Evolution of the DASD storage control

by C. P. Grossman

This paper identifies the major requirements and design points for storage controls and describes how these requirements have been met over time. It also describes the interplay of the three critical components of a subsystem: hardware technology, microcode, and software.

Much of the improvement achieved in the performance, function, and availability of DASD storage subsystems has been made possible by the DASD storage control. Until the emergence of the IBM 3880 Models 13 and 23 cached storage controls, these improvements were transparent to the majority of large-system users. Cache storage made such a dramatic difference in DASD subsystem performance that user awareness of storage controls increased noticeably. Now, with the availability of the IBM 3990 Model 3 storage control, the capabilities of a storage control to enhance system performance and availability using fast write and dual copy have been extended even further.

Hardware design and technology have played a large role in the development of the DASD subsystem. However, as we review the characteristics of storage controls, the ever-increasing participation of microcode and operating system support becomes quite clear. Innovations in the hardware have been accompanied by corresponding enhancements to the microcode and the system's software to exploit the new capabilities. Because the storage control serves as an intermediary between the processor channel and the disk device itself, we also discuss significant changes to these components as they relate to the storage control.

From a user's perspective, the goals of a data storage system are good performance, data availability, easy access to data, and cost-effectiveness. All components of a computer system contribute to attaining these goals. When we focus attention on the role of the storage control within the system, many of its design considerations can be categorized into one or more of the following areas:

- Overlapped component operation at various levels: processor/channel, channel/device, storage control/device, device/device, etc. (The more various components can function in parallel, the more efficient overall system operation will be.)
- Efficient communication, which encompasses a number of items, including the speed of data transfer (or bandwidth), the number of communication paths among the hardware components, and the architecture of the channel commands
- Consistent and efficient techniques for addressing data and defining data structures in terms of both access time and storage utilization
- Accessibility and manageability in that data are available to the user when needed (Failures can be promptly repaired, and system changes made with minimal user impact.)

As we shall see, in the very early systems there was no separate storage control. The need for such a

component developed gradually, as system design addressed fundamental user needs. In its most elementary form, a storage control is a processor that off-loads host-processor workload by receiving I/O commands from the host system, translating these to orders to the DASD, and managing the transfer of data between the host processor and the DASD. Gradually, this outboard processor has assumed more responsibility for error detection and correction, subsystem problem diagnosis, more sophisticated host–device synchronization (such as rotational position sensing [RPS]), anticipation of host data requirements (cache), speed-matching slow channels and faster devices, and new functions such as dual copy. All of these items are discussed in greater detail at appropriate points in this paper.

I/O control in early IBM systems

The earliest IBM computers, such as the IBM 701, directly controlled the attached I/O devices; CPU hardware and programs performed all required functions for device control and data transfer for the attached card readers, punches, and printers.[1] The requirement for specialized hardware and software to control I/O devices was recognized early, and systems incorporating such capability evolved rapidly. Table 1 gives a chronological sequence of the development of IBM I/O control devices.

Precursor of the channel and early I/O control software. In 1957, the IBM 709 introduced the data synchronizer, a precursor of today's channel. The data synchronizer was a significant advance in that it allowed as many as six devices to access main-storage core buffers while the processor performed other work.[2] At this time, the first input/output control system (IOCS) software was introduced.[3] The parallelism made possible by the data synchronizer encouraged standardization of channel programs and CPU-channel synchronization through the IOCS. The 709 used instruction loops to test for completion of I/O operations by the data synchronizer. The program interrupt, introduced later with the IBM 7090 family of processors in 1958, handled overlapped I/O operations even more efficiently by eliminating these instruction loops.[3] A program interrupt occurred whenever an I/O operation completed or an unusual condition occurred, and the processor automatically branched to special routines to process the interrupt.

The first IBM I/O control units. At the same time the processors were improved to process I/O more efficiently, the devices themselves were also rapidly changing. The increasing data requirements of applications demanded multiple tape drives on a system. The IBM 702 introduced control units for its I/O devices, which provided a standard interface to the processor. These control units also provided electrical power and diagnostic capabilities. This meant that a specific device or devices and associated control unit could be removed from the system to run independent diagnostics or other operations. Many installations took advantage of this capability to perform card-to-tape, tape-to-card, and tape-to-printer operations. This was an early example of the continuing effort to off-load I/O functions from the processing unit.[4,5]

Table 1 Development of I/O control in IBM devices

Year	Device	Key Characteristic
1952	701	Processor directly controls I/O devices
1953	702	First I/O control unit
1956	305	RAMAC—first IBM DASD
1957	709	Data synchronizer
1958	7090	Program interrupt
1962	7631	First IBM disk control unit
1964	System/360	New processor, channel, and I/O device architecture
	2841	First IBM microcode-controlled disk storage control unit
1971	3830	Extensive use of microcode in the storage control
1979	3880	Improved performance, reliability, availability, and serviceability
1987	3990	Dual copy and DASD fast write

The need for better performance and economic considerations led to the development of the first tape control unit in 1956. This unit provided for overlapped I/O operations and processor execution, and some of its features were later incorporated into the IBM 709 data synchronizer.[6] The tape control unit also allowed system designers to concentrate expensive control circuitry inside the control unit, which all devices shared as required.

The first IBM DASD. The IBM 305 RAMAC was the original IBM DASD and was introduced in 1956. Like other I/O devices of this period, the disk was initially under direct CPU control for all operations. I/O programming became more complicated because the new device had characteristics that required different approaches. For example, the single read/write head had to be positioned to the correct disk surface and moved to the correct track. This motion also intro-

duced new considerations for performance. DASD required new record formats.[6] Because of its high cost, the primary uses were for highly active data with rapid response requirements, and this prompted a great deal of research leading to sophisticated techniques for record searching. Unfortunately, all of the coding had to be devised for each application and each device type.[7]

For the first time, high-level macros were provided for processing sequential files on disk.

The first IBM DASD control unit

In the early 1960s, hardware and software combined to provide partial solutions to the response and record searching problems just described. The IBM 7631, the first disk control unit, worked with the IOCS to improve performance by allowing overlapped seek operations.[6] The 7631 used a DASD track to describe the format of the data on a cylinder, thereby allowing each cylinder of data on the device to have a different record format. The most significant software advances, however, were in the area of standardized data access routines, buffer management, and automatic blocking and deblocking of records. For the first time, high-level macros were provided for processing sequential files on disk. The improved buffer management assumed responsibility for synchronizing I/O from multiple devices. Automatic blocking and deblocking relieved the application programmer from having to work with physical tracks. Random-access applications, however, typically required sophisticated hashing algorithms and techniques to calculate the physical track location of data.[7]

System/360: New architecture and solutions

A true revolution in DASD management and use took place with the introduction of System/360 in 1964. Growing user demand for faster response led to requirements for multitasking, data access, and record formats, which could not be met with the current systems.[8]

At the time of System/360 development, the need for performance as well as reliability, availability, and serviceability (RAS), and for data and space management were known. Some preliminary steps had already been taken to resolve these needs, and the architecture of System/360 provided further improvements.[8,9] DASD hardware and software design efforts from 1964 to the present still focus on this same set of requirements.

The rest of this paper discusses ways in which System/360 initially addressed these areas and the continued provisions of solutions by subsequent storage control units and related software. Items to be discussed include the following:

- Technological improvements in performance, reliability, and new functions
- Data access—channels and connectivity
- Disconnected operations
- Configuring for availability
- Migration ease
- Failing component isolation (fencing)
- Serviceability—microcode diskettes, concurrent maintenance, status information, diagnostics, fault-tolerant operation, error detection and fault isolation (EDFI), and configuration definition
- Channel command retry, selective reset, and error-correction code
- Hardware identification
- Channel command architecture
- DASD performance improvements
- Software interactions

System/360 innovations. System/360 laid the foundations upon which all the subsequent improvements in DASD subsystems hardware and supporting software rest. Therefore, it is important to understand the scope of changes it introduced, before we discuss later improvements.

The significance of System/360 I/O operations lies in the definition of an encompassing architecture that accommodated the following:[8–11]

- Consistency of I/O access methods across device types
- Standardized record structures (count key data)
- Standardized file structures

- Buffering in Data Management for some access methods
- Data management and allocation
- Channel architecture
- Microcode-controlled disk storage control unit
- New DASD
- Standard I/O interface—Original Equipment Manufacturers' Information (OEMI)

System/360 introduced the following new set of access methods:

- Sequential Access Method (SAM)
- Basic Direct Access Method (BDAM)
- Indexed Sequential Access Method (ISAM)
- Basic Partitioned Access Method (BPAM)

SAM and BDAM were extensions of existing access methods and ISAM and BPAM were new to System/360. The latter two for the first time freed the programmer from the responsibility of translating a logical record identifier into the physical address (cylinder, head, record) on disk.[6,12]

Until System/360, allocation of disk space was manually controlled, so that it was possible for one program to write over data belonging to another application. System/360 introduced direct-access device storage management (DADSM), which centralized the management of DASD space allocation. The key component of DADSM was the volume table of contents (VTOC) contained on the volume, which recorded information about the used and free space on the volume. Another important element was the set-file-mask capability. This limited the scope of operation of a channel program to a particular extent on the DASD and provided further protection of data.

The one architectural component of System/360 that probably gave the biggest throughput boost was multiprogramming. This made it possible for one program to execute while others waited for I/O operations to complete. It insulated total system performance from the seek and rotational delays inherent in DASD access and considerably improved system resource utilization.[6]

The 2841 storage control. The IBM 2841 storage control unit, which appeared with the System/360, was IBM's first microcode-controlled storage control unit.[6] It was central to the fulfillment of the mandate of the designers of System/360 to provide a device-independent interface between the processor and the DASD. To do that, microcode translated the general channel command words of the System/360 architecture into device-specific commands necessary to access data. System/360 introduced a new disk record format that is known as *count key data* (CKD), which is still in use today and has only recently been enhanced by the new Extended Count Key Data (ECKD™) architecture.[13,14] Characteristics of count key data format and Extended Count Key Data architecture are summarized in Table 2. Microcode allowed for the implementation of this structure, without separate hardware components for each device type.[6]

Table 2 Characteristics of count key data format and Extended Count Key Data (ECKD) architecture

Count key data provides for:

1. Self-description of all tracks
   - Track identifier (home address)
   - Track descriptor (record 0)
2. Self-description of all data blocks (count field)
   - Address
   - Key length
   - Data length
3. Key field (optional) used to locate records quickly using the search channel commands
4. Data block, which is also called a *record* in hardware terminology
5. Set of instructions, called *channel commands* for accessing data on the disk

In addition, the ECKD architecture provides an extended set of channel commands for prenotification and optimization of storage control and DASD operations.

Microcode has been essential to storage-control design for a number of reasons. It is easier and faster to modify microcode both during and after the development cycle than it is to change hardware logic modules or circuits. It also introduces a great potential for flexibility in introducing or refining storage control capabilities.

Two types of delay are inherent in any DASD access: (1) *seeking*, which is moving the read-write mechanism to the correct track; and (2) *latency*, which is waiting for the correct record to be positioned under the head for access.

Earlier systems had begun to compensate for these delays through buffering, the data synchronizer, or the 7631 disk control unit to allow some degree of host-device activity overlap. In these systems, the data path from the device to the data synchronizer was kept busy during the entire operation.[6] The combination of the IBM 2841 control unit and the System/360 channel allowed the device to free the

control unit and the channel while it completed a seek operation. After the actuator was properly positioned, the device signaled the control unit that it was ready, and the control unit asked for the next channel command word (CCW) from the channel.[15] This capability was significant for two reasons: it freed valuable channel and control unit resources to service other requests, and it allowed up to eight devices independently to seek requested tracks.

Evolution of the DASD storage control unit

The development of the IBM 2841 control unit was followed by three succeeding families of IBM DASD storage controls: the IBM 3830, the IBM 3880, and the IBM 3990. Each of these families has introduced significant improvements in performance, availability, function, and versatility, when compared with its predecessors. In the case of the 3830 and the 3880, the units have been extended to new environments beyond their initial capabilities, through new microprogramming and modest hardware changes within the family architecture. As we shall see, the microcode has become more important in providing new functions. In the following sections, we trace the development of major capabilities through each of these storage control families. The roots of the IBM cache and extended storage control function are illustrated in Figure 1.

The IBM 3830 control unit. The 3830 represented a substantial advance in storage subsystem design, exhibiting many examples of the power of combining storage control hardware and microcode, processor architecture, and system control programs. Major examples include rotational position sensing, channel command retry, selective reset, microcode error-recovery routines, read-only microcode diskette, presenting sense information to the host for logging (LOGREC), and microcode diagnostic routines.

The IBM 3880 storage control. In 1979, IBM introduced the 3880 family of storage controls.[6] Announced initially to support DASD in intermediate systems, the capabilities of the 3880 were rapidly expanded to provide new options and functions for all IBM systems, from intermediate to very large, evolving from the support of a simplified record format for smaller DASD to a high-performance cache storage control for the more demanding on-line systems. The new capabilities included the following:

- Fixed-block architecture
- Improved reliability, availability, and serviceability
- Improved connectivity and channel switching
- Increased channel speeds
- Device-level selection (DLS) and two storage directors, each of which was the functional equivalent of a 3830
- Speed-matching buffer and new channel commands for such functions as multitrack operations
- Channel command stacking
- Cache

The IBM 3990 storage control unit. The hardware design and microcode of the 3990 family provide more capabilities than earlier IBM storage controls and include the following:

- Four-path access to DASD, including device-level selection enhanced (DLSE) and nondisruptive DASD installation capability
- Improved performance
- Concurrent maintenance
- More extensive diagnostics
- Service information message (SIM), error detection and fault isolation (EDFI), and error logging
- Remote support and microcode patch application
- Vital product data (VPD), including microcode-prompted description of the configuration

With the exception of DLSE, nondisruptive DASD installation capability, and concurrent maintenance, the above functions are common to all 3990 models. The 3990 Model 1 has only one storage cluster and so cannot operate in the DLSE mode, provide for nondisruptive DASD installation, or allow concurrent maintenance.

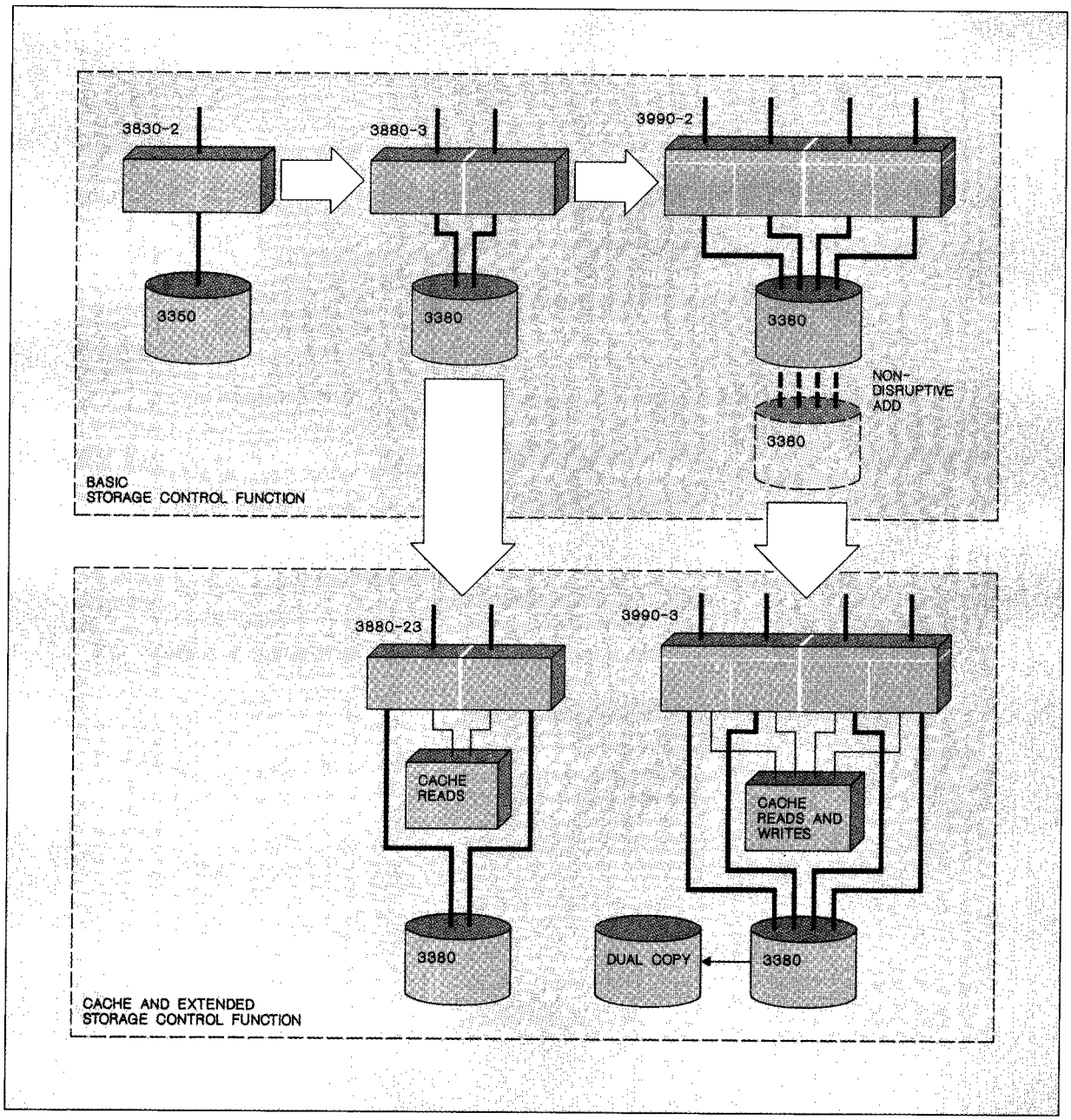
The additional features of a 3990 Model 3 cache subsystem include the following:

- DASD fast write, which extends cache read-hit performance to write operations
- Dual copy for maintaining two logically identical copies of key volumes
- Cache fast write, i.e., specialized fast writes for temporary data
- Larger cache sizes up to 256 megabytes
- Improved subsystem resource management

Technological improvements for performance, reliability, and new function

The 3880. The IBM 3880 storage control was the first IBM storage control to use a single large-scale integration (LSI) chip microprocessor,[16] which eliminated many connections inside the storage control. Two

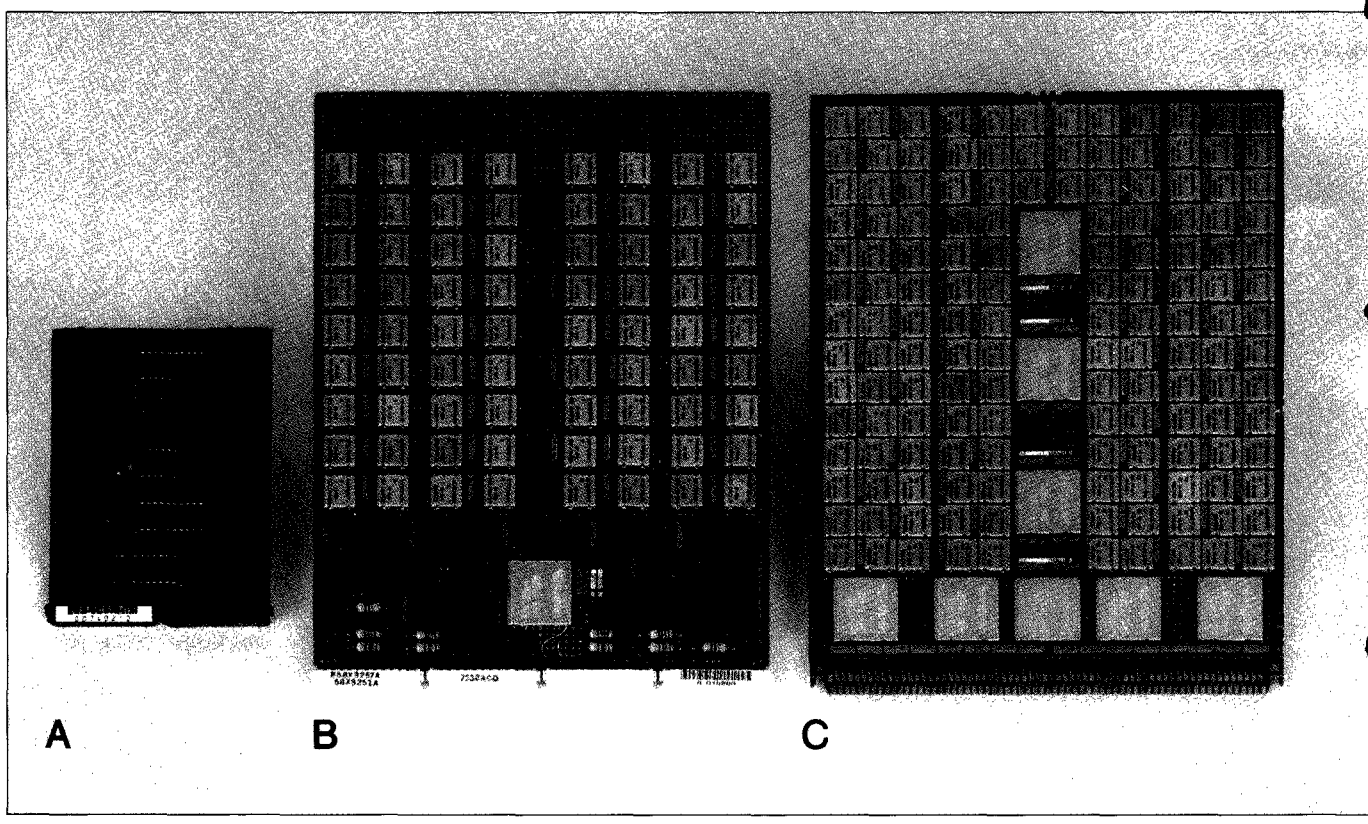
Figure 1 IBM storage control development



consequences of this are faster internal processing and improved RAS. The new design of the storage control also incorporated substantially more control storage—the memory inside the storage control that holds the microcode and its control blocks. These are the major hardware characteristics that made the many innovations of the 3880 family possible.

Improved packaging of the 3880 circuitry allowed for two storage directors—each the functional equiv-

Figure 2 (A) 3880 Model 13, 64-kilobyte card; (B) 3880 Model 23, 4-megabyte card; (C) 3990 Model 3, 16-megabyte card



alent of a 3830—housed in a unit roughly the same size as a 3830. Aside from the obvious savings in floor space, this has a significant benefit for problem determination in that the two storage directors are interconnected so that when one storage director fails it can send error data to the functioning storage director. The active storage director, in turn, sends the data to the host system notifying it of the failure and providing sense information necessary to diagnose the failure.[17]

The 3990. The design of the IBM 3990 takes advantage of the DASD subsystem enhancements developed through the years and combines these in a new storage control architecture having innovations of its own. New high-density chip technology and new packaging techniques permit four-path access to devices and totally redundant storage path components. High-density memory chips allow for a much larger capacity control storage and larger cache sizes, and static, random-access memory (SRAM) technology chips allow for nonvolatile storage (NVS).

The microprocessor chip used in the 3990 is IBM's newest design and has approximately 7000 circuits. This powerful new microprocessor facilitates the 3990 Model 3's dual data-transfer capability, whereby up to eight operations can take place simultaneously inside the storage control (two for each of the four storage paths included in the Model 3), and other algorithm improvements for both data transfer and cache management that are discussed later in this paper.

New packaging techniques, including zero-insertion-force (ZIF) connectors, led to housing the storage-control function in a much smaller unit. In the 3880, the circuitry for two storage paths with up to eight-channel switching occupied a gate of about six cubic feet. In the 3990, two storage paths with eight-channel switching are housed in a storage cluster less than three cubic feet in size. Being able to put four storage paths in a machine the same size as a two-storage path 3880, facilitated the implementation of four-path DASD operations.

The shared control array contains information about the current configuration and status of all the DASD attached to the 3990. Much of this information was formerly contained in the dynamic path selection (DPS) array of the 3380 A-unit. Moving this information into the storage control improved performance because the storage control no longer has to interrogate the arrays in the head of string before beginning an operation.

The 3990 can accommodate much larger cache sizes, i.e., up to 256 megabytes, using new packaging techniques. Figure 2 shows the difference in storage density on each card type. The 3880 Model 13 cards held 64 kilobytes, using 16-kilobit chips, whereas the 3880 Model 23 cards held 4 megabytes of storage, using 1-megabit chips. The 3990 Model 3 card contains 16 megabytes of storage, using the same 1-megabit chip. One factor allowing the higher density is the use of the ZIF connectors. These connectors allow the tabs on the card to be much closer together, as seen in the figure.

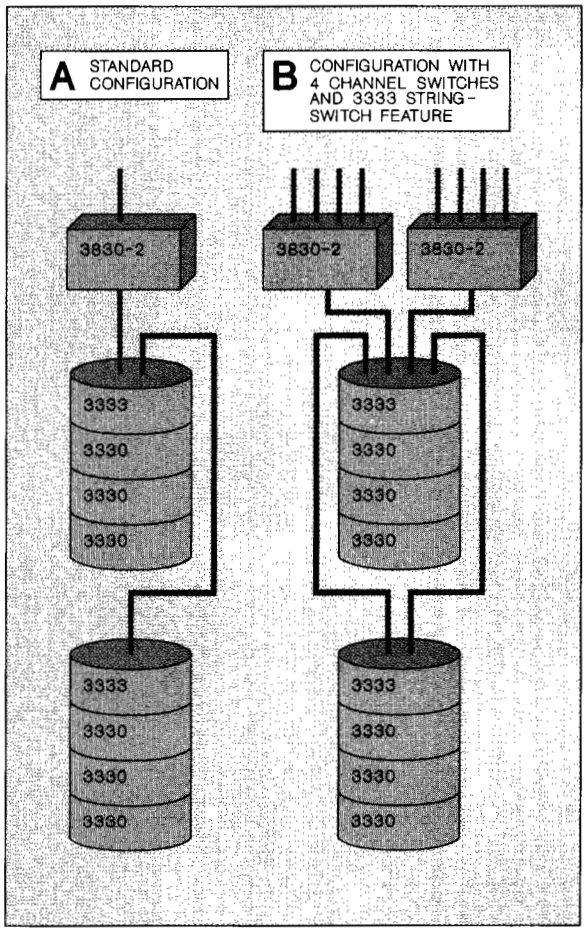
Nonvolatile storage for dual copy and DASD fast write uses static RAM chips and a battery to protect storage for essential subsystem data in case of a power failure. SRAM chips are used because they require less power drain on a battery. The battery can maintain the data in the NVS for as long as two days.

Component communication

Since the introduction of the 2841, the 3830, and the 3880, an increase in the complexity of data-processing installations has taken place. Typical 3880 and 3990 installations have multiple processors, most of them sharing access to the DASD subsystem and a large number of storage devices. This has had an impact on the design of storage controls, because channel connectivity—the capability of attaching to multiple processing units—becomes much more important. More data must be shared and must be shared among more processors. Also, availability requirements make multiple connections essential between a DASD subsystem and any one processor.

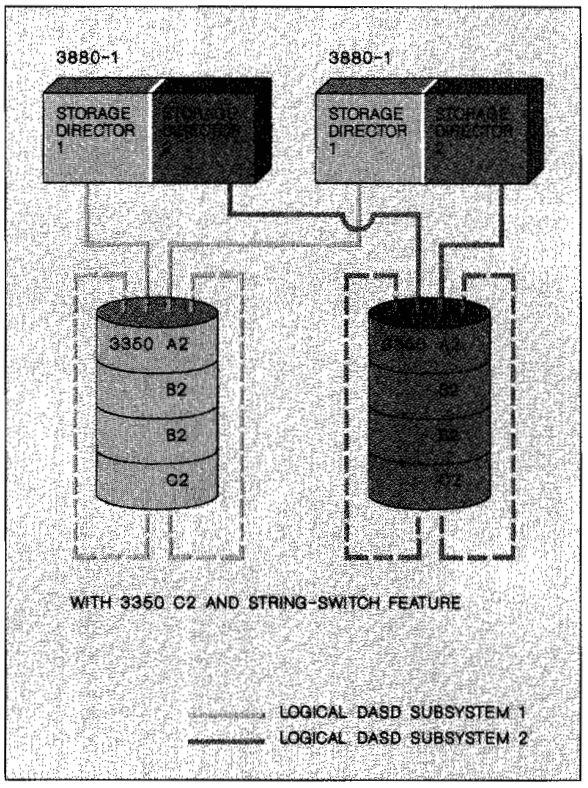
Connectivity. Connectivity is a key component of the availability and performance characteristics of a DASD subsystem. (In this paper, connectivity encompasses the channel-to-storage director connections—also called the upper interface—and the storage director-to-device [DDC]—also called the lower interface.) The number of connections, the capability of dynamically selecting those connections for reconnect operations, and the number of connections that can be in use at one time directly affect the performance of a DASD subsystem.

Figure 3 3830 storage control: Alternate path capability



The 3830 in a standard configuration provided for one channel attachment and a single connection to one or two strings of DASD. (See Figure 3A.) Connectivity of a 3830 DASD subsystem could be improved by adding up to four channel switches per storage control. By means of the string-switch feature of the 3333 DASD unit, a second 3830 could be attached to a 3333. This allowed for an alternate control unit path to the DASD. An added advantage of such a configuration was an increase in the channel access capability of the DASD from four to eight channels, as shown in Figure 3B.

Figure 4 3880 storage control: Alternate path improvements



The 3880 addressed the connectivity requirements by extending the four-channel switch capability of the 3830 to eight channels. Full use of this capability required two 3880s, cross-configured as shown in Figures 4 and 5. When the 3880 was so configured, a string of DASD could be accessed by as many as 16 channels, a two-fold improvement over the 3830 capability.

Another form of connectivity on the lower interface also had RAS implications. Devices, such as the 3330, which attached to both 3830s and 3880s, used one DASD controller (the 3333). This controller could communicate with exactly one storage director (or 3830) at a time. For availability, the controller could be string-switched to a second storage director (or 3830), which allowed access to the DASD, if one of the storage directors or 3830s failed. Because there was only one controller, there was no performance advantage. A failure of the disk controller prevented access to any of the attached DASD. The 3350 DASD, while still having a single active disk controller, provided relief for this situation by allowing for an alternate disk controller—the C2 shown in Figure 4—to be manually switched into the configuration in case of a failure of the A2 disk controller.

The 3380 DASD, in conjunction with the 3880, provided substantial improvement in availability by incorporating two DASD controllers in the 3380 head-of-string, or A-unit. Not only were there two controllers, but they could also function concurrently and be dynamically selected for use with two devices inside the 3380 string. This provided a measurable improvement in availability, compared to earlier IBM DASD. As an added benefit, it also provided several performance advantages that are described next. (See Figure 5.)

3350 and earlier devices had a single data-transfer path. The operating system defined the total data-transfer path from channel to storage control or director to head-of-string to device, for the I/O operation to use. Furthermore, the I/O operation had to return along the same path. (See Figure 6A.) As system workloads increased, it became more likely that some component of this I/O path would be busy

Figure 5 3880 storage control and 3380 DASD: Alternate path improvements

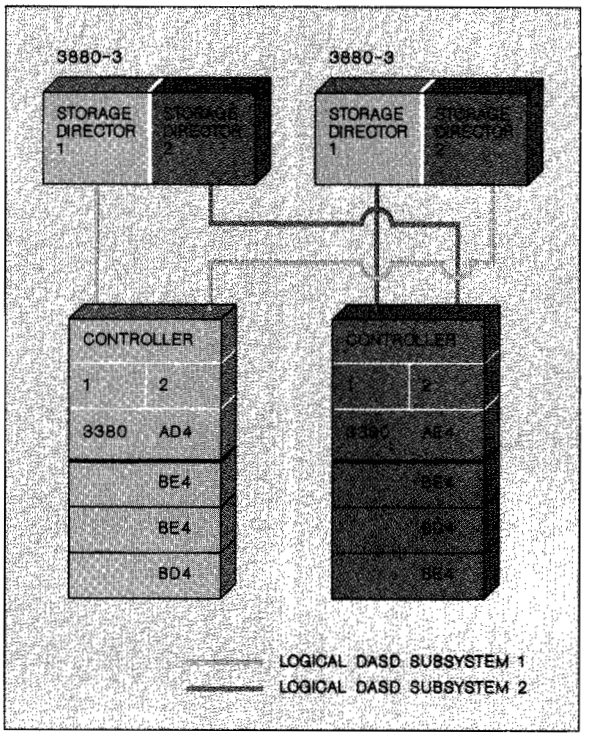
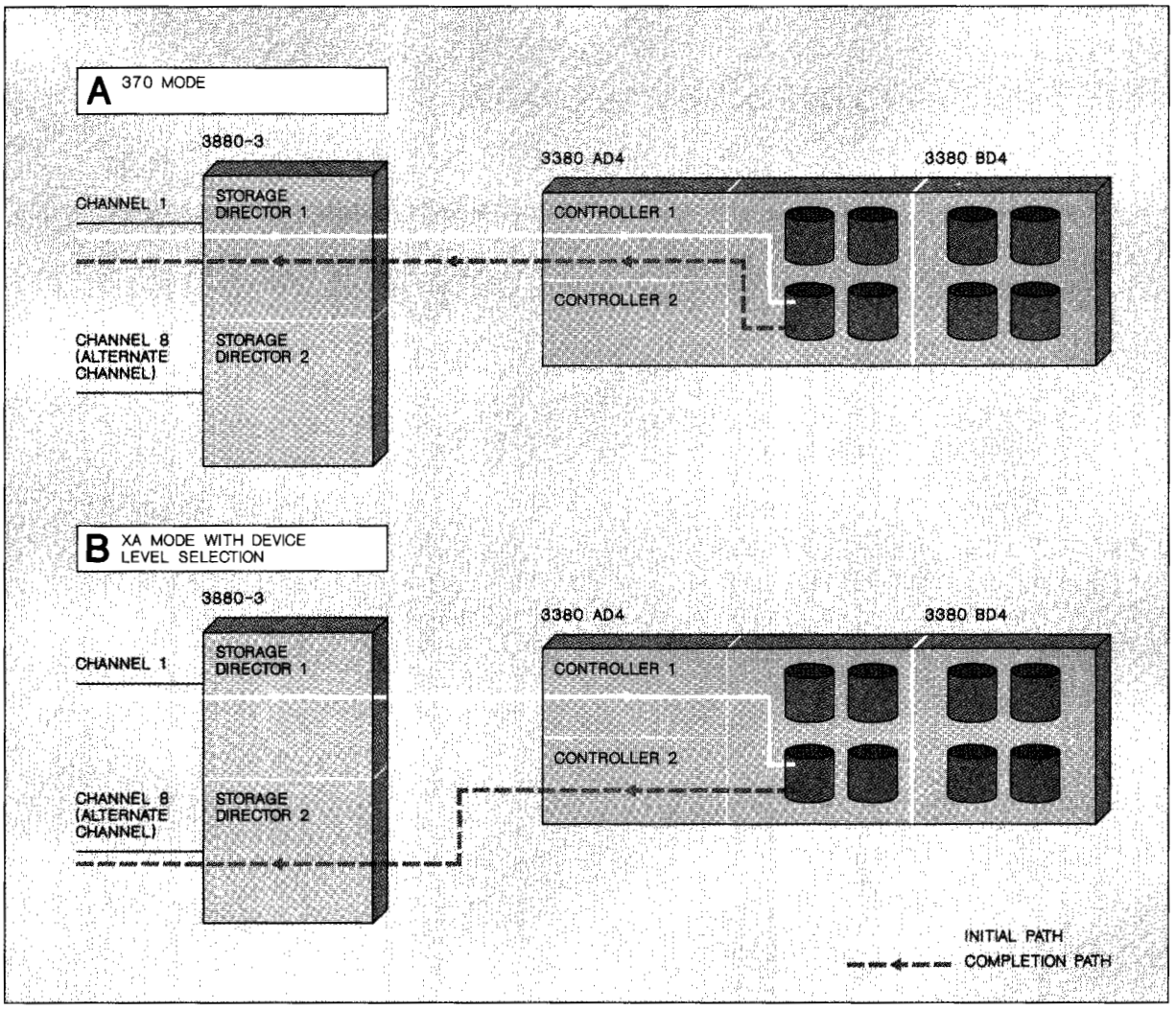


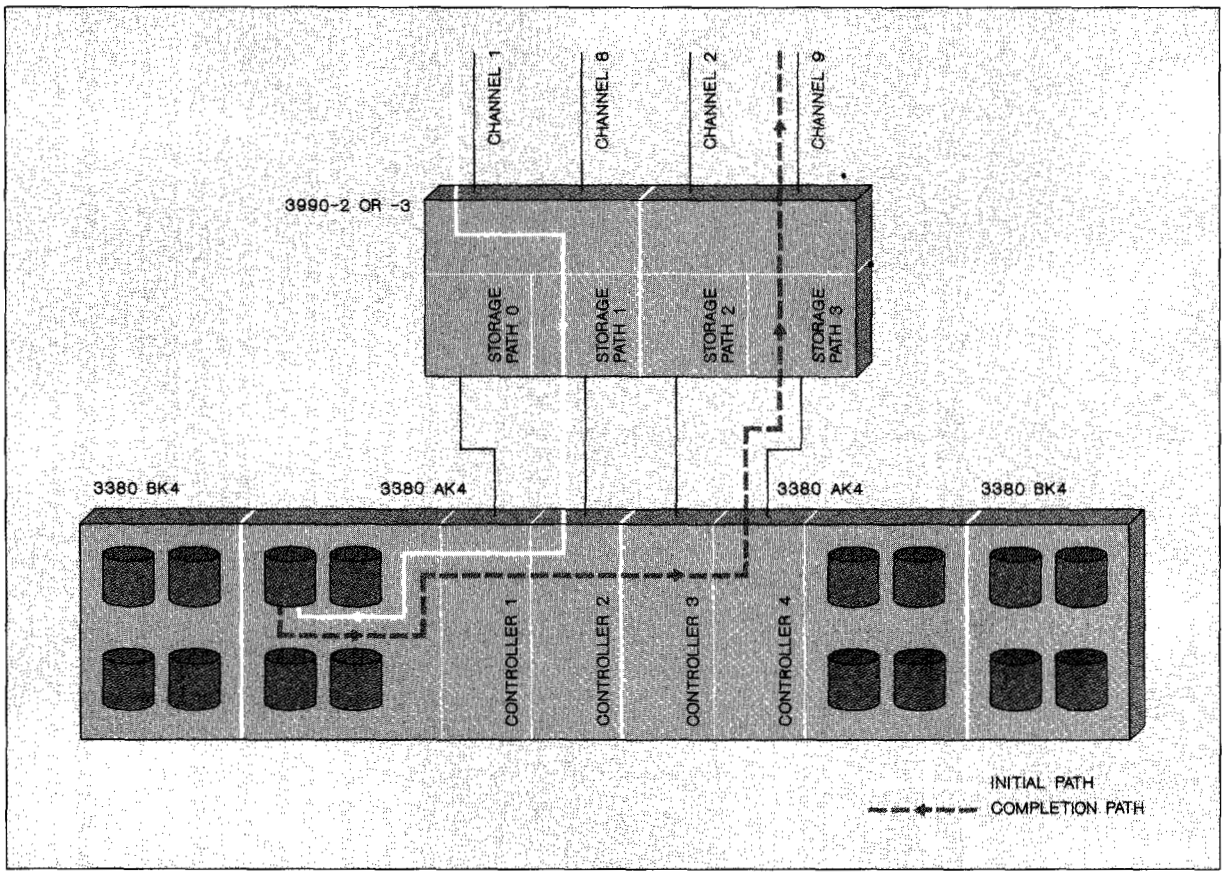
Figure 6 Path reconnect development in the 3880



with another I/O operation, blocking the completion of the first I/O due to an RPS miss. (This is discussed in the section on overlapped component operation—disconnected operations, later in this paper.) Queueing-theory studies have shown that as the use of a path increases, the response time of a device increases even more rapidly.[18] This became a serious limitation on the performance of the DASD subsystem. The 3880 and 3380 DASD, in conjunction with MVS/XA™ (extended architecture) and new processor channel capabilities, relieved this performance bottleneck with device level selection (DLS). (See Figure 6B.) In DLS mode, an I/O operation can return along either available path between the host processor and the DASD. An I/O operation could return from a 3380 device along the first available path, greatly increasing the probability of avoiding an RPS miss. Also, the presence of two DASD controllers in the A-unit meant that two I/O operations for different devices could be processed concurrently.

When used in an MVS/XA or MVS/ESA™ environment with the 3380 DASD Models J and K, the 3990 Models 2 and 3 offer connectivity improvements. (See Figure 7.) The most significant performance enhancement available to both the noncached 3990 Model 2 and the cached 3990 Model 3 is device level selection enhanced (DLSE). DLSE extends the two-

Figure 7 Path reconnect development in the 3990 storage control and 3380 J/K DASD



path DLS capability of a 3880 to four paths when the 3990 is used with four-path DASD. Used with the dynamic-reconnect capability of MVS/XA or MVS/ESA, DLSE dynamically selects whichever of the four paths is free for completion of an I/O operation that is disconnected from the channel. Allowing an I/O operation to complete along any of four independent paths almost eliminates RPS misses and increases the possible utilization of the channel. Because of the four-path capability, installations can schedule volume dumps more easily: the performance impact on a DASD subsystem of running a volume dump is less severe in a four-path environment. Moreover, availability is improved because the loss of one path has less impact when there are three paths remaining than when only one path remains, as in a 3880.

The 3990 can have up to eight channel interfaces on each cluster. This is similar to channel switching in the 3880. Inside a single DASD subsystem, DASD can be attached to as many as 16 channels, with up to four channels from any one processor. When the 3990 is used with four-path 3380 DASD, up to four of these channel connections can be active at one time, which is twice the capability of a 3880.

Channel speeds. The 3880 family is capable of higher data-transfer rates than earlier control units: from the 886 thousand bytes per second of the 3330 to the 3 million bytes per second of the 3380 when attached to the 3 million-bytes-per-second data-streaming channels (via the 3880).

Both the 3880 Model 23 and the 3990 Model 3 cache allow data transfer out of the cache to a 4.5 million-bytes-per-second channel at that speed. I/O operations that access the DASD operate at device speed for data transfer. Because cache and channel operations are independent of physical device characteristics, they can take place at the higher channel speed,

thus improving subsystem performance by as much as 7 percent.[19,20] The improvement will typically be greater in a 3990 Model 3 using DASD fast write, because the faster transfer rate will apply to write operations into cache as well as read operations out of the cache.

The channel speed is set by the IBM customer engineer in the storage control. Because the speed is set for each channel interface, the storage control transmits data at the correct speed for each of the attached channels.

Channel command architecture. The 3830 continued to use the same CKD channel command set introduced with the 2841. The first member of the 3880 family—the 3880 Model 1—could be used with either the standard CKD format introduced with System/360 or in a new fixed-block architecture (FBA) format. FBA is intended for intermediate systems using the 3310 and 3370 drives. It uses a fixed, 512-byte record size with a new, simplified set of channel commands. Several of these new channel commands provide early information to the storage control about the nature of the channel program to follow.[21] While these are significant for efficient operation of the FBA DASD, they take on even more significance with later machines that use them in the speed-matching buffer, cache, and the ECKD architecture described in more detail later in this paper.

The 3380 DASD operates at a data transfer rate of three million bytes per second between the device and the storage control. Many installations require the sharing of 3380 devices between processors with channels capable of transferring data at this speed and systems with slower channels. This requirement has been met by implementing a speed-matching buffer in the hardware, microcode, and system software. The significance of this feature extends beyond its capability to match a slow channel to a fast device. This speed-matching capability was the next step toward the development of the new ECKD channel command architecture. The speed-matching buffer provided a microcode-controlled hardware buffer in the 3880. On a write operation, data accumulated in the buffer until the connection with the device was made. The device connection was delayed long enough for the channel transfer to finish at the same time the device write operation was completed. On a read operation, the opposite took place. The device began writing to the buffer at the same time the channel began reading out of the buffer. The device finished filling the buffer before the channel could complete reading the data out of the buffer. To optimize the performance of this buffer, the host-system software used the same LOCATE RECORD channel command used in FBA to help the storage control prepare itself for subsequent I/O operations. As we shall see, this concept of prenotification was subsequently enlarged into the ECKD architecture.

Both the cached storage controls and the speed-matching buffer make extensive use of the LOCATE RECORD and DEFINE EXTENT channel commands. These commands are used by the software to improve effectiveness and performance by enabling the storage control to anticipate the requirements of the channel program and respond appropriately.

The 3990 family uses the ECKD architecture. The LOCATE RECORD and DEFINE EXTENT commands are an integral part of the ECKD architecture. For compatibility with older applications, the 3990 accepts CKD channel programs and emulates their operation. As in the case of the 3880, this new architecture with its prenotification capability allows the storage control to anticipate future operations and function more efficiently.

The READ TRACK command, which is available on all models of the 3990, is used for full-track read applications, such as the IBM program products DFSORT and Data Facility Data Set Services (DFDSS). When READ TRACK is used, the storage control sends a pseudo count area. That is, eight bytes are sent with a value of X'FF' after the last record on the track, which eliminates the need for the host processor to clear the rest of the I/O buffer. The result can be significant savings in processor cycles for applications that read massive amounts of data in full-track mode.

Overlapped component operation: Disconnected operations

The 3830 control unit, in conjunction with the 3330 DASD and the block multiplexer channel, provides rotational position sensing (RPS), which allows the control unit to disconnect from the channel while certain DASD angular orientation operations complete. This results in higher channel throughput, because the channel can service other devices while positioning completes. Success of RPS depends on the fact that, in many cases, the control unit either knows or can predict where a particular record is on a track. A track is divided into a fixed number of angular sectors.[13] The control unit instructs the de-

vice to position to a specific sector. When this sector approaches the head, the device notifies the control unit. The control unit, in turn, reconnects to the channel, and the I/O operation is completed as the desired record passes under the head. Because the channel is not needed until the record reaches the head, the channel can service more I/O requests than it could with earlier DASD subsystems that did not have RPS. Here again, performance improvement is achieved by the coordinated design of storage control, DASD, processor channel, and system control program (SCP) software. This capability required the new System/370 block multiplexer channel as well as new software in the I/O supervisor (IOS) and the access methods to handle the sector operations.[22]

Dual copy is a new option that protects key DASD data from device failures.

To optimize channel use, storage controls (3830 and later) have been designed to disconnect from the channel while the storage control directs the physical positioning of the DASD for an I/O operation. With the 3830, this occurs on a seek operation. In the 3880, channel command stacking eliminates the need for a reconnect after a seek operation. When the 3880 receives a SEEK command, it saves the address for the seek but does nothing to move the device heads until the next command—typically a SET SECTOR—is received. Then the SEEK and the SET SECTOR commands are executed in parallel.[21] Performance benefits in two ways: (1) The two operations are overlapped rather than executed sequentially; and (2) a reconnect sequence is eliminated, thereby also eliminating the possibility of delay caused by some other device occupying the channel.

Data accessibility and manageability

Each family of storage controls has provided additional capability for enhanced availability, manageability, and serviceability of the DASD subsystem. Configuration options have been expanded to allow more redundancy. Storage controls have allowed attachment of both current and previous generation DASD for ease of migration. The 3990 has the capability for allowing installation of additional DASD units without disrupting system access to already installed units. Many changes have been made to allow faster repair actions, and in a number of cases the storage control can retry operations that had an error.

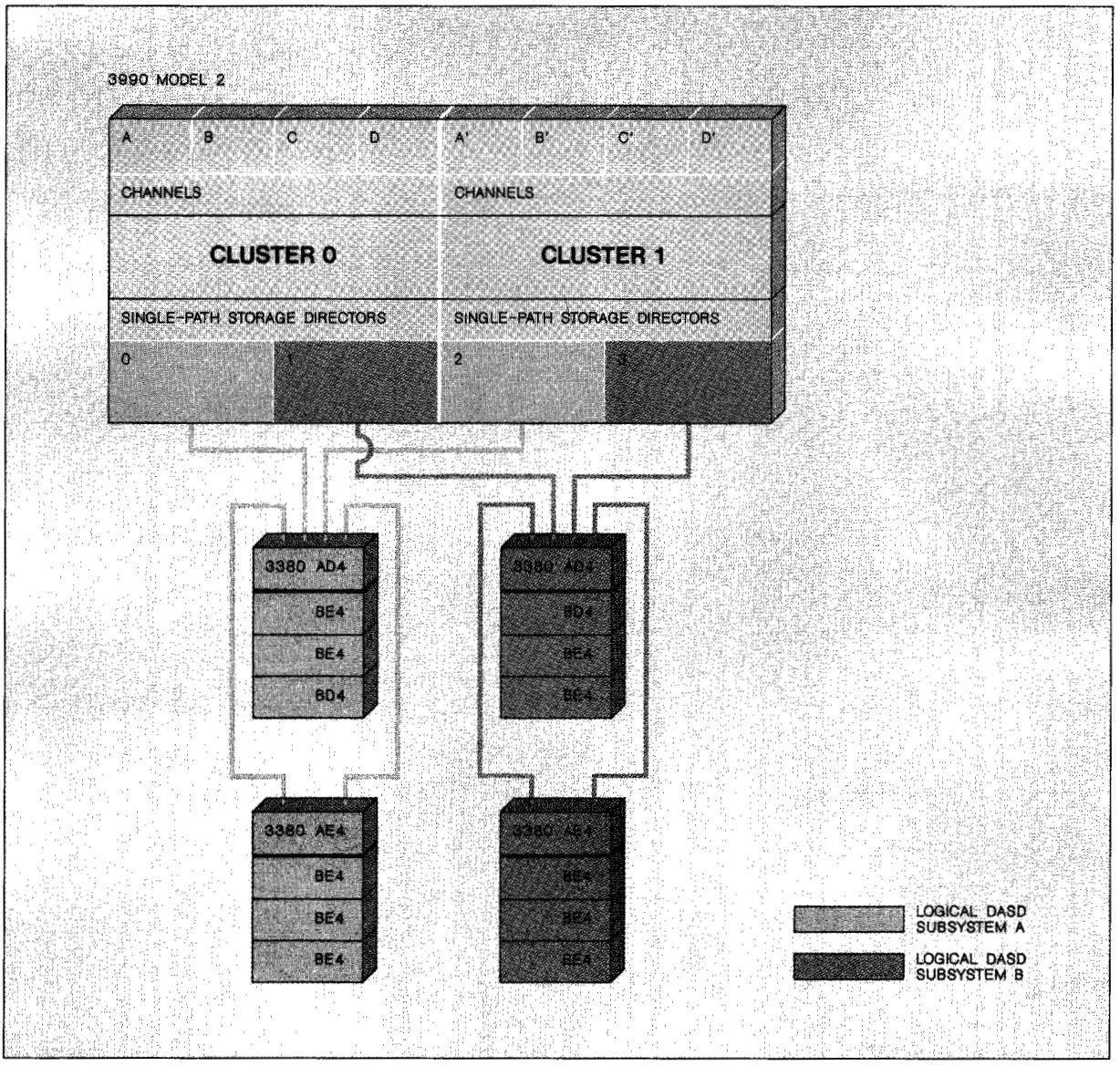
Configuring for availability. It is possible to configure a 3880 so that both of its storage directors attach to the same strings of DASD. However, for availability reasons, many installations configure DASD in such a way that a string of DASD is connected to a storage director in each of two different 3880 storage controls to eliminate the single points of failure between the two storage directors connected to the same strings of DASD. In 3880 Models 1, 2, or 3 this is called cross-configuration. (See Figures 4 and 5.) In a 3880 cache subsystem, it is implemented in a dual-frame configuration. Because the cache is shared by two storage directors, each director must be physically cabled to the cache. Furthermore, timing considerations dictate that they be as close together as possible. This is why two cache storage controls (of the same model type) are butted together and cross-cabled in a dual-frame configuration to provide the same capability. In either case, storage directors connected to the same DASD strings appear logically to the system as though they were in the same physical storage control.

Configuration of a 3990 Model 2 or Model 3 storage control may be somewhat different from the configuration of a 3880. Four-storage-path 3990 configurations can be likened to dual framed pairs of 3880 Model 23s, where the 3880 storage control is the equivalent of a cluster. When used in DLS mode with two-path 3380s, the 3990 is configured exactly as a cross-configured 3880 would be, with one storage path from each cluster connected to up to two strings of DASD to form one logical DASD subsystem. (See Figure 8.)

There is no equivalent to DLSE mode of operation in a 3880 configuration. In this case, there are four paths to the DASD, in two separate clusters. (See Figure 9.)

Two 3990 Model 2 storage controls or two 3990 Model 3 storage controls may be configured in a dual frame, as shown in Figure 10. This is similar to

Figure 8 3990 Model 2 in DLS mode with two logical DASD subsystems

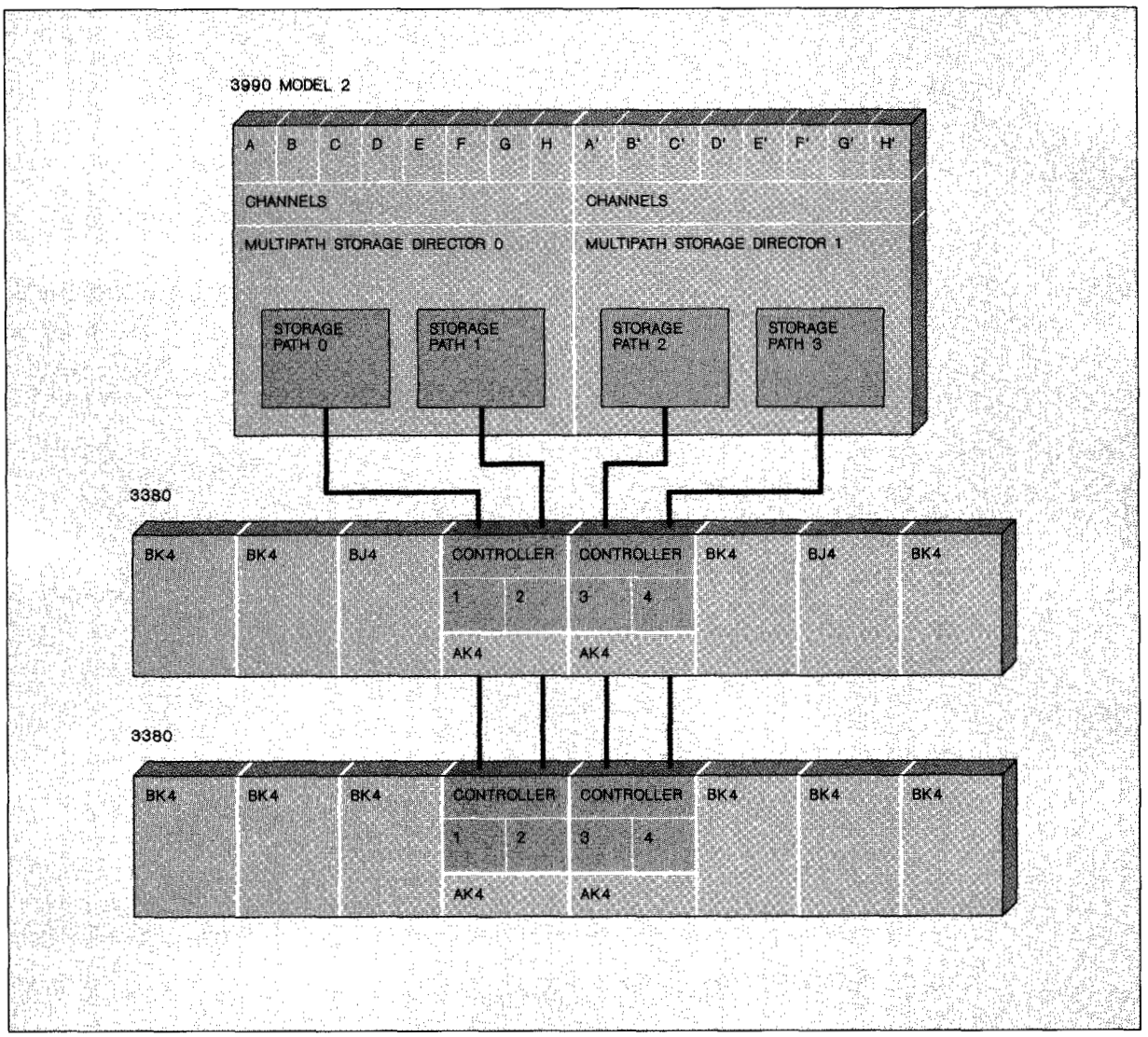


the dual-frame configuration used in the 3880 Model 23. The 3990 dual-frame configuration eliminates the single points of failure within the 3990 storage control that might prevent access to data.

Dual copy. Dual copy is a new option that protects key DASD data from device failures. With this option, the 3990 automatically writes all updates to a primary DASD volume and also to a secondary copy. (See Figure 11.) The two copies of the data are logically identical. If there is a hardware failure on the primary device, all I/O activity is switched automatically to the secondary device. By maintaining two copies of the data and providing an automatic switchover, the dual-copy function can improve system and application availability by eliminating single-device failures as a cause of outages.

Use of dual copy is transparent to application programs and to the operating system (except for some

Figure 9 3990 Model 2 in DLSE mode with optional four-channel switch additional feature



specialized error recovery and utility routines). No application changes are needed as long as standard IBM access methods are used, or for EXECUTE CHANNEL PROGRAM (EXCP), as long as standard record zero is used.

If there is a permanent data check (that is, a read error) on the primary volume, the data are automatically read from the secondary volume. Subsequent write operations go to the primary volume, as well as to the secondary one. (In practice, however, almost all read errors disappear when the record is rewritten.) On a write failure, the primary volume is taken out of operation or suspended, and the write operation is completed on the secondary. All subsequent I/O operations are directed to the secondary volume. In either case, the data recovery action is transparent to the application. Utility commands restore the configuration to normal after the hardware problem is fixed. The NVS keeps track of all changes to the active volume, so that if the failing device can be restored to operation without replacing the head-disk assembly (HDA), only the changed cylinders need to be copied. If the HDA is replaced

Figure 10 3990 Model 3 dual-frame configuration

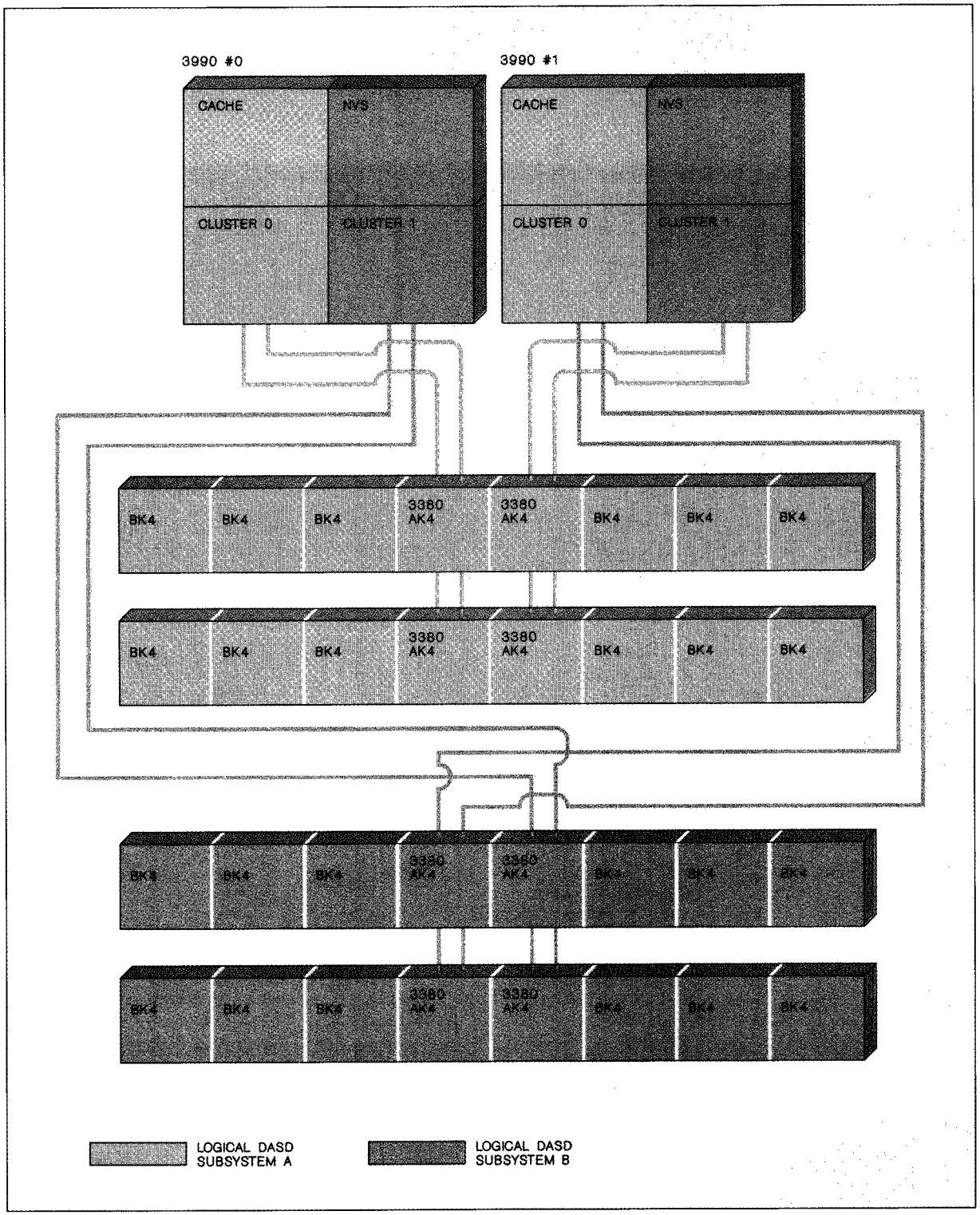
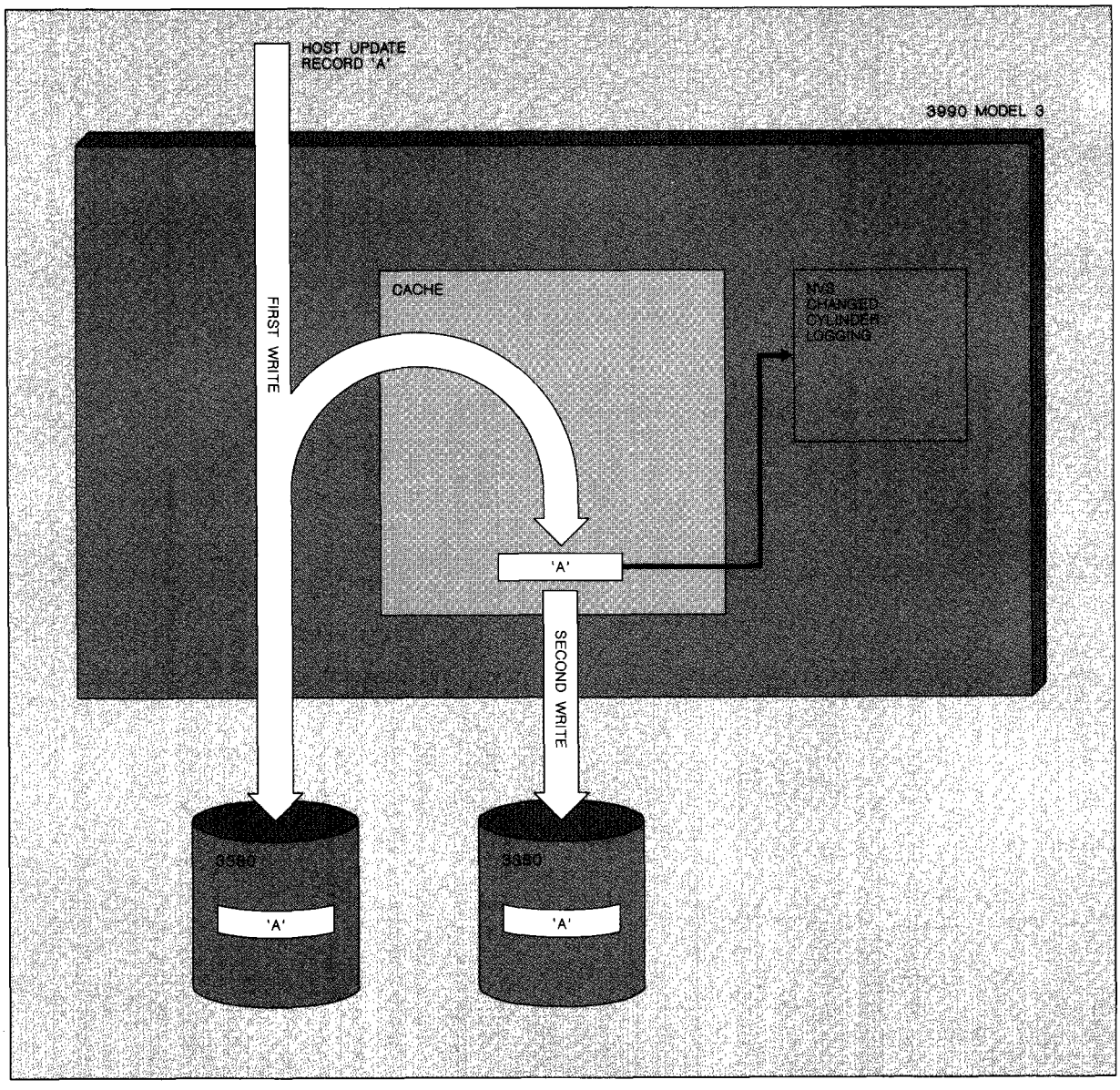


Figure 11 3990 Model 3 dual-copy operation



or the installation decides to restore dual-copy operations with a different secondary device, the subsystem is directed to perform a full volume copy. This can be done so that applications are not denied access to the data. By using a parameter in the utility command for restoring duplex operations, the installation can control the frequency of access by the host system, and thereby reduce, to some extent, the level of performance impact to the application or the duration of the copy operation. This capability is also available when only the changed cylinders need to be copied. This is advantageous because a very active volume can have many changed cylinders.

When the dual-copy function is active on a cached volume (recommended mode), performance should be equivalent to performance of the same volume

under a 3880 Model 23, or somewhat better. For the best possible performance, dual copy should be combined with the DASD fast-write function. Except for subsystems having high I/O rates, performance should be about the same as that of a 3990 cache with DASD fast write. (See Figure 16, later.)

Failing component isolation (fencing). *Fencing* is a technique used with recent releases of MVS and the 3880 and 3990 storage controls to limit the system-wide impact of a hardware component failure. MVS varies a channel path to a device off line when errors are encountered on one path, but I/O operations can be completed on other paths to the device. By fencing a failing component, only the remaining functional components are used. The failing component remains unavailable until the appropriate repair action has been taken.

By fencing a failing component, only the remaining functional components are used.

3880 hardware can likewise fence off a failing storage director or a DASD controller. The 3880 detects error conditions in either the channel or one of the DASD controllers that might prevent one storage director from completing I/O operations. The 3880 fences off the failing storage director and prevents further use of it. Similarly, the 3880 can fence off a failing DASD controller.

One of the more significant uses of fencing is WRITE INHIBIT. Certain hardware failures might cause erroneous data to be written to a device. MVS error-recovery programs, together with the 3880, cause all write operations through the failing component—whether a channel, storage director, or DASD controller—to be rejected, thereby limiting the potential damage that could be caused by the error condition.

The 3990 storage control extends the fencing capability even further. When operating in DLSE mode, with two storage paths in each storage director, an entire storage director fence is less likely. In most cases, just a single path is fenced. For another example, when certain errors occur on the connection between a storage path and the cache, that storage path may be fenced to allow the remaining paths to continue operating with the cache. In a number of situations, when error thresholds are exceeded, certain components may be fenced. One example is that of fencing a channel from a storage path. Other examples include fencing one path in a multipath storage director or fencing a device from a storage path, provided the other storage path in the multipath storage director is available.

Migration ease: Intermix configurations and nondisruptive DASD installation. The 3990 Models 2 and 3 can run in either DLS or DLSE mode. The correct mode is set by the customer engineer in the VPD at installation time. For migration purposes, DLS-only devices (3380 standards and 3380 Models D and E) can be attached to a 3990 operating in DLSE mode with a string of four-path 3380s. (Two-path J and K models cannot be intermixed with four-path J and K models in one subsystem.) This extends the 3880 DLS and two-path capability. Four-path DLSE subsystems represent a new level of basic storage control performance and availability.

All IBM storage controls—except the new 3990—require that the storage control and all the attached DASD be powered down for some part of the physical installation of a new device in the string. For installations that have extensive on-line demands, this can cause scheduling difficulties. New four-path DASD can be added to an existing DLSE mode 3990 Model 2 or 3 subsystem without affecting the system's capability to access data on existing devices in the subsystem. For this to work, the installation must plan for the eventual target configuration and describe the maximum desired configuration to both the IOGEN and the 308X/3090 IOCP to avoid a host-processor IPL and describe the same configuration to the VPD in the 3990. If the VPD must be changed to account for the configuration change, the 3990 has to undergo an initial microprogram load (IML), during which the host processor cannot access data in the subsystem.

If these installation-planning conditions are met, the DASD installation is completed by fencing one storage path at a time. The device is attached to that path. Diagnostics are run. Finally, that path is unfenced. The process is repeated for each of the four paths.

Serviceability. Serviceability is a critical characteristic of any data processing equipment. When there is a maintenance action, a configuration change, or installation required, the availability requirements of large systems make fast, sure action mandatory. This section describes several of the most significant components contributing to enhanced serviceability.

The functional microcode for the 3830 was written on an 8-inch diskette that was kept in the diskette drive inside the storage control.

Microcode diskettes. While the 2841 was a microprogrammed storage control unit, its read-only microcode storage and its small control storage limited its capabilities. The next DASD control unit—the IBM 3830—was truly innovative in that it had a diskette reader that accepted different microcode loads readily. The functional microcode for the 3830 was written on an 8-inch diskette that was kept in the diskette drive inside the storage control. This microcode was automatically loaded into the control unit every time it was powered on or at initial microprogram load (IML). The importance of this for control unit development cannot be underestimated. It meant that microcode programming could become much more sophisticated, it could be altered or replaced much more quickly, new device types could be incorporated into a DASD subsystem much more easily, and diagnostic capabilities could be greatly improved.

Key to many of the 3990 functions is the presence of read/write diskette drives that are similar to those used in personal computers. Just as the read-only diskette introduced with the 3830 opened many new possibilities for storage control enhancements, so does the use of the first writable diskette drive in an IBM storage control. The 3990 logs error information and temporary error counts for IBM customer engineers to use. The error information can be shared with a remote service support group through the Remote Support Facility, which provides a modem connection. This facility can also be used to download microcode patches to the storage-control diskette. The customer engineer can instruct the storage control to use the patch at the next IML or just store it on the diskette for future use.

Concurrent maintenance. Each of the major components of the 3990 (the two clusters, and if present, the cache and the NVS) is in a separate power and service region. They are independent of each other in the sense that a hardware problem in one will not cause failures in any of the others. (There may be loss of function or performance, depending on which component fails.) One cluster may still access the DASD, and use of the cache and NVS are unaffected when the other cluster is taken off line for service. Similarly, the NVS can be taken off line for service without preventing the use of the cache or the storage clusters. When the NVS is taken off line, DASD fast write will be terminated and dual-copy cylinder logging does not occur.

Status information. The 3990 retains status information for all of its devices in a storage area inside each cluster and on the writable diskettes. In addition, the Model 3 writes status information on device-status tracks inside the subsystem. The information includes configuration data, cache status, DASD fast-write status, cache fast-write status, and dual-copy status (primary volume, secondary volume, suspended duplex). This means that after a power-on or an IML, all devices inside the subsystem are automatically restored to their status at the time of the power-off (or just before the IML, if not powered off). This is an improvement over the 3880 Model 23, which could not store subsystem status across IMLs or power-off conditions. Thus, in a 3990 Model 3 subsystem, no operator or host-system action is required to ensure that the subsystem is reactivated with the desired cache, dual copy, and DASD fast-write configuration.

Diagnostics. In the 3830, diagnostic routines could be loaded by the IBM customer engineer and executed by the control unit microprocessor. These diagnostic routines called special control unit or device commands to aid in problem determination. Error-recovery microcode contained inside the storage control's functional microcode generated sense information that was returned to the host system. This information was used by the customer engineer to determine the nature of the problem.

Next to the RAS improvements made possible by increased circuit density and more internal error-checking circuitry, the most significant tool for improving the availability of a 3880 subsystem was the maintenance device (MD). The MD was a portable, independent, microprocessor-based tool that connected to the 3880. It contained maintenance analysis procedures (MAPs) that guided the customer engineer through the steps required to identify hardware problems. Because the MD replaced many of the procedures previously contained in the maintenance manuals and host-dependent test routines, it provided more efficient problem determination. The MD collected data from the 3880, which could be analyzed on site by the customer engineer, or in unusual situations the data could be transmitted via telephone link to a remote service support group. If necessary, the MD could receive microcode patches from the remote site, which could then be written to the control storage of the 3880. However, because the 3880 has no way of storing this data across IMLs, the patches had to be reapplied at each IML of the 3880.[22]

The 3990 has been designed to continue operation to the degree possible after a failure, although perhaps in a degraded mode.

The support facility is an important feature of the 3990 because it performs the same functions as the MD used in the 3880 but with many enhancements. Each cluster has its own support facility, which is a major contributor to the enhanced RAS characteristics of the 3990. The support facility performs such critical functions as the following:

- Subsystem status monitoring
- Power sequencing and monitoring
- Initial microprogramming loading
- Diagnostics and MAPs
- Error logging
- Error detection and fault isolation (EDFI)
- System information message (SIM) generation
- Remote support
- Entering and retaining configuration information (i.e., Vital Product Data)
- Microcode patch application

All of the diagnostic routines and the maintenance procedures are contained on the microcode diskette. The customer engineer uses a CE panel attached to the support facility to begin diagnostics and to select the necessary routines to be executed. The support facility guides the customer engineer through the procedures. The self-contained, built-in support facility offers the improvements just listed compared to the separate diagnostic diskette used in the 3830 and the MD used in the 3880. In addition, diagnostic routines can be executed on one storage cluster without disrupting data access through the other cluster.

Fault-tolerant operation and error detection and fault isolation. The 3990 has been designed to continue operation to the degree possible after a failure, although perhaps in a degraded mode. Not only is the circuitry of one storage cluster replicated in a second cluster, but it also has extensive error-retry capabilities. Even if a permanent hardware failure occurs on one path (i.e., that path is not recoverable), data access continues through the remaining paths, and at this point, the installation must decide which of the following actions to take: (1) take the subsystem or a subset (cluster, cache, NVS) off for repair, or (2) defer maintenance until a more convenient time. In the past, this was a difficult decision to make, because the storage control did not supply sufficient information. Usually, the failure had to be reproduced to determine the effect of a repair action.

Sophisticated error detection and fault isolation (EDFI) circuitry and microcode routines in the 3990 identify the effect of a failure, the effect of the repair, and even the parts that may have failed. This information, contained in a SIM record, for the first time allows an installation to assess the effect of a failing component in an IBM storage control and make an informed decision about scheduling the repair action. Furthermore, because the installation can provide the SIM information to the support center when the service call is placed, the customer engineer knows which parts to bring to repair the machine. The net result is fewer system interruptions and outages of shorter duration.

Configuration definition. Previous IBM storage controls, such as the 3830 and the 3880, used switches to define operating modes and configuration information. In the 3990, these switches are replaced by Vital Product Data (VPD), a microcode-based function inside the support facility for recording configuration information. The customer engineer is prompted for this information during installation, it is checked for validity, and stored on the diskette. It is then read into the control storage of the 3990 at IML time.

Hardware identification. The large number of storage controls in many installations make identification of a specific unit more difficult, because the addresses that describe a device within a given system might not be unique across the complex. Easier identification of the 3880 storage director that is causing a sense record to be generated is done by including a unique storage director ID (SDID) in the sense information supplied by the storage control. This ID is assigned by the customer engineer at installation time and should be unique to that storage director within the installation. Thus, sense information can be collected for a specific storage director. The SDID is also used by the caching models to report caching status and usage statistics. In the 3990, the SDID is replaced by the subsystem identifier (SSID). In this case, the two storage directors comprising a logical DASD subsystem have the same SSID. One advantage of this is that now the Model 3 subsystem caching statistics are presented on a logical subsystem basis, not on a storage director basis.

Channel command retry, selective reset, and error-correction code

The 3830 storage control contains many RAS enhancements. One such enhancement is channel command retry. In previous DASD subsystems, the host-system software was responsible for re-executing an entire channel program if the channel program terminated with an error condition. Not only was this inefficient, but in certain circumstances it was also impossible to reconstruct the channel program correctly. The 3830, working in conjunction with the System/370 channel, can hold the last channel command in the channel and have the channel redrive it on request.[22]

Channel command retry can also handle other special, nonerror conditions, such as read miss processing in the cache storage controls. (This is a process described later in this paper.)

Most error conditions can be recovered by channel command retry, if they are caused by transitory conditions. Errors of a more permanent nature, such as parity errors or control store errors, cannot be recovered in this way, but some of them can be recovered using a System/370 command: SELECTIVE RESET. The control unit initiates this process by issuing an I/O error alert sequence. In response, the channel issues a SELECTIVE RESET command to restart the microprocessor and execute microcoded error-recovery routines. In case the error cannot be recovered, the control unit disengages from the channel so that it does not interfere with I/O operations to other control units on the channel.[22]

Most error conditions can be recovered by channel command retry, if they are caused by transitory conditions.

The 3830/3330 subsystem supplies an error-correction code (ECC) to aid in correcting burst errors in data read from disk. Host-system software uses this ECC to correct most read errors.[22]

When attached to the 3380 Enhanced Subsystem Models (AJ and AK), the 3880 Models 3 and 23 and all 3990 models offer an important enhancement to error correction, called outboard ECC. This capability is made possible by enlarging, from 512 bytes to 64K bytes, the automatic data transfer buffer (ADT), which transfers data between device and channel.[23,24] This buffer is sufficiently large to hold an entire 3380 track. When the DASD detects an ECC correctable error, the storage control automatically corrects the error in the ADT buffer, using the ECC information provided by the DASD A-unit. Before this capability was available, the 3880 would transfer the record in error and the ECC code to the host system, whose DASD error-recovery programs then performed the necessary correction. The 3880 Models 3 and 23 and the 3990 now do this with no host-system overhead for 3380 AJ and AK models.

Cache

3880 cache storage controls. The next step in storage control evolution was the introduction of the first IBM cache storage control, the 3880 Model 13, and its successor, the Model 23. These storage controls incorporated a cache storage under control of the 3880 microcode to improve the performance of the DASD subsystem. The design of these machines is based on the principle of locality of reference. That is, if a record is accessed, there is a high probability that it or a close neighbor will also be accessed within a short period of time. These storage control models use least-recently used (LRU) algorithms to manage the cache storage and provide higher levels of subsystem performance. In essence, they insulate DASD subsystem I/O from much of the physical device movement. For the first time, it is possible for a portion of a DASD subsystem's I/O to complete without the host processor waiting for physical device positioning. (See Figure 12.)

In a 3880 cache storage control, read operations benefit from caching.

Read operations in a 3880 cache storage control. In a 3880 cache storage control, read operations benefit from caching.[25] If a record requested by the host is in the cache, it is immediately sent across the channel to the host processor. (Finding the requested record in the cache is called a *read hit*.) All electromechanical motion—seek and latency—is eliminated. RPS miss is eliminated, because there is no need for a disconnect from the channel to complete an I/O operation. The more read hit operations there are, the better the subsystem performance.

A read miss occurs when the record requested is not in the cache. In this case, the requested record and all subsequent records on the track are staged into the cache. That is, they are read from the DASD into the cache. Channel command retry is used to complete the read operation. In a 3880 Model 23, the record is transferred to the channel and simultaneously staged to the cache. This concurrent transfer to the channel and the cache is called a branching transfer. Overlap of these operations provides a significant performance enhancement. This action takes about the same amount of time as a noncached DASD I/O operation. The other records on the track are staged in anticipation of future requests by the host processor for other data on the same track.

Write operations in a 3880 cache storage control. In a 3880 cache, write operations do not benefit from the cache, because the cache storage is volatile. If there is a power failure, data in the cache are lost. Applications usually cannot tolerate this uncertainty. When the storage control signals that the I/O operation has completed (device end), applications assume that the data have been written to DASD. For this reason, when a write hit (that is, the record being updated is in the cache) occurs, the 3880 Model 23 performs a branching transfer of the data to both the cache and DASD. Because the two write operations are simultaneous, write hit performance is equivalent to a noncached DASD write operation. Write misses go directly to DASD.

Other modes of operation. 3880 cache I/O operations can be modified through the DEFINE EXTENT channel command. DEFINE EXTENT instructs the storage control how to process a record. The default mode is normal caching, where hits and misses are processed as previously described. However, several other modes are available, of which perhaps the most important for performance is sequential mode. Sequential access methods, such as QSAM, set the sequential bit on in the DEFINE EXTENT channel command. When the storage control senses that this bit is on, it automatically begins staging the next track after the first one has been accessed by the host processor. Each time the channel program accesses a new track, the next sequential track is staged in anticipation of its use. (See Figure 15, later.)

Because the default mode is normal caching, most applications can effectively use the cache without modification. An access method modification is required only when the application needs to use one of the special cache modes.

3990 Model 3 enhancements

The 3990 Model 3 provides improved cache management through track segmentation. Previous IBM

Figure 12 IBM 3880 Model 23 cache operations

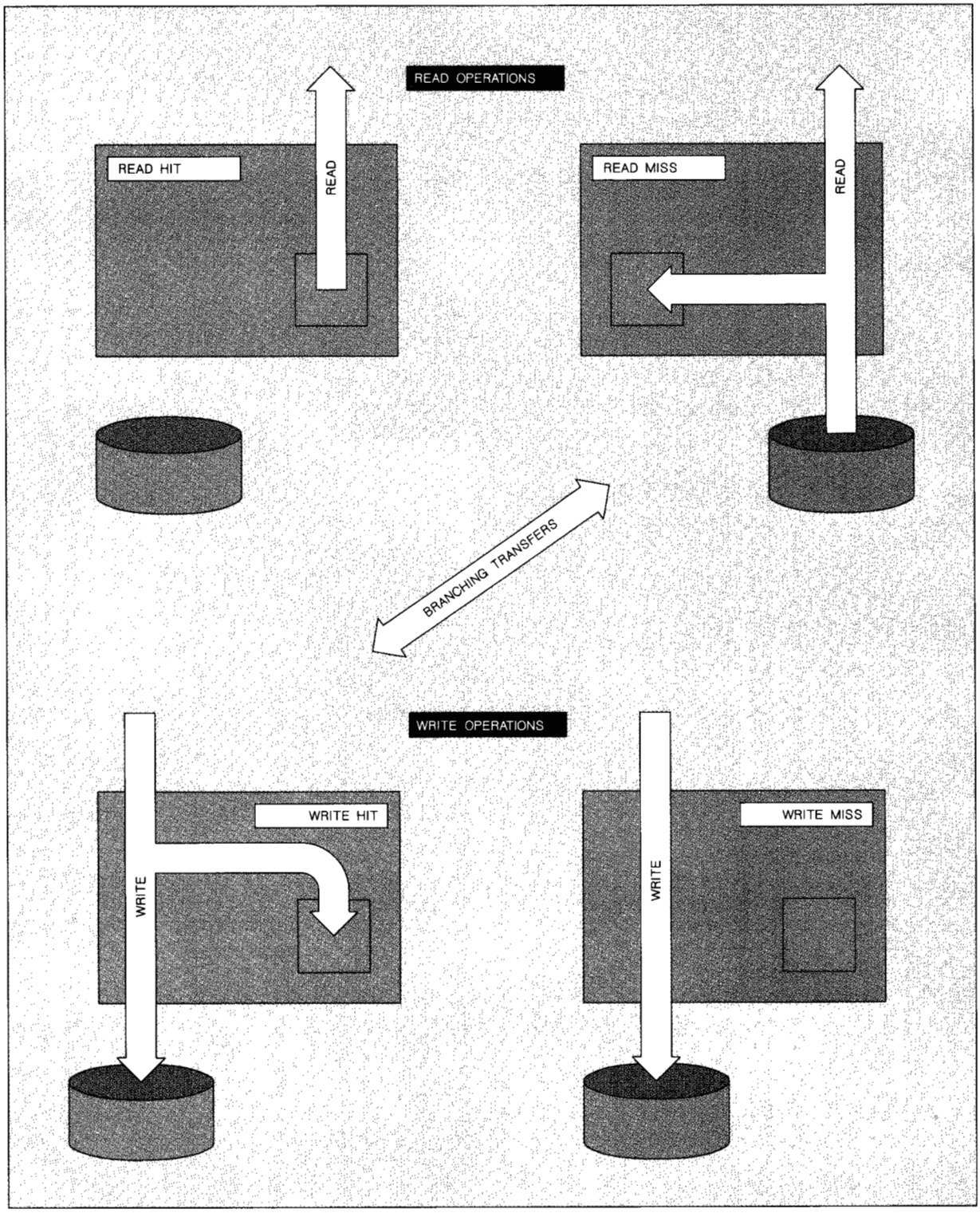
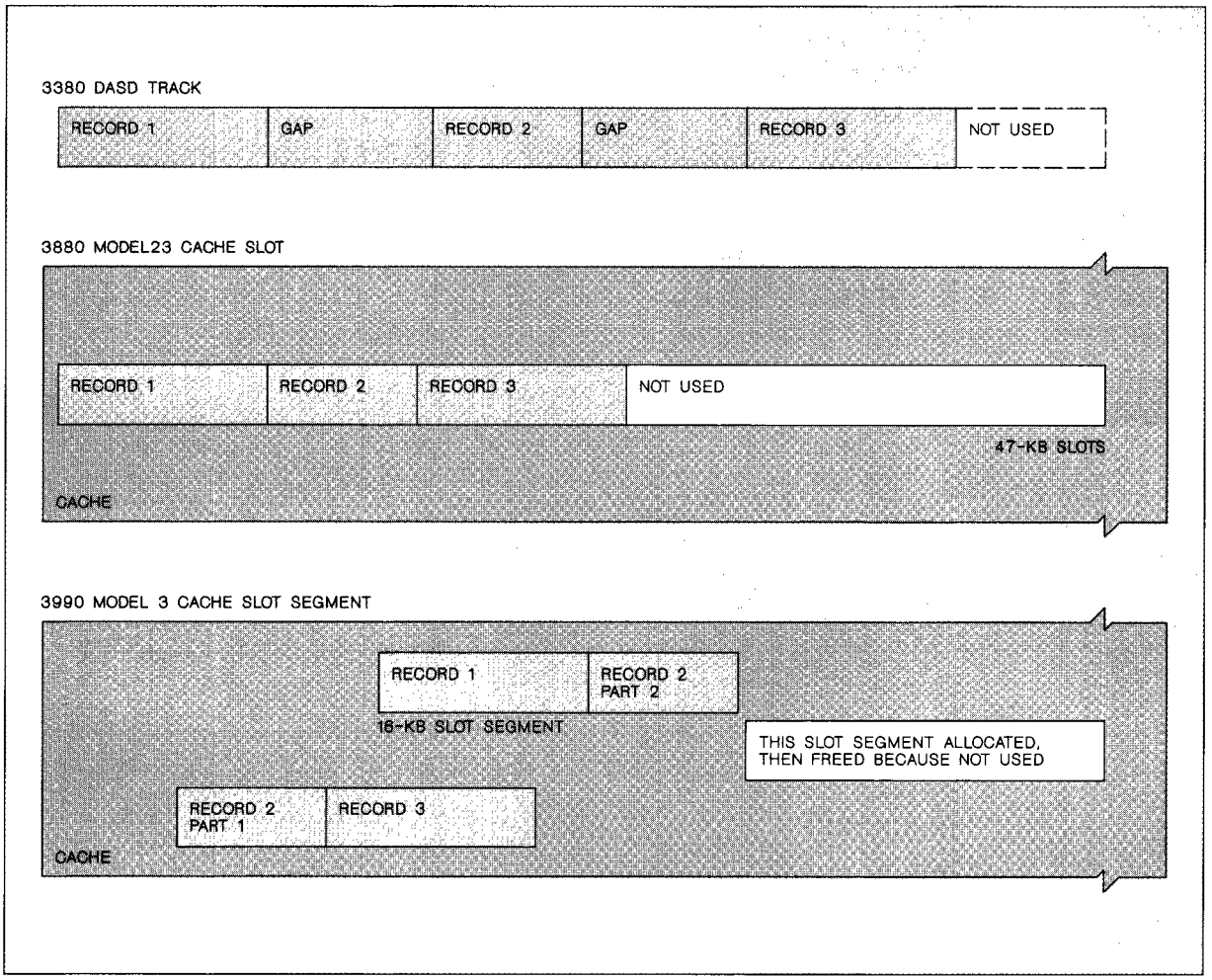


Figure 13 Cache space management

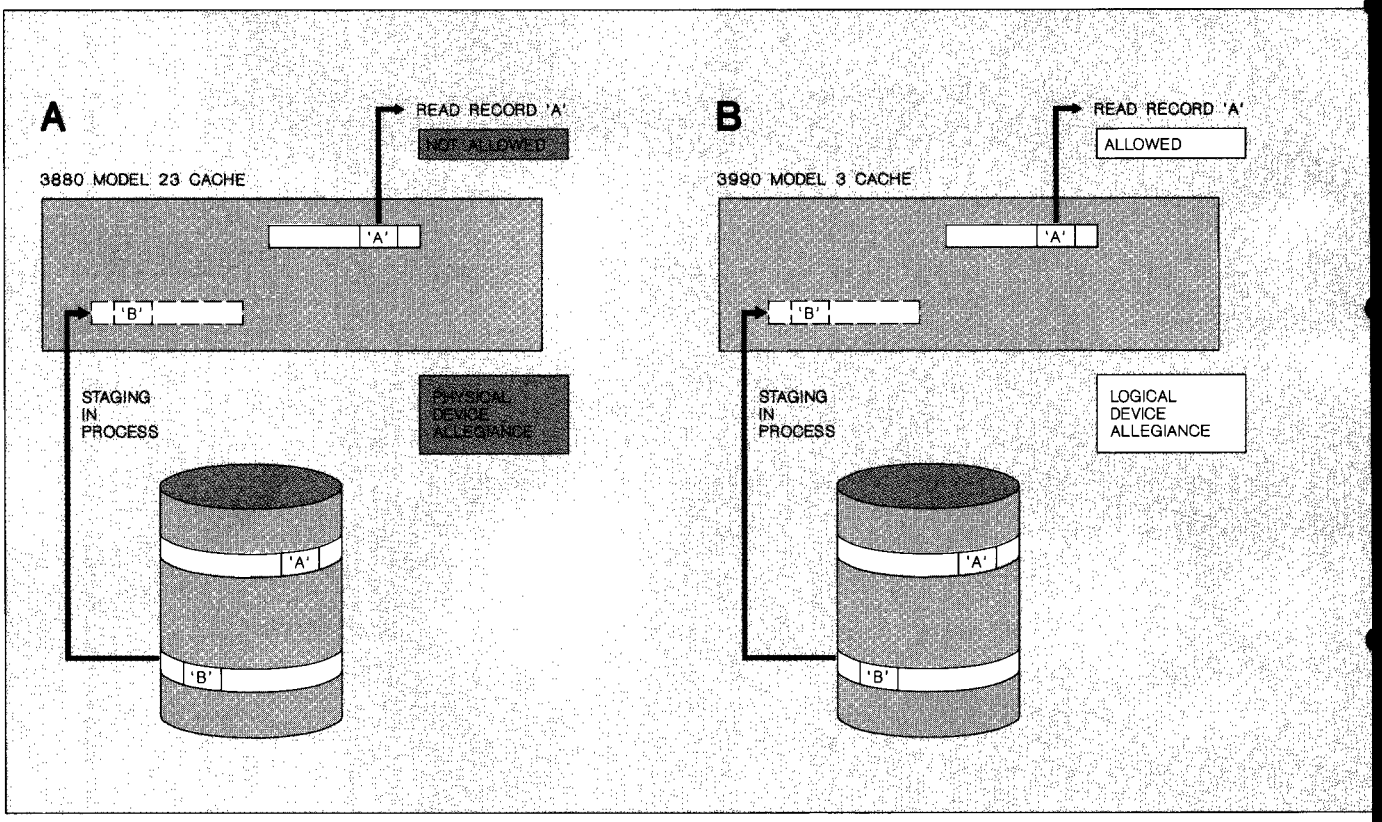


cache models managed the cache on a track basis in 47-Kb track slots. In the 3990 Model 3, each track slot is divided into three, possibly noncontiguous, 16-Kb track slot segments. When the data staged into the cache fit into one or two slots, the unused slots are freed for other tracks to use. Since the gaps—i.e., the empty spaces between records on DASD—are not loaded into cache, space savings can be significant. Modeling runs have shown that this results in better cache utilization, which means that for an equivalent DASD subsystem response, less cache is required. This cache space management is shown in Figure 13.

Dual data transfer allows a storage path to manage simultaneously a data-transfer operation between the channel and cache (the upper interface) and between the cache and a device (the lower interface). An operation between the channel and the cache can access the same logical device that is busy staging or destaging data on the lower interface, as long as different tracks are being accessed. This logical device allegiance introduces a higher level of concurrency than ever before possible in an IBM storage control. (See Figure 14.)

Cache sequential operations are improved by staging ahead two tracks instead of one, and retaining in cache the last track accessed. Logical device allegiance ensures that host-processor access to the tracks in cache is allowed even while data are being staged from DASD. Some applications, such as the

Figure 14 Cache/device access improvements: (A) physical device allegiance; (B) logial device allegiance;



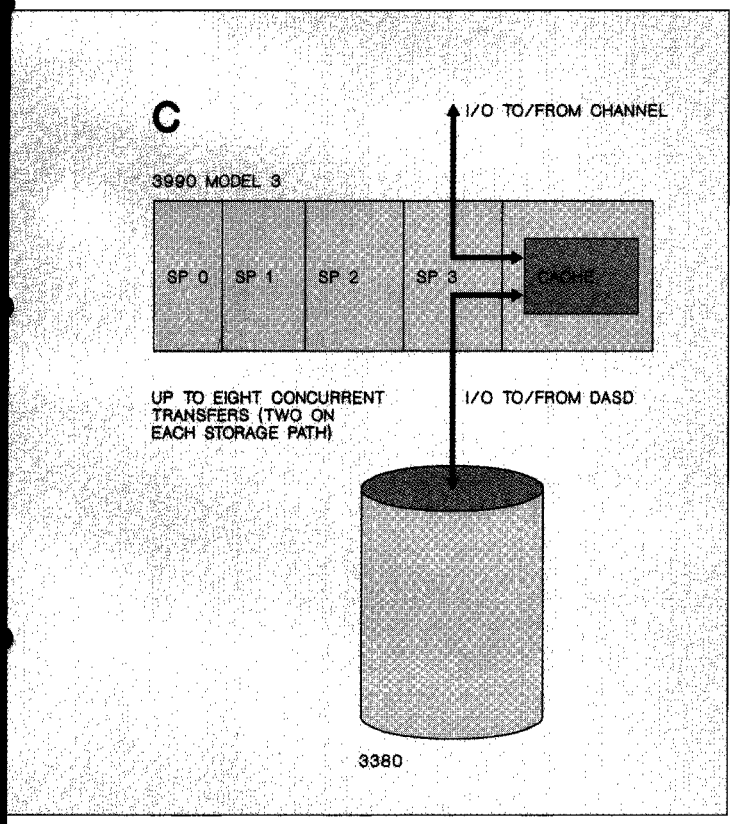
Information Management System (IMS), occasionally refer back to records previously read. By keeping these tracks in cache longer, cache miss situations can be minimized. Figure 15 illustrates sequential operations.

The faster microprocessors used in the 3990 improve execution times for various I/O operations. However, this is only part of the improvement the new storage control offers. Even more significant gains are possible because of the increased concurrency of internal operations, more efficient use of subsystem resource, use of DLSE, and new functions such as DASD fast write and cache fast write. These gains are demonstrated in the modeled performance curves shown in Figure 16.

DASD fast write. DASD fast write extends the performance benefits of caching to write hits, where the definition of write hit has been expanded to include format writes of an entire track. (See Figure 17.) The 3880 Model 23 did not provide any performance benefit for write operations because of the requirement to write all updates to DASD before signaling DEVICE END. Applications consider data safely written to DASD when DEVICE-END status is returned. When DASD fast write is in use, a write hit causes a branch write of the record to both the cache and the NVS. Because there is now a nonvolatile copy of the record in NVS, DEVICE END can be returned to the host processor. The 3990 asynchronously writes the record to DASD. If there is a cache or NVS failure before this operation is complete, the remaining copy is emergency-destaged to disk. If the record cannot be written to DASD, for example, if room power is lost, the record is retained in the NVS and is automatically destaged from NVS to disk at storage control IML.

Use of DASD fast write is automatic for any write operation to a volume with cache and DASD fast write enabled. No application changes are needed. If necessary, DASD fast write can be disabled at the channel program level by setting a parameter for the

(C) 3990 Model 3 dual data transfer



DEFINE EXTENT command. The cache management algorithms treat write hits and misses in the same way they treat read hits and misses.

Cache fast write. Cache fast write is a 3990 Model 3 performance option intended for uses where a nonvolatile copy of the data is not required. Recent releases of IBM DFSORT use cache fast write for sort-work files. Because these files are temporary and the usual recovery from a hardware error is restarting the sort from the beginning, there is no need to write the data to disk. Typically, all write hits are written to cache only; the NVS is not used, and DEVICE END is given immediately. Cache fast-write data are destaged to DASD only if required because the cache is full or the host software instructs the 3990 to destage it. (See Figure 18.)

To invoke cache fast write, the application must do two things: (1) Set the cache fast-write bit on in the DEFINE EXTENT channel command, and (2) obtain and use the current cache fast-write token. Because it is possible for the cache to be powered down, or the cache directory to be relocated without the application being notified, the token was implemented to ensure that the application accesses the correct data. Each time the storage control has an IML or relocates its cache directory, a new cache fast-write token is created. When an application prepares to use cache fast write, it obtains the current token value, which it sends back to the storage control in each channel program using cache fast write. The storage control compares this token value to its current value. If the two do not agree, the application is notified. The application may terminate with an I/O error or execute a special error recovery routine to continue execution. Note that this mechanism is not required for DASD fast write because either a cache failure or a relocation of the directory will cause an emergency destage from the NVS.

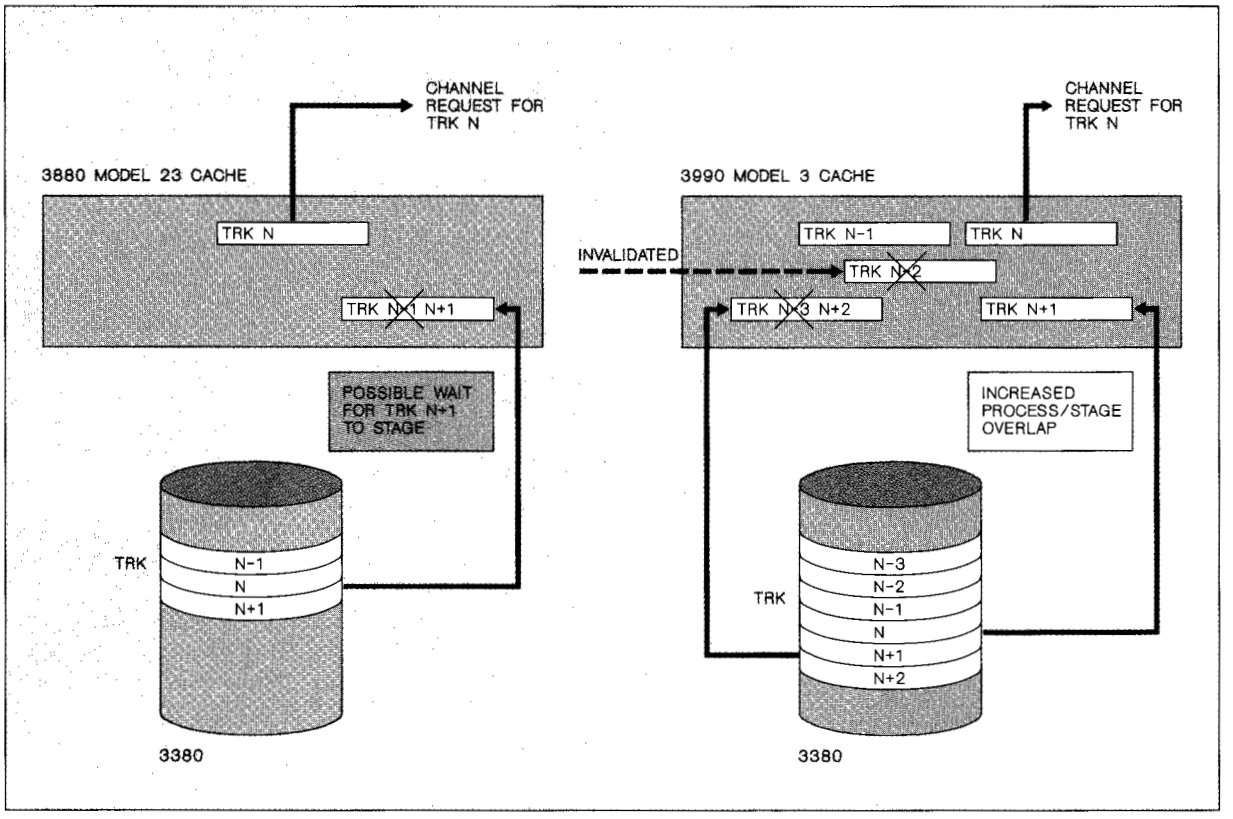
Cache fast write is available for all caching volumes within a subsystem, unless it is turned off for all volumes by using a special command. If cache fast write is invoked in a channel program for a volume that has DASD fast write active, cache fast write will be used for that channel program. All other write channel programs automatically invoke DASD fast write. If dual copy is in use for a volume, the cache fast-write parameter for the DEFINE EXTENT command is ignored—the write operation is completed as if cache fast write had not been specified.

Software interactions

Software (MVS/ESA or MVS/XA and MVS/DFP™) plays an integral role in achieving the capabilities of the 3990 family of storage controls, and making these DASD subsystems easier to administer. This software includes the following capabilities:

- Extensions to IDCAMS (the program name for access method services) utility functions and operator commands to establish, manage, monitor, and recover an extended-function subsystem (that is, a 3990 Model 3 subsystem using DASD fast write and/or dual copy)
- Interactive Storage Management Facility (ISMF) panels to define cache and extended-function characteristics of DASD volumes and subsystems
- DFSMS™ management of 3990 subsystems
- Working with MVS/ESA to optimize the performance and availability of critical system data sets
- SIM logging and interpretation
- Reset event notification

Figure 15 Cached sequential processing

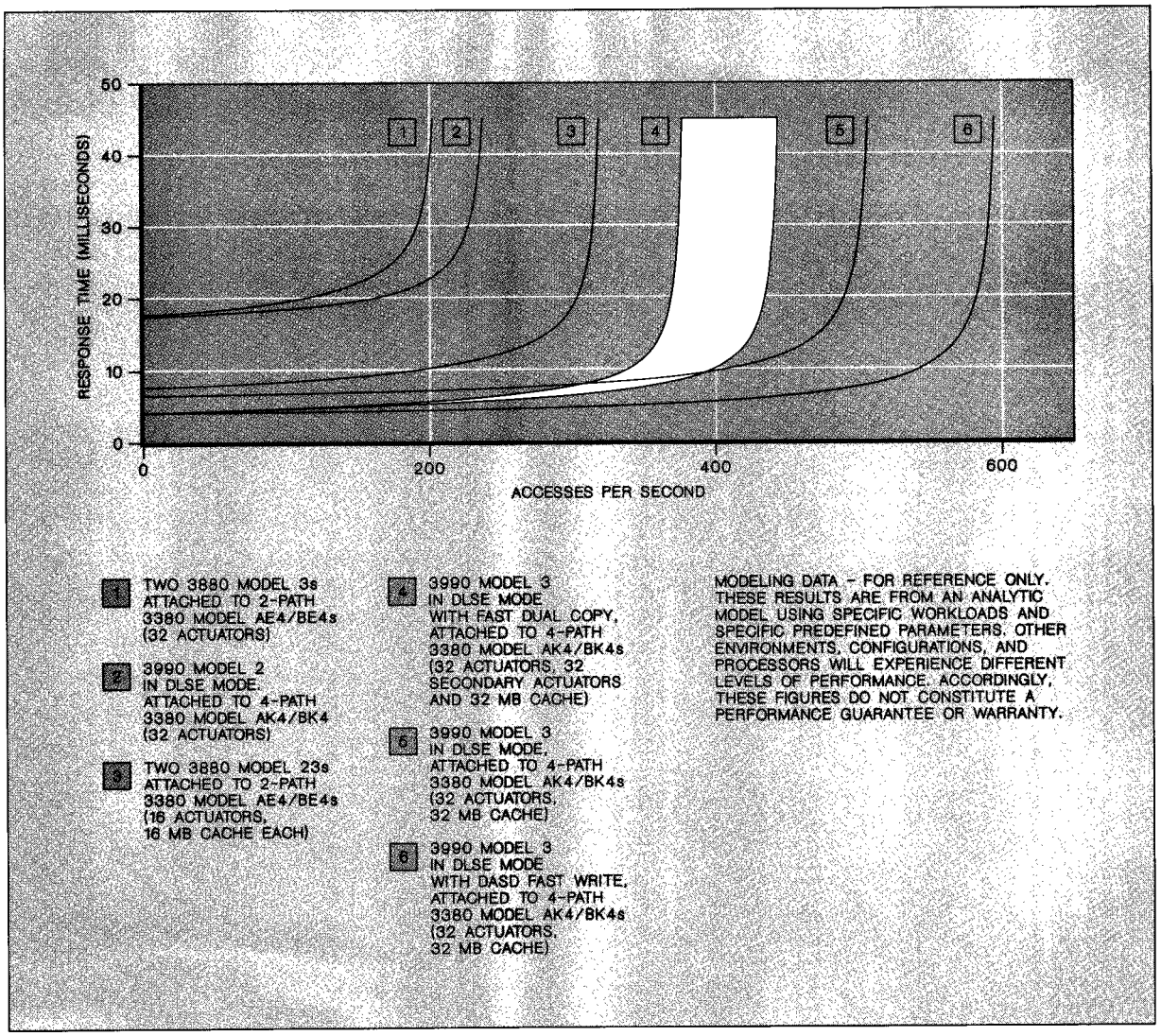


DFSMS provides many new capabilities for storage management, most of which are beyond the scope of this paper. However, among its functions are those for controlling cache and extended-function resource at a data set level. When users follow installation-defined rules for data set allocation, data sets that require the high availability protection afforded by dual copy can be allocated automatically to dual-copy logical volumes. Likewise, data sets requiring the highest levels of performance can be directed to volumes using DASD fast write.

MVS/ESA provides the capability to place critical data sets, records, or buffers in processor expanded storage to improve system performance. The 3990 Model 3 together with MVS/ESA ensure that these data receive the best possible availability and performance conditions. Availability is enhanced through dual copy, and performance improvements can be achieved in a variety of ways, depending on the type of data set and software implementation. For example, heavily used partitioned data sets can have their directories resident in processor expanded storage for quick look-up. The partitioned data set members themselves get best performance when they are placed on a 3380 using the DASD fast write capability of the 3990 Model 3.

Two other software capabilities are central to the improved RAS characteristics of the 3990 storage control family. One of these, the SIM, has been previously mentioned. When the 3990 detects a permanent error, or when temporary errors exceed thresholds, a special sense record—called a SIM—is sent to the host processor whose software logs the SIM to the Error Recording Data Set (ERDS) and generates a SIM alert console message. Environmental Record Editing and Printing Program (EREP) interprets the SIM and provides the installation with failure impact and repair impact information. The alert also

Figure 16 Comparison of TSO performance with different storage controls



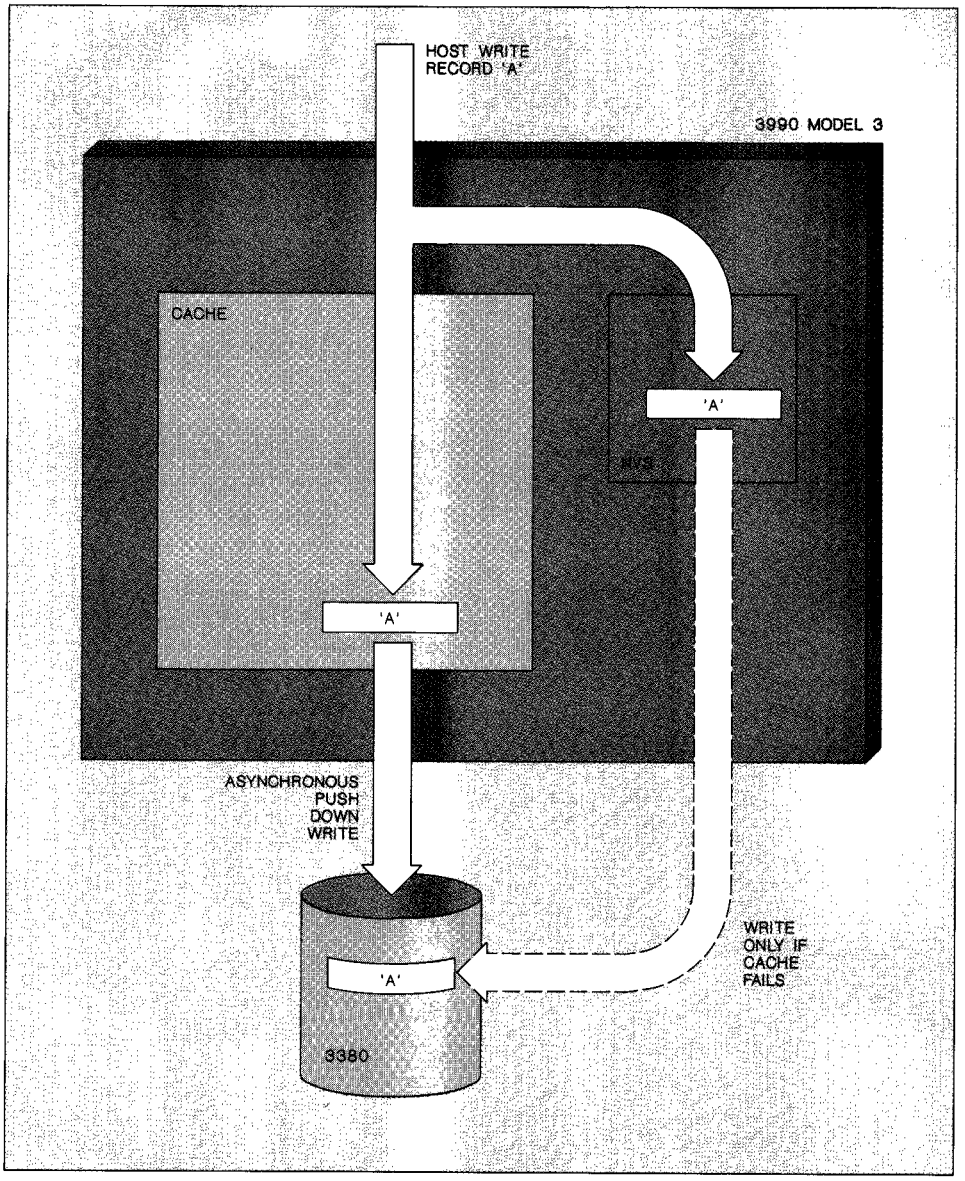
provides the IBM customer engineer with sufficient information to identify which repair parts to bring on site to repair the problem.

The other capability, reset event notification, allows the storage control to indicate to the host processor any condition that could cause the shared control array information to be out of synchronization with host-device and path status. Host-processor software is then invoked to resynchronize the information, which thus reduces the need for corrective action by the installation.

Concluding remarks

The four generations of IBM storage controls, beginning with the 2841 and System/360, have exhibited steady growth and development of new functions and performance capability through improvements in hardware technology, packaging techniques, innovative architecture, microcode, and close interaction between the hardware and the host-processor operating systems. The current high point of this line of progress is the IBM 3990 Model 3 storage control, whose major capabilities include the following:

Figure 17 3990 Model 3 DASD fast-write operation



- Improved DASD subsystem performance
- Applicability of cache to almost all data in an installation, owing to large cache sizes, more efficient cache and storage control internal operations, and DASD fast write
- New levels of hardware device availability through the dual-copy function
- Enhanced RAS through its new cluster design, fault tolerance (through fencing and redundant components), improved error detection and reporting, remote maintenance support, nondisruptive DASD installation capability, and support facility
- When used with the expanded storage capabilities of MVS/ESA, the highest performance of an IBM

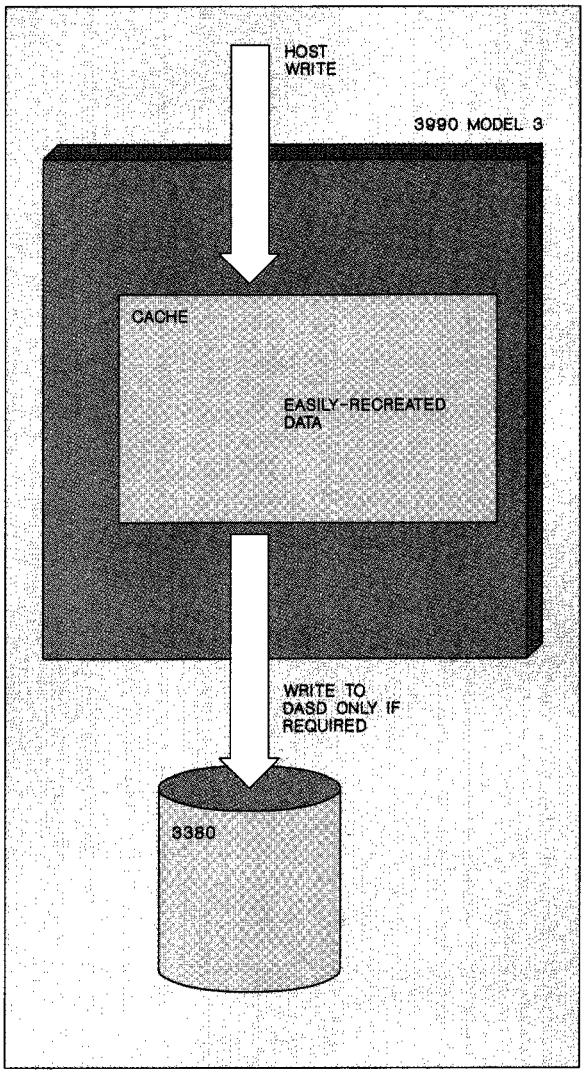
3090/MVS/ESA system

- When used with DFSMS, a higher level of automated data set placement and management, based on installation-defined criteria for performance and availability

Cited references

1. W. Buchholz, "The system design of the IBM Type 701 computer," *Proceedings of the IRE* **41,** 1262–1275 (1953).
2. J. L. Greenstadt, "The IBM 709 computer," *Proceedings of the Symposium on New Computers, a Report from the Manufacturers*, Los Angeles, CA, March 1957, pp. 92–96.
3. C. J. Bashe, W. Buchholz, G. V. Hawkins, J. J. Ingram, and N. Rochester, "The architecture of IBM's early computers," *IBM Journal of Research and Development* **25,** No. 5, 363–375 (September 1981).
4. C. J. Bashe, P. W. Jackson, H. A. Mussell, and W. D. Winger, "The design of the IBM Type 702 system," paper no. 55-719, *AIEE Transactions* **74,** Part 1, *Communication and Electronics*, 695–704 (1956).
5. C. J. Bashe, W. Buchholz, and N. Rochester, "The IBM 702, an electronic data processing machine for business," *Journal of the ACM* **1,** 149–169 (1954).
6. L. D. Stevens, "The evolution of magnetic storage," *IBM Journal of Research and Development* **25,** No. 5, 663–675 (September 1981).
7. W. W. Peterson, "Addressing for random access storage," *IBM Journal of Research and Development* **1,** No. 2, 130–146 (March 1957).
8. D. T. Brown, R. L. Eibsen, and C. A. Thorn, "Channel and direct access device architecture," *IBM Systems Journal* **11,** No. 3, 186–199 (1972).
9. A. Padegs, "System/360 and beyond," *IBM Journal of Research and Development* **25,** No. 5, 377–390 (September 1981).
10. *IBM System/370 Principles of Operation*, GA22-7000, IBM Corporation; available through IBM branch offices.
11. *IBM System /360 and System/370 I/O Interface Channel to Control Unit Original Equipment Manufacturer's Information*, GA22-6974, IBM Corporation; available through IBM branch offices.
12. W. A. Clark, "The functional structure of OS/360, Part III: Data management," *IBM Systems Journal* **5,** No. 1, 30–51 (1966).
13. R. E. Matick, *Computer Storage Systems and Technology*, John Wiley and Sons, Inc., New York (1977).
14. *IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide*, GA32-0100, IBM Corporation; available through IBM branch offices.
15. *IBM System/360 Component Descriptions—2841 and Associated DASD*, GA26-5988, IBM Corporation; available through IBM branch offices.
16. J. I. Norris, "A high performance microprocessor," *Disk Storage Technology*, 27–30, IBM Corporation (1980); available through IBM branch offices.
17. S. J. Duchak, "Maintenance of the IBM 3880 storage control," *Disk Storage Technology*, 78–82, IBM Corporation (1980); available through IBM branch offices.
18. E. G. Coffman, Jr., and P. J. Denning, *Operating Systems Theory*, Prentice-Hall, Inc., Englewood Cliffs, NJ (1973).

Figure 18 3990 Model 3 cache fast-write operation



19. *Introduction to the IBM 3880 Storage Control Model 23*, GA32-0082, IBM Corporation; available through IBM branch offices.
20. L. C. Blount, R. G. Edison, G. L. Grossman, J. S. Hyde, W. E. Langstroth, S. E. Williams, *IBM 3880-23 Performance Measurements with IBM 3380 J/K*, GG09-1007, IBM Corporation; available from IBM branch offices.
21. Tik-Fai Cho, "The design concept of the IBM 3880," *Disk Storage Technology*, 63–64 (1980), GA26-1665, IBM Corporation; available through IBM branch offices.
22. G. R. Ahearn, Y. Dishon, and R. N. Snively, "Design innovations of the IBM 3830 and 2835 Storage Control units," *IBM Journal of Research and Development* **16,** No. 1, 11–18 (1972).

23. H. Bardsley, "The IBM 3880 microprocessor control," *Disk Storage Technology*, 69–72 (1980), GA26-1665, IBM Corporation; available through IBM branch offices.
24. *IBM 3880 Storage Control Models 1, 2, 3, and 4 Description*, GA26-1661, IBM Corporation; available through IBM branch offices.
25. C. P. Grossman, "Cache-DASD storage design for improving system performance," *IBM Systems Journal* **24**, Nos. 3/4, 316–334 (1985).

General references

IBM 3880 Storage Control Model 23 Description, GA26-1661, IBM Corporation; available through IBM branch offices.

IBM 3990 Storage Control Introduction, GA32-0098, IBM Corporation; available through IBM branch offices.

IBM 3990 Storage Control Reference, GA32-0099, IBM Corporation; available through IBM branch offices.

IBM 3380 Direct Access Storage Introduction, GC26-4491, IBM Corporation; available through IBM branch offices.

Using the IBM 3380 Direct Access Storage in an MVS Environment, GC26-4492, IBM Corporation; available through IBM branch offices.

MVS/Extended Architecture Storage Management Library: Configuring Storage Subsystems, GC26-4262, IBM Corporation; available through IBM branch offices.

IBM System/370 Extended Architecture Principles of Operation, SA22-7085, IBM Corporation; available through IBM branch offices.

Reference Manual for IBM 3830 Storage Control and IBM 3330 Disk Storage, GA26-1592, IBM Corporation; available through IBM branch offices.

Reference Manual for IBM 3830 Storage Control Model 2, GA26-1617, IBM Corporation; available through IBM branch offices.

Carol Porter Grossman *IBM General Products Division, Tucson, Arizona 85744.* Mrs. Grossman is an advisory planner for storage control subsystems. Since 1982, she has worked in the areas of marketing requirements, product planning, education, and implementation for the IBM 3880 Models 13 and 23 and the 3990, including extended functions. She is the author of a number of articles on the installation and use of both the 3880 and the 3990 storage controls. She joined IBM in Chicago in 1974 as a systems engineer. While there, she was a storage systems and Mass Storage System (MSS) specialist. She received her B.S. in 1967 and M.S. in 1968, both in mathematics, from Northwestern University, Evanston, Illinois. Mrs. Grossman is a member of Phi Beta Kappa and Pi Mu Epsilon.

Reprint Order No. G321-5355.