

When such operations as reading or writing magnetic tape terminate, the channel and the device are freed simultaneously, and the channel program termination is signaled to the CPU by a single interruption. On other operations, such as line printing, the device may be involved with execution after data have been transferred to the device buffer. When all facilities involved in the execution of the last operation of the channel program are not freed concurrently, up to three distinct interruptions may occur, signaling the end of the operation at the channel, at the control unit, and at the device. By signalling the availability of the facility as soon as it is freed, maximum flexibility is provided for sharing facilities among operations.

When the program so indicates, the channel may also cause an I/O interruption during the execution of the channel program. The program can flag selected CCW's, thus requesting that the channel interrupt the CPU program upon fetching these CCW's. The interruption, which alerts the CPU program of the progress of the channel program or of the progress of data transfer within an I/O operation, can be exploited for programmed dynamic storage allocation.

controlling input/output interruptions

To provide the CPU program with a means of establishing in advance when conditions pending in the channel or in the devices should alert the program, a mask bit is associated with each physical channel. A masked channel cannot cause an I/O interruption, and consequently the CPU can suppress I/O interruptions by masking the channels. The conditions in the channels and devices are preserved until accepted by the CPU. The program can find out whether an interruption condition is pending in the channel by issuing the instruction TEST CHANNEL.

Channel masking allows the CPU to accept I/O interruptions selectively by channel. However, more than one I/O device can contain conditions pending to cause program interruption. The instruction TEST I/O allows a program to accept interruptions selectively by I/O device. This instruction gives the program the status of the designated device and clears any interruption conditions pending in the device. TEST I/O provides the same information as an I/O interruption.

Thus, two methods are provided for synchronization of CPU and I/O activity. The CPU program may normally keep all channels unmasked and depend on I/O interruptions for information concerning the progress of I/O operations. In this mode, the CPU is alerted to I/O events as they occur, and the channels are temporarily masked only to permit orderly handling of simultaneous interruption conditions. This approach is suited for real-time applications, such as process control or transmission of data over communication lines. The alternative mode of operation, which keeps all channels normally masked and interrogates devices by TEST I/O, permits the program to avoid interference due to conditions unrelated to the problem at hand.

Physical channel organization

Implementation of the described logical functions requires the channel to have some storage and a certain capability for arithmetic and logical operations. The storage facilities must accommodate a count, a main-storage data address, the address of the current CCW, and other control and status information. The channel must also provide storage for the buffering of data words during assembly and disassembly. The logical capability must provide for modification of addresses and the count, for assembly and disassembly of data, and for communication with I/O devices and CPU.

performance spread

To make the attachment of an I/O device economically feasible, channel implementation must keep the cost of device attachment commensurate with device data rate. The data rates of I/O devices are lower than that of main storage, but in spite of the disparity, a reasonable cost-to-performance ratio for channel equipment can be obtained for data rates down to the order of 100,000 byps. Thus separate channel equipment is justified for the data flow provided by most magnetic tapes, disks, and buffered equipment.

A different situation exists for such devices as unbuffered card equipment, where data transfer is of the order of 1000 byps; communications lines, for which data rates are measured, at most, in hundreds of byps; and manual input and output terminals, which operate at the rate of 10 byps and slower. Functionally, each of these devices must have the same channel facilities as tape units and disk storage, but the reduced performance desired for the lower data rates is not attainable at proportionately lower cost.

I/O equipment sharing

In particular, the problem in providing economical channel equipment for slow data rates is efficient utilization of the channel's logical capability and of the mechanical equipment required to attach the devices and support the logic. Since core storage affords considerable cost latitude, independent channel storage is economically feasible even for the low-speed devices. But no available technology provides a similar latitude in the cost and switching speed of logical circuitry. The system, therefore, must share equipment for the logical functions.

To provide for the sharing of channel facilities among operations, the data handling and storage facilities of SYSTEM/360 channels are conceptually separated. The portion of channel storage associated with an I/O operation is referred to as a *subchannel* and is the only part of the channel uniquely identified with the operation. The data handling capability of the channels may be shared among multiple subchannels, and, depending upon whether or not such sharing takes place, two types of channels are distinguished.

selector channels

Channels intended solely for high-speed operation contain only one subchannel. These channels are referred to as *selector channels* and are used for attachment of such devices as magnetic tape, disk, and drum units. A selector channel can sustain only one data-transfer operation at a time. Once a data-transfer is initiated, a logical connection is established between the addressed device and the channel. This connection lasts for the duration of the operation, such as reading or writing a tape block. The logical connection for data transfer is established always under program control, the device being "selected" by the channel. Other devices cannot communicate with the channel while a data-transfer operation is in progress, although they may, in the meantime, be executing such operations as backspacing a tape file or positioning a disk access mechanism.

multiplexor channels

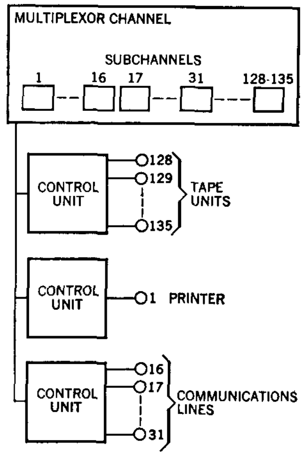
A channel intended for low-speed devices contains multiple subchannels. Such a channel is referred to as a *multiplexor channel* and may consist of as many as 256 subchannels. A multiplexor channel can concurrently sustain one data transfer operation on each of its subchannels. The logical and data-handling facilities of the multiplexor channel may be shared among the operations. A device involved in an operation stays logically connected to the channel and uses channel facilities only for the time required to transfer a section of information, such as a command code, a byte of data, or a status byte. Many such sections may be needed for the performance of an I/O operation and these sections may be intermixed for unrelated operations in response to demands from devices. Whenever, in the progress of an operation, information must be transferred, the device requests access to the channel and, when permitted by the priority assignment mechanism, identifies itself to the channel. The channel uses the address provided by the device to select the associated subchannel and performs the sequence specified by the device and the subchannel. The program is not aware of the multiplexing of channel facilities, and each subchannel appears to operate independently.

Each subchannel in the multiplexor channel is permanently associated with the corresponding I/O equipment. In the case of such devices as printers, inquiry stations, and communications lines, where control unit equipment does not restrict the concurrency of operation, one subchannel is assigned to each device, and the subchannel is identified by the device address. For such devices as magnetic tape units and disk access mechanisms, the control unit permits only one data-transfer operation to be executed at a time. In this case, a single subchannel is provided for the control unit, thus matching the data-handling capability of the channel with that of the control unit. All addresses that specify devices on the shared control unit cause the channel to refer to the same subchannel, and the subchannel appears busy when involved in a previously initiated data transfer.

multiplexor channel organization

The relation of the subchannels to I/O devices and control units is indicated in Figures 2 and 3, which show some typical I/O devices attached to the multiplexor channel, and some possible address assignments. As shown in Figure 2, certain sets of devices, such as tape units and communications lines, share physical control units and communicate with the channel over a single connection. Devices identify themselves to the channel by their addresses, which are then used by the channel to associate devices with subchannels.

Figure 2 Attachment of I/O devices to the multiplexor channel



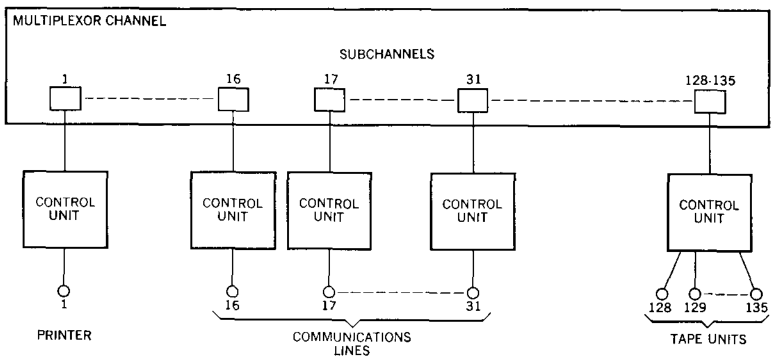
Since the program is not aware of the multiplexing of channel facilities, the single physical connection can be replaced by one conceptual connection for each subchannel. Furthermore, in the case of communications-line attachment, the control unit is time-shared for a number of concurrent operations. Hence, each subchannel can be considered as communicating directly with the associated device, as shown in Figure 3. In the case of magnetic tape units, all units share, conceptually as well as actually, a single control unit and subchannel, although each device is identified by a unique address.

In the case of the selector channel, the conceptual and physical organizations do not differ, since there is only one subchannel.

I/O and CPU equipment sharing

Sharing of channel facilities among I/O operations provides for an economical attachment of multiple low-speed I/O devices when the aggregate I/O data rate justifies separate channel equipment. To make this attachment economical where only a few low-speed devices operate, in low-performance models the equipment for logical and arithmetic operations is made common to both the CPU and the channels. When no I/O activity exists, the equipment is used for execution of the CPU program. When an I/O device requests service or the program encounters an I/O instruction, the

Figure 3 Conceptual relation between subchannels and I/O devices



contents of the required portion of the machine are replaced by the information of the pertinent subchannel, and the equipment performs the channel function. In the meantime, the CPU information is preserved in special local storage. At the completion of the I/O sequence, the I/O control information is restored in the subchannel, and, if no other I/O device is requesting service, CPU activity is resumed.

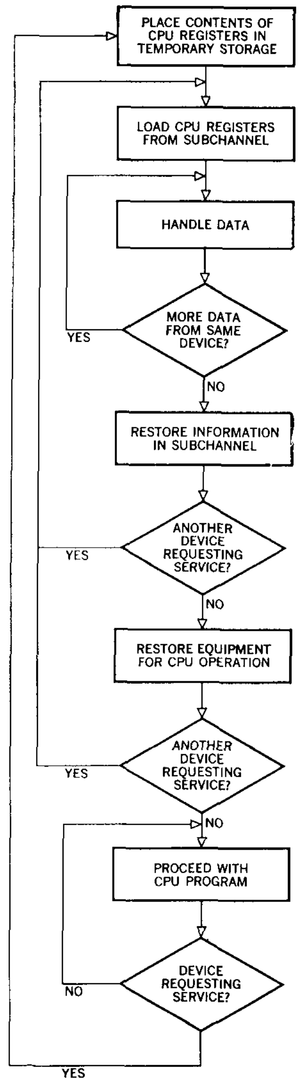
The dumping and restoring of CPU information as well as the fetching and storing of subchannel contents interfere with, and therefore limit, the productive work of the system. To reduce the frequency of switching of shared facilities, a buffered device may present bursts of bytes each time it is logically connected to the channel. The bursts must be sufficiently short not to affect the operation of unbuffered devices on the channel. Alternatively, when such a device as a tape unit is operated on the multiplexor channel, the data rate may be too high for the equipment to be time-shared, and a single I/O device with the associated subchannel may monopolize the facilities of the multiplexor channel for the duration of the operation. In this case, the multiplexor channel appears as a single selector channel, and the other subchannels and the CPU, when sharing the same equipment, are temporarily locked out. When the burst of data has been transferred, the channel reverts to multiplex operation, and buffered devices resume any previously initiated data transfer. The switching of the shared equipment is outlined in the flow diagram of Figure 4. The length of time each I/O device occupies the shared facilities is predetermined, and the program can schedule I/O activity in such a manner that interference among operations is avoided.

sample figures

The following figures of SYSTEM/360, MODEL 40 illustrate the effect of sharing equipment by the multiplexor channel. Operating at 400 cards per minute and an average data rate of 533 byps, the 1442 MODEL N1 card reader-punch occupies channel and CPU equipment 2.9 percent of the time, leaving 97.1 percent for CPU activity. This interference with CPU operation includes the time spent dumping and subsequently restoring the CPU; fetching, updating, and restoring the subchannel contents; communicating with the device; and transferring the byte of data. The interference figures assume that each of the above events occurs every time a byte of data is transferred. As more I/O devices request service, the multiplexor channel service cycles become more frequent, and less time is left for the CPU program. The channel data rate at which execution of the CPU program completely ceases depends upon the delays introduced by communications with the device. Assuming a device with very fast response times, with each device transferring one byte each time a logical connection is established, the capacity of the multiplexor channel is 50,000 byps. This rate can be achieved only when the I/O device next requesting service can be identified before the CPU is to be restored, so that the machine fetches the contents of the new subchannel immediately after storing the contents associated with the preceding byte service. At this rate, the CPU program is locked out, and no time is spent dumping and restoring the CPU. When the multiplexor channel operates as a selector channel, transferring data for only one subchannel, the maximum data rate is 200,000 byps. It should be noted that the above rates are provided only as rough approximations for illustrative purposes. The exact performance of a given channel depends upon additional factors such as the type of device, the order and the distance of the devices in relation to the channel, whether reading or writing takes place, and whether data chaining is specified in the program.

From the programming point of view, the selector and multiplexor operate identically, and the program need not be aware of the sharing of equipment among I/O operations and between I/O and CPU activity. Except for those aspects of communications pertaining to the physical channel, such as the control of I/O interruptions, a subchannel on the multiplexor channel is identical to a selector channel. The multiplexor and selector channels are identical also with regard to the attachment of I/O devices. Both channels match the same I/O interface, and each channel can accommodate up to 256 I/O devices. A program written for an I/O device can be executed on either type of channel with identical results.

Figure 4 Switching of equipment between CPU and I/O activity



Summary

Control of I/O operations in a multiprogrammed system involves many complicated functions, such as allocation of facilities and protection against mutual interference among programs. In SYSTEM/360, most responsibilities associated with multiprogramming I/O operations, as well as all device dependent functions, are assigned to the programming system, and channel equipment is designed to operate in conjunction with a supervisory program. Channels permit overlapping of CPU and I/O activity and provide general functions more economically implemented in hardware than in software.

Input/output operations in SYSTEM/360 are controlled by instructions, commands, and orders. Instructions, decoded by the CPU, initiate and monitor the channel programs. Commands, decoded by the channel, specify such operations as reading and writing. Any functions depending on the type of device, such as rewinding tape, are specified by device-decoded orders, which are transferred to the devices in a control operation. The flexibility of the I/O system is enhanced by such features as conditional branching in the channel program, protection against unauthorized storing of information in main storage, and the facility to transfer data to or from noncontiguous locations in main storage. The channel can alert the CPU program of I/O operation status by means of the interruption mechanism.

The logical channel organization is implemented in nine distinct channel designs. Economical implementation of the logical organization is achieved by sharing equipment whenever justified.