G. R. Ahearn
Y. Dishon
R. N. Snively

Design Innovations of the IBM 3830 and 2835 Storage Control Units

Abstract: The IBM 2305 Fixed Head Storage and IBM 3330 Disk Storage provide significant performance improvements over previously available disk facilities. Many of the improvements were made possible by the design of the control units, which are complex systems that integrate analog and digital interfaces. Definition of control unit requirements and characteristics permitted a large degree of commonality to be achieved between the two control units. The available design alternatives and implementations are discussed.

Introduction

The IBM 3330 Disk Storage using an IBM 3830 Storage Control, and the IBM 2305 Fixed Head Storage, using an IBM 2835 Storage Control, are new high-performance disk drives designed for IBM's larger data processing systems. The new performance capabilities are extensions of the well-known principles applied to previous IBM disk facilities, especially the IBM 2314 and IBM 2301. A large part of these new capabilities are directly related to the design of the control units. Table 1 shows a performance comparison between the previous designs and System/370 facilities[1-3].

The control unit of a disk facility connects the individual disk drives to a well-defined channel interface. In the case of the 3830 and the 2835, the interface is the new System/370 Block Multiplexer Channel[4]. This channel is an evolution of the System/360 interface that allows greater system throughput and a higher data rate. The block multiplexing feature of the channel, that is, the interleaving of portions of an operation, is an important factor in improving disk facility performance. The control unit accepts selection from one of two channels, establishes connection to the requested device, decodes the command from the channel, and controls the transfer of data to or from the device.

In providing these functions, the designers attempted to reduce costs by maximizing the number of circuits common to the two control units. They also attempted to provide extensive error-detecting and logging circuitry in order to increase reliability and ease of maintenance.

The purpose of this paper is to highlight the design innovations in the 2835 and 3830 control units. The design of a high-speed disk control unit does not, of course, concentrate on one single problem; rather it integrates the requirements of data processing, recording technology, coding techniques, and interfacing other units into a complex system that interacts with both the input/output channel and the disk drive. Some of the design problems can be described only briefly, since their full exposition would require separate papers. Only a few papers have been published in the past that address themselves specifically to disk storage control unit design[5]. The general principles of mass storage enumerated in a review paper by T. H. Bonn[6] some five years ago are still very applicable.

Design architecture

Many of the architectural improvements for System/370 were determined by the characteristics of disk storage equipment. To take advantage of these characteristics, the 2835 and 3830 control units were designed to make use of the following System/370 architectural extensions: block multiplexing, command retry, and high-speed data transfer. Each of these is covered in the following sections.

- *Block multiplexing*

The new control unit should be capable of disconnecting from the channel during those periods when information

Table 1 Comparative data on disk facility performances.

	IBM 2314	*IBM 3330*	*IBM 2301*	*IBM 2305 Model 1*	*IBM 2305 Model 2*
Data rate	312 kbyte/s	806 kbyte/s	1.2 Mbyte/s	3.0 Mbyte/s	1.5 Mbyte/s
Average access time	60 ms	30 ms	8.6 ms	2.5 ms	5.0 ms
On-line capacity/ disk pack	29.7 Mbyte	100 Mbyte	4.1 Mbyte	5.43 Mbyte	11.26 Mbyte
Rotation period	25 ms	16.7 ms	17.1 ms	10 ms	10 ms
Rotational position sensing	No	Yes	No	Yes	Yes
Error correction	No	Yes	No	Yes	Yes
Facility error logging	No	Yes	No	Yes	Yes

is not actually flowing. This capability is provided by using the new block multiplexing feature of the I/O channel. The Record Ready function of the disk facility may serve as an example. This function is implemented with a new I/O channel command, SET SECTOR. The command transmits a one-byte argument that specifies the angular position on the disk of a particular record. Upon receipt of this command, the control unit disconnects from the channel with Channel End status. The disk device monitors its own angular position and as the disk approaches the position specified by the SET SECTOR argument, the control unit, if not otherwise occupied, interrupts the channel and presents Device End status for the SET SECTOR command.

If the channel is available, reconnection occurs and subsequent commands then process the record found in the approaching angular position. During the disconnected period the channel is free to communicate with any of its other control units or devices, and the control unit may accept commands from the I/O channel.

The Record Ready feature of the control unit may be extended to allow the channel to communicate with a single device as if it were several separate devices. Several SET SECTOR arguments, each identified by a different logical device address but actually referring to the same physical device, are set into a microprogram-scanned table. When the angular position of the device corresponds to one of the SET SECTOR arguments, an I/O interrupt is initiated against the corresponding logical device address. The table is stored in writeable control store and contains the file mask, track and record address, and SET SECTOR argument for each logical device address. This concept is implemented only in the 2305 because of its fixed-head architecture.

Another example of disconnection between channel and control unit occurs during execution of a SEEK command within a chain. During the time that the read/write heads of the 3330 disk facility are mechanically positioned, the control unit and I/O channel are free to operate with other devices. The block multiplex feature allows the chain to continue when the control unit presents Device End status, indicating that the SEEK is complete.

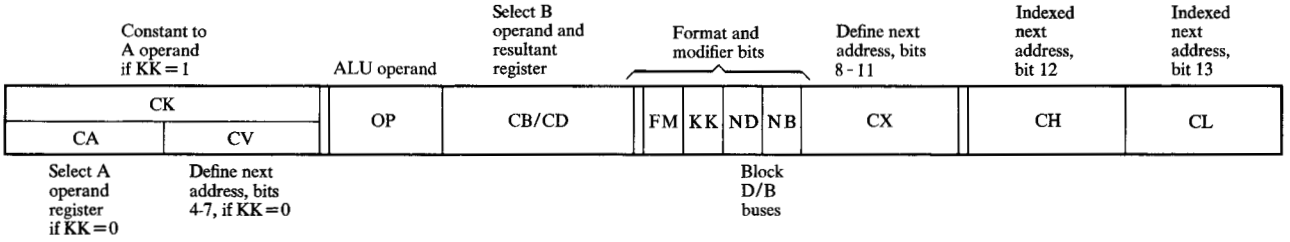
• *Command retry and error recovery*

The new control unit should be able to repeat a command that was not successfully completed. This ability should be independent of the program operating within the system. Experience shows that most errors do not recur when the command is retried, since the errors are often caused by external electrical noise or similar intermittent conditions.

In prior equipment, retry was performed by error recovery programs that repeated the entire chain of commands from the beginning. It was often difficult or even impossible to reissue a specific command because the mechanical position of the I/O equipment was dependent on commands which precede the one that failed.

A significant improvement of the 3330 disk facility is the capability of offsetting the recording head by a fraction of a track width in either direction from its nominal position in order to recover the data from a record that initially was read with an uncorrectable error indication. This capability is not required for the fixed-head, nonremovable disk 2305. The procedure thus allows the control unit to retry any read-type command, in a way that is transparent to the using program.

Some hardware errors prevent proper microprogram execution, making normal retries impossible. The System/370 I/O interface provides for the initiation by the control unit of an I/O Error Alert sequence. The channel then issues a SELECTIVE RESET command that generally enables the control unit to recover its processing capability if the error is not permanent. System error recovery



(Address bits not defined by microword are carried over from previous instruction)

Figure 1 Microword format.

procedures must then be executed to restart the channel command chains. The error is logged to permit later analysis by the customer engineer.

• *High-speed data transfer*

System/370 I/O equipment should be capable of transferring data at rates in excess of three million bytes per second. The data transfer rate is limited by the physical delays introduced by signal cable and interface logic circuitry. System/370 architecture circumvents these limitations by providing a data interface two-bytes wide between the channel and control unit. Two additional tags, called Mark lines, identify the width of the data transfer bus. The bus extension and the additional tags are used by the high-speed model of the 2305 but not by the 3330 disk facility.

System/360 channel architecture imposes another limit on cable length and data transfer rate by interlocking each data transfer with the Service In and Service Out tag lines. A single data transfer using the total interlocking of these two tags requires four cable propagation delays. With this interlocking scheme, the 3330 and 2305 disk facilities would have been restricted to shorter cables and lower data transfer rates.

System/370 architecture alleviates this limitation by providing two additional tag lines called Data In and Data Out. The interlocks are arranged so that byte transfers are alternately signaled by Service-In/Service-Out and by Data-In/Data-Out. A single data transfer then uses on the average only two cable delays, and either cable length or data rate can thus be doubled with respect to System/360 limitations.

Common control unit architecture

When the increased capabilities required of the control unit are examined, it becomes evident that a large quantity of status and control information has to be stored. In order to store the necessary information required for command retry and data error correction, a scratch pad memory would probably suffice. However, to provide the additional requirements of error logging, diagnostic inline programs, record ready and other functions, a writeable control store becomes necessary. Such a store also makes future update and debug of microprograms very easy, provides flexible solutions to many engineering problems and contributes to the commonality of the 2835 and 3830 control units. The writeable control store is implemented as part of an independent processor used in both control units.

In the choice of a control store, the main factors were cost, availability, speed and reliability. The monolithic memory chosen is a compact, high-density, fast storage device designed especially for control units or high-speed buffers. It is organized in array cards of 4000 × 2 bits. Any word size may be selected, subject only to packaging limitations. A word size of four bytes (32 bits) was determined by considerations of microprogram format and control processor architecture. A total capacity of 4000 words was considered adequate to handle the required functions, including a two-channel switch feature. In considering these factors, much thought was given to previous experience with the 2314 control unit.

The microword format was defined to minimize the microprogram word length and the hardware required to decode the micro-instruction and at the same time to reduce the number of words required to execute a microprogram. Each microword generates the control store address from which the next microword will be fetched, eliminating the need for an instruction counter in the processor. This enhances the processor speed and its ability to control the high-speed data stream between the I/O channel and the disk device. It also permits a very simple processor data flow. The address-carrying scheme provides a powerful branching capability simply by modifying some of the low-order bits. A four-way indirect branch is provided by interrogating selected status bits in the processor. Further, microword format fields for ALU operations, operand and resultant registers, and constants are defined.

An example of a microword format is shown in Fig. 1. Twelve such formats are provided to permit variations of

Table 2 ALU operations.

OR
AND
Exclusive OR
Add A to B, Don't Save Carry
Add A to B, Save Carry
Add A to $\overline{B}$ and Carry, Save Carry
Add A to $\overline{B}$ and Carry, Save Carry
Subtract B from A

operations such as status recording, fetch/store operations, extended register selection capability, or unique special operations to activate particular hardware.

Most processor operations monitor or set control bits, make logical decisions on a combination of status conditions, count, manipulate data on a bit basis, and transfer data between registers. Because memory fetch/store operations are designed on a four-byte basis, while ALU operations are on a one-byte basis, there are relatively few accesses to memory for data compared to accesses for instructions. The ratio is typically 1:20. A relatively large proportion of machine cycles will be spent on processing data already present in the control registers. A simple repertoire of eight ALU operations suffices, as shown in Table 2. Conspicuously absent from these operations are any shift instructions. The lack of such instructions is compensated for by the powerful logical branching capability of the processor, which makes shifting and testing bits unnecessary. Also, there is no need for multiplication and division arithmetic.

Figure 2 shows the processor data flow. The design of this architecture evolved around the microword format and was influenced by the experience gained wih the 2841 and 2314 control units. The processor consists of an ALU, two operand registers, control registers, a control store data and addressing interface and decoders for the microword format, ALU operations, and branch conditions. Four of the control registers are connected to the control store data interface. Two registers serve as status registers, in which the bits may be set by the microprogram or by the channel or device interface logic.

All data processing is done on a byte basis, but 4-byte words are used for fetching or storing data from the control store. The microprogram has the option to select only one data byte out of the four fetched. Each microword instruction is executed within one machine cycle, which includes fetching of a new microword from control store. The control store has a nominal 145-ns access time. Microinstruction address decoding requires an additional 45 ns. Therefore, a 200-ns machine cycle was chosen, including 10-ns contingency. The architecture of this control unit did not require that the processor have an interrupt mechanism. The only forced branches occur after a reset is issued by the I/O channel.

Care was taken to provide sufficient error indications to localize the faults to a minimum number of replaceable units. About 25 percent of the processor circuitry is devoted to this task. The methods used are duplicating critical circuits, predicting ALU parity, and providing logic circuitry to detect error conditions. Errors in the processor are critical since they prevent further reliable execution of the microprogram. Such errors are designated as primary errors, in distinction from secondary errors that occur elsewhere in the control unit. Errors may be logged and commands may be retried as long as the microprogram processor is still functional.

The various primary errors (clock failure, ALU parity error, control store error, etc.) set a two-byte wide error register. When such an error occurs, the machine clock stops and the channel attachment initiates an I/O Error Alert sequence. The channel responds with a SELECTIVE RESET which restarts the processor and forces a branch to an error recovery routine, saving the instruction address associated with the error and the contents of the error register. This information is later used for maintenance analysis in order to detect marginal operation and prevent a solid failure of the control unit. If a primary error is permanent, the control unit disengages from the channel interface so that the I/O channel can continue to communicate with other control units.

By choosing a volatile monolithic writeable control store, one is confronted with the problem of how to load the control microprogram after starting up the control unit. The three alternatives considered were to load the control store 1) from the channel, 2) from the disk device or 3) from an independent low-cost disk or casette tape drive. The first two alternatives were rejected because there was no control over the primary storage medium. Further, to communicate with either channel or device, an extensive hardware interface would have to be built, and it is this very hardware that is otherwise controlled by the microprogram.

Since the 3330 usually holds the operating system, loading from the channel would require a stand-alone program, loaded either from cards or from tape. If the control microprogram were stored on the disk device, expensive bootstrapping control circuitry would be required. In addition, the microprogram would be very difficult to maintain, since every disk used on the machine would have to contain the correct microprogram for that machine. Correct microprograms would be impossible to guarantee because of the interchangeable disks used with the 3330 disk facility.

The third alternative was finally adopted, and a miniature microprogram load, read-only disk drive, using a small interchangeable plastic disk, was selected. Data

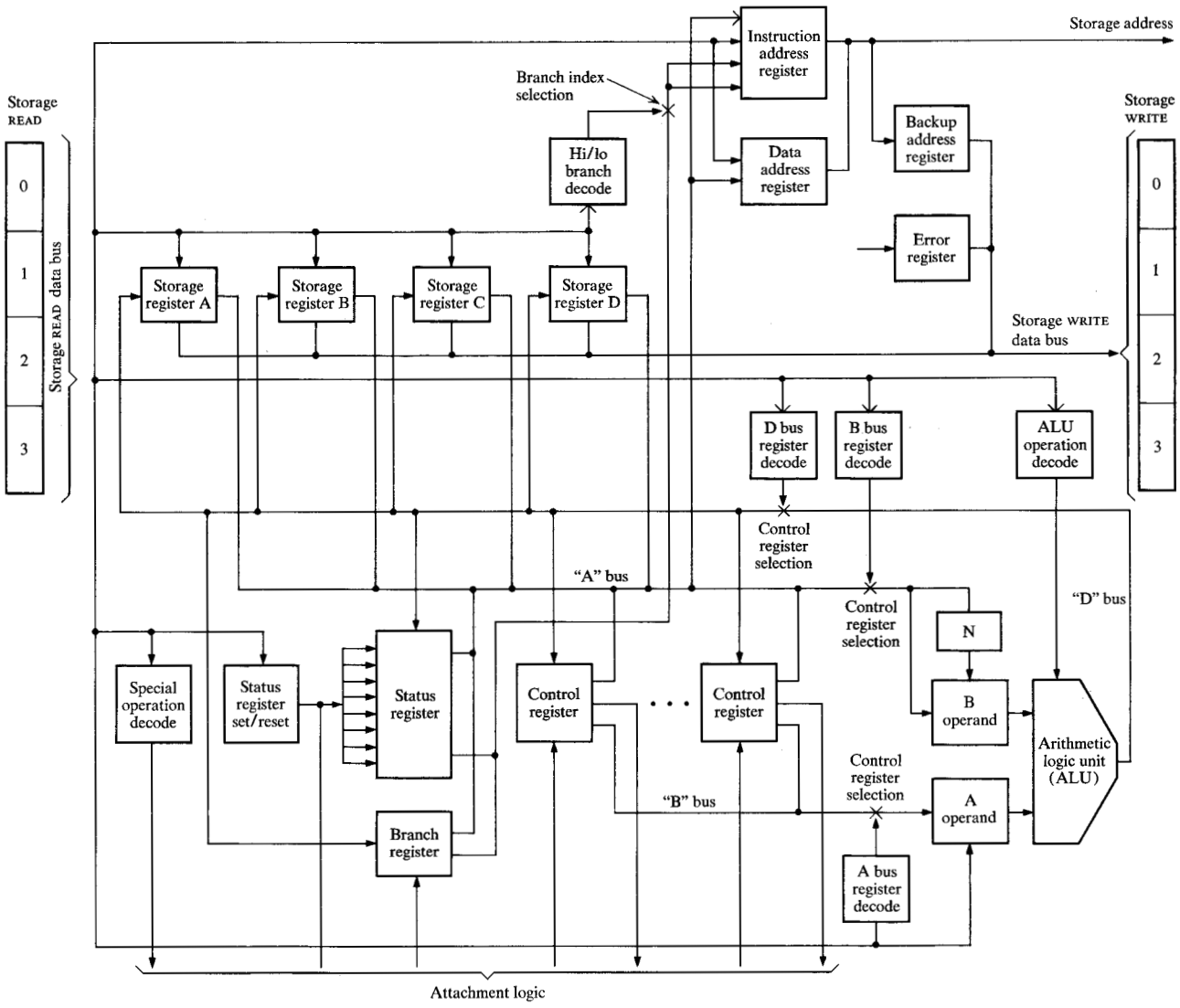


Figure 2 Processor data flow.

are entered on disks at the factory, providing very good control over the data. The disk is mailable and disposable, so that microprograms in the field can be updated easily and efficiently. The disk contains the functional microprogram, the inline and the stand-alone diagnostic programs. The interface to the control unit is relatively simple and is integrated with the processor. When the control store is loaded, the interface reads the first track into the control store. This track contains a bootstrap program that controls the loading of the remainder of the microprogram. The main considerations in selecting this disk drive over a tape casette were cost, random access capability and commonality with other IBM systems.

Previous disk storage control units were interfaced to the disk drives with control lines that provided only for a limited set of control commands and for a small amount of status information from the drive. The philosophy of providing extensive diagnostic information to maintenance personnel requires an expanded set of diagnostic commands to the drive and much more status information than was previously available. Such status information is especially critical for the complex drive servo system. These requirements are best met by designing an interface resembling a simplified channel between the control unit and the drives. By judicious specification of standard control signals and sequences, this channel was made common to both the 3830 and the 2835 control units, further enhancing the commonality between them.

The device interface includes an outgoing and an incoming parity-checked data bus eight bits wide, a tag bus

a half-byte wide, a module selected bus six bits wide, and deskewing, verification and error signaling lines. These buses are connected in parallel to all the drives.

Commands are encoded on the tag bus and associated data or control information is placed on the outgoing data bus to the drive. Under microprogram control, a tag-gate line is raised to indicate that the tag is valid. The drive acknowledges receipt of this information by putting a signal on a tag valid line.

Selection is accomplished by a combined physical and logical addressing scheme. Each disk drive is assigned a nonvariable physical address at installation time. A moveable address plug assigns to each drive a logical address that can be varied by the operator. To select a drive, the control unit places the logical address on the outgoing data bus, places a TRANSMIT MODULE ADDRESS command on the tag bus and raises a module select gate line in addition to the normal tag gate. The selected drive replies by placing a three-out-of-six encoded physical address on the module selected bus. The encoded physical address ensures that one, and only one, drive is selected and that all error status can be logged and related to a physical drive, independently of its logical address.

Interface attachment architecture

The individual characteristics of the 2305 and 3330 disk storage devices required that certain functions be implemented differently in each control unit. Among these are the read/write and serializer/deserializer (ser/des) functions, the channel interface and control logic, and correction of data errors read from the disk. Extensive error checking is again performed in each of these areas to provide information necessary to retry procedures and for maintenance analysis.

The linear read-write circuits and amplifiers are physically located in the disk device, as close to the read-write heads as possible to avoid noise effects. The timing circuits for write data and the data-separation circuits for read data, as well as the ser/des, are all physically located in the control unit, because the low-speed logic circuits used in the drive cannot deserialize data at the file data rate. In addition, the 3830 would require up to eight relatively expensive serializer-deserializer circuits if the circuits were placed in the drives.

The ser/des is frequency-stabilized and clocked by two oscillators. A phase-locked oscillator (PLO) is driven by reference pulses recorded on the disk in the 3330 or detected from the spindle in the 2305. This provides a clock frequency locked to the rotational velocity of the disk. A variable frequency oscillator (VFO) drives the read data separation circuitry and ser/des clock pulse generator. The VFO's used in the two facilities are identical. The VFO is synchronized either to the PLO for writing or for clocking across data fields and gaps, or to the read data stream for separating data. The separated data is then deserialized by means of conventional deserializing rings. This double-oscillator system provides reliable and consistent data recovery which allows disk pack interchangeability for the 3330 facility. The oscillators are phase checked during write operations to prevent any possibility of writing data incorrectly.

The channel interface selection logic and channel interface line drivers are common to both the 2835 and 3830. Switching between channels is accomplished in a similar manner on both machines. However, because of the great difference in data rates, the data transfer interface is quite different in the two machines. Data transfer between the ser/des registers and the channel interface can be controlled by the microprogram if the required microinstructions for each byte transfer can be executed within a single-byte recording interval. A typical data handling loop is shown in Fig. 3. The worst-case command is a SEARCH, where bytes from the channel must be compared with bytes from the ser/des in addition to the other housekeeping functions. A minimum of six microinstructions is required in the SEARCH command data handling loop. The 3330, with 1.24 μs between bytes, allows the 200-ns cycle processor to successfully transfer data without overruns. The 2305, with the capability of making 1.5 million transfers per second across a one- or two-byte parallel interface, allows fewer than 4 microinstructions per transfer, so the transfer must be done by a hardware-controlled buffer using hardware tag controls and a hardware transfer counter.

The transfer of data between the I/O channel and the disk drive via the microprogram presents problems of synchronization between the channel, the control unit processor, and the disk. To prevent overruns, the channel's data transfer rate must exceed that of the disk. To allow for synchronization within the transfer of a record, the data must be buffered and transfer must be controlled dynamically by either microprogram or channel, depending on which side is ready for a byte transfer.

The 3830 control unit has a unique double buffer interface to transfer data between the microprogram and the channel. One buffer is controlled by the Data-In/Data-Out interlocking data transfer tags, while the other is controlled by Service-In/Service-Out. Thus, one buffer is always available for a transfer to or from the channel, while the other may be loaded or unloaded by the microprogram. The microprogram controls data transfer if both buffers are available to it; otherwise the channel controls the transfers.

The ECC hardware is designed to correct burst errors that may occur in read data. Previous disk storage systems used cyclic code checks, principally to detect burst errors, but no error correction was provided. The error

correction capability in the present system enables the control unit to correct most read errors. Home address, count and key field errors are corrected by the control unit and are presented to the using system upon execution of a CHANNEL RETRY command, which is transparent to the user program. Data-field errors are corrected by the using system and the required error correction information is supplied upon execution of a RETRY command. In the case of the 3330 disk facility an attempt is made to correct "uncorrectable" errors by rereading the record while the recording head is offset from its nominal position on the track. The correcting power of the error correcting codes chosen provides for greater capability of detecting long-burst errors.

Reliability, availability, and serviceability

As the complexity of systems increases, the probability of failure of any subsystem or component increases. At the same time it becomes more and more difficult for maintenance personnel to diagnose the failure. Extended interruptions of system operation become more and more costly. Therefore, particular attention was given to reliability, availability, and serviceability in the basic design philosophy of the control units.

To provide a high degree of availability to the customer, the prevention of errors, recovery and repair facilitation is applied at several levels of the disk facility. A logic family with an order-of-magnitude improvement in reliability is used in the control units. Data errors occurring on the recording medium are corrected by means of ECC procedures. Eighty percent of the control unit circuitry is duplicated or else checked. Logic failures are therefore detected and isolated to a small area, and visual as well as software-recognizable indicators are provided. Test routines and analysis procedures assist to isolate failures either without the use of an oscilloscope or else by enhanced oscilloscope techniques. By distinguishing between primary and secondary errors and using I/O Alert and Command Retry features, most soft errors do not cause a using system interrupt. Rather, errors are logged and their indicators reset, permitting deferred maintenance at a time convenient to the customer. Dynamic error logging under operating programs is provided, as well as utility programs to summarize, edit and print error data for the operator or the customer engineer.

Disk drives that need repair or maintenance service can be isolated from other drives that remain in customer service. By placing the control unit in the "Inline" mode a drive with a service address plug inserted is operationally connected only to the control unit and appears unavailable to the channel. Diagnostic microprograms can now be run on the control unit, interleaved with normal functional microprograms running on the other drives, which continue to be available to the system.

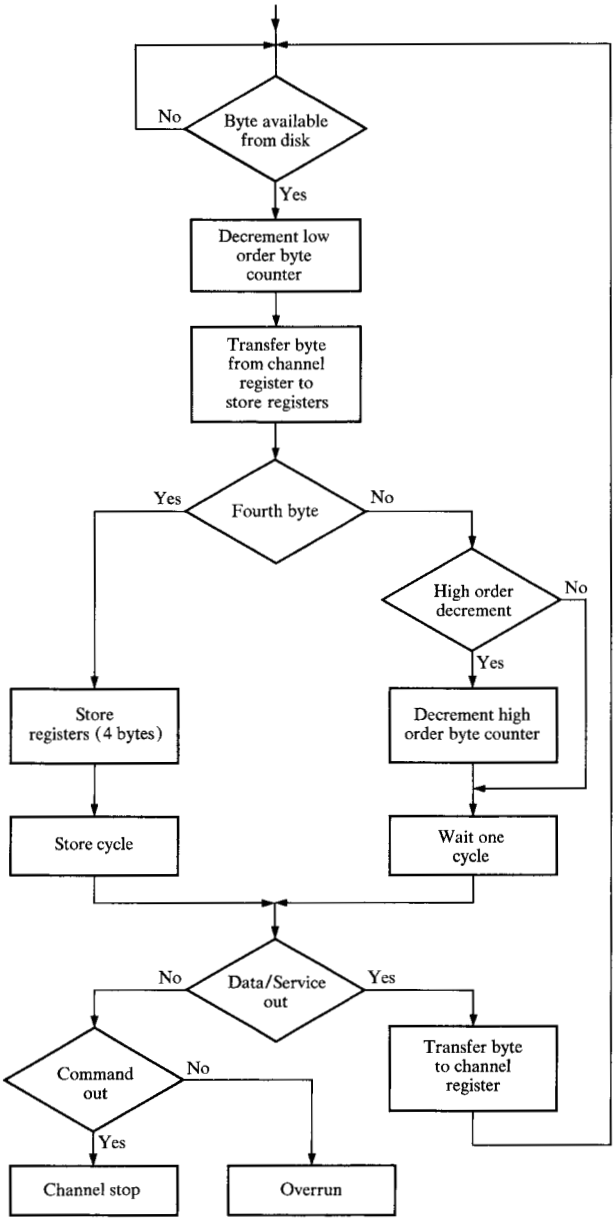


Figure 3 Read data byte transfer loop.

While microdiagnostics that can be run without affecting customer throughput have been used in previous control units, the writeable control store and the microprogram load file have given the customer engineer an unlimited supply of diagnostic microprograms that he can call up at will. This added flexibility has greatly simplified the problem of developing a Maintenance Analysis Procedure designed for the special requirements of both the control unit and the disk devices. The extensive checking hardware allows the microdiagnostics to isolate and identify hardware failures far more completely than has such hardware in the past.

The maintenance philosophy is a unique characteristic of the 2835 and 3830 control units. It is based on the Integrated Maintenance Plan, a collection of microprogram diagnostics (both online and stand-alone), system diagnostic programs, and various levels of maintenance documentation.

The principal guide to the use of these tools is the Maintenance Analysis Procedure (MAP). This procedure is described by a series of special flow charts and tables that define an analysis method for whatever combination of symptoms is found by a customer engineer. Symptoms range from an excessive number of soft errors to a complete power failure. The MAP indicates the microdiagnostics or system diagnostics that are to be executed, the results to be expected, and the cards or set of signal lines that should be changed and checked for each malfunction discovered. This method is expected to greatly increase the ease of servicing the facilities and, therefore, to decrease both the cost of servicing and the inconvenience to the customer.

Conclusion

By identifying similar functions, it was possible to design, for dissimilar devices, two control units that share much commonality. The key to commonality is the processor and its writeable control store, and the device interface. Reliability, availability and serviceability have been improved by providing checking circuitry and diagnostic microprograms which isolate and locate faults and also provide statistics for preventive maintenance. The I/O channel, which is one of the system resources, has been made more available to other control units by the concept of block multiplexing and by independent command retries by the control unit.

Acknowledgment

It must be emphasized that many people, with whom the authors worked as a team, contributed to the design and development of the 2835 and 3830 control units. In particular it is noteworthy that the architecture definition and development phase required close cooperation between hardware-oriented and microprogramming engineers. The contribution of many to this paper is gratefully acknowledged.

References

1. "Component Summary—2835 Storage Control and 2305 Fixed Head Storage," IBM Systems Reference Library Order No. GA26-1589, IBM Corporation, White Plains, New York, 1970.
2. "Component Summary—3830 Storage Control and 3330 Disk Storage," IBM Systems Reference Library Order No. GA26-1592, IBM Corporation, White Plains, New York, 1971.
3. "Introduction to IBM System/360 Direct Access Storage Devices and Organization Methods," IBM Systems Reference Library Order No. GA20-1649, IBM Corporation, White Plains, New York, 1969.
4. "IBM System/360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturers' Information," IBM System Reference Library Order No. GA22-6974, IBM Corporation, White Plains, New York, 1971.
5. A. Van De Goor and C. G. Bell, "A Control Unit for a DEC PDP-8 Computer and a Burroughs Disk," *IEEE Trans. Computers*, **C-18,** 1044–1048 (1969).
6. T. H. Bonn, "Mass Storage: A Broad Review," *Proc. IEEE*, **54,** 1861–1870, (1966).

Received May 7, 1971

G. R. Ahearn and Y. Dishon are located at the IBM Systems Development Division laboratory at San Jose, California 95114. R. N. Snively is currently on assignment at the IBM Systems Development Division laboratory at Boeblingen, Germany.